Universidad de Matanzas
Facultad de Ciencias Técnicas
Departamento Informática

TRABAJO DE DIPLOMA EN OPCIÓN DEL TÍTULO DE INGENIERO INFORMÁTICO

PLUGIN DE NOTIFICACIÓN DE SOLICITUD DE CREACIÓN DE CURSOS EN LA MOODLE A LOS GESTORES DE CATEGORÍAS

Autor: Ana Mónica Rodríguez García
Tutor: M.Sc. Luis Andrés Valido Fajardo

Matanzas, 2023

Declaración de Autoría y Nota Legal

Yo, Ana Mónica Rodríguez García , declaro que soy la única autora de la siguiente tesis, titulada Plugin de notificación de solicitud de creación de cursos en la Moodle a los gestores de categorías y, en virtud de tal, cedo el derecho de copia de la misma a la Universidad de Matanzas, bajo la licencia Creative Commons de tipo Reconocimiento No Comercial Sin Obra Derivada, con lo cual se permite su copia y distribución por cualquier medio siempre que mantenga el reconocimiento de sus autores, no haga uso comercial de la obra y no realice ninguna modificación de ella.

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

_____

Ana Mónica Rodríguez García

Agradecimientos

La vida de las personas que se rodean de gente mas que buena es un constante AGRADECER es por ello que no puedo dejar de agradecer a una y cada una de las personas que hicieron posible que llegara a escribir justamente estas palabras :

- Gracias infinitas a Dios, tú, sin lugar a dudas escribes derecho en renglones torcidos.
- Gracias a mis padres, mi abuela y mis hermanos solo ustedes saben cuanto lloré porque pensaba que no podía con la carrera, solo ustedes saben cuantas horas estaba sentada delante de una computadora tratando de entender cosas que veía inalcanzables, solo ustedes supieron esperarme a las tantas de la madrugada con un café caliente para levantarme a estudiar.
- Gracias Leo ahora si te puedo decir tenias razón puedo con esto y con más, gracias por ser mi mejor maestro,mi mejor amigo y el mejor compañero de universidad que pude tener. No existe palabra que pueda agradecer todo lo que hiciste por mí.
- Gracias Javier Alejandro por estar, por no irte, por no abandonar el barco y luchar a la par mía por llegar a este día. Hiciste de estos cuatro años de carrera una aventura preciosa, nunca imaginé que la informática me regalaría algo tan bonito como lo que tú y yo hemos creado.
- Gracias a mi tutor Valido nada absolutamente nada de lo que hice en esta tesis hubiera sido posible sin tu ayuda, tu paciencia con mi mala memoria y tu compresión, gracias por decirme Tranquila compañera que lo vas a poder hacer!!!
- Gracias a todos mis amigos y el resto de mi familia que estuvieron ayudándome en el transcurso de mi carrera con un mensaje una llamada o simplemente con estar y escuchar las tantas veces que decía que no iba a poder con la carrera.

"Los sueños no se cumplen como quien cumple años, sino los sueños se estudian se madrugan se trabajan y un día ese sueño a lo mejor se hace realidad ".

Dedicatoria

A mi papá, mi mamá , mi abuela y mis hermanos todo lo que haga en esta vida siempre, siempre va a ser dedicado a ustedes .

Resumen

La Universidad de Matanzas cuenta con los Entornos Virtuales de Aprendizaje los cuales son un apoyo digital al proceso docente. La presente investigación surge a partir de la necesidad de contribuir al proceso de solicitud de los cursos de la Universidad de forma organizada. El objetivo fundamental es desarrollar un plugin que permita realizar la notificación de solicitud de cursos en la Moodle a los gestores de categoría . Para el desarrollo de la solución propuesta se realiza un análisis de las principales herramientas, tecnologías y metodologías que se utilizan en la construcción de un software. El proceso estuvo guiado por el uso de las siguientes herramientas y tecnologías: Visual Paradigm como herramienta CASE, UML como lenguaje de modelado, Moodle como plataforma en la cual se va a desplegar el plugin y PHP como lenguaje de programación. Finalmente se obtuvo un sistema informático que permite mejorar el proceso de verificación de los parámetros de calidad de los cursos mediante tablas, gráficas y generación de información en formato duro.

Palabras claves: Entornos Virtuales de Aprendizaje, Plugins, Moodle, PHP.

Abstract

The University of Matanzas has Virtual Learning Environments which are a digital support for the teaching process. In them there are numerous courses which are qualified by the distance education work group of the aforementioned university. The present investigation arises from the need to contribute to the verification process of the quality parameters required in the evaluation process of virtual courses by experts on the subject. The fundamental objective is to develop a plugin that allows increasing the degrees of the verification process of the quality parameters of the virtual courses of the platform. For the development of the proposed solution, an analysis of the main tools, technologies and methodologies used in the construction of software is carried out. The process was guided by the use of the following tools and technologies: Visual Paradigm as a CASE tool, UML as a model language, Moodle as a platform on which the plugin will be deployed, and PHP as a programming language. Finally, a computer system was obtained that allows improving the verification process of the quality parameters of the courses through tables, graphs and generation of information in hard format.

Keywords: Virtual Learning Environments, Plugins, Moodle, PHP.

Tabla de Contenido

Introducción

El contexto socio-cultural contemporáneo, caracterizado por la presencia ubicua y el uso intensivo de las Tecnologías de la Información y la Comunicación (TIC), coloca a los centros de enseñanzas independientemente de su tipo y nivel frente a la demanda de desarrollar en sus alumnos la alfabetización digital necesaria para la utilización competente de las herramientas tecnológicas. Los Entornos Virtuales de Aprendizaje (EVA) resultan un escenario óptimo para promover dicha alfabetización, ya que permiten abordar la formación de las tres dimensiones básicas que la conforman: el conocimiento y uso instrumental de aplicaciones informáticas; la adquisición de habilidades cognitivas para el manejo de información hipertextual y multimedia; y el desarrollo de una actitud crítica y reflexiva para valorar.
La Universidad de Matanzas no se ha quedado ajena a los nuevos retos que imponen las TIC, es por eso que ha incorporado a los procesos educativo e instructivo de la institución la utilización de los Entornos Virtuales de Aprendizaje. Los Entornos Virtuales de Aprendizaje (EVA) de la Universidad de Matanzas está sustentando con la tecnología *Moodle*. En él se encuentra publicada la gran mayoría de los cursos que componen las diferentes carreras universitarias que se imparten en la institución.
Como parte del proceso renovador que se lleva cabo en el país con respecto a la educación superior en la cual se hace más enfasís en la educación a distancia se hace necesario que los cursos publicados en los diferentes Entornos Virtuales de la Universidad de Matanzas tenga accesibilidad desde una red externa de la Universidad. Para lograr esto se realizaron un grupo de acciones:

1. Publicación de un Entorno Virtual de Aprendizaje de la Universidad de Matanzas con visibilidad externa a la red de la Universidad.

2. Solicitudes de creación de un espacio para los cursos en el Entorno Virtual de Apren-

dizaje de la Universidad de Matanzas por parte de los docentes que lo impartirán.

La *Moodle* como plataforma de gestión de cursos propone una organización de jerarquía de árbol donde los cursos son ubicados dentro de categorías que estas a su vez pueden estar ubicadas dentro de otras categorías. Esta estructura ha sido aprovechada por en la Universidad de Matanzas para la organización de los cursos virtuales. Por ejemplo la plataforma destinada a la formación de pregrado presentan una categoría para cada facultad que presenta la universidad a su vez dentro de la facultad existe una categoría destinada a cada carrera de la facultad y así dentro de esta existe una categoría para cada modalidad de estudio de la carrera que a su vez se organiza en dos categorías internas para los períodos en que se organiza un curso y dentro de estas son los colocados los cursos que responde a las asignaturas que se imparten en las carreras.
Cuando un docente solicita la creación de un espacio para la publicación de su curso en la plataforma desde la misma plataforma se le notifica a los usuarios administradores y gestores de la plataforma los cuales son los únicos encargados por su rol capaces de aprobar o rechazar dichas solicitudes. Este proceso puede aprobación presentan un grupo de inconvenientes en su realización:

- El grupo de solicitudes de creación espacios para nuevos cursos aumenta considerablemente teniendo en cuenta la apertura de nuevas carreras y cambios de planes de estudio de las carreras lo que provoca la aparición de nuevos cursos.

- Los administradores y gestores de la plataforma no son necesariamente docente y no conocen el claustro de docentes de cada carrera asi como como las asignaturas que componen la malla curricular de dicha carrera así como los docentes que la imparten lo que provoca que el proceso de aprobación se retarde producto a que se debe verificar que sea una solicitud sea correcta.

Para solventar la situación anterior se designaron por cada carrera un profesor que se le dio los permisos de gestor en la categoría que responde a la carrera para que pudiera gestionar todo lo referente a su carrera en la plataforma entre ellas la aprobación o no de las solictudes de nuevos cursos en su categoría.

El esquema de roles y permisos que propone la plataforma Moodle solo permite que los gestores del sistema y no los gestores de categorias sean los que reciban las notificaciones de solicitud de cursos lo cual imposibilita que los gestores de categorías sean notificados sobre la solicitud de creación de nuevos cursos. Debido a lo descrito anteriormente se propone como:

Problema de investigación: ¿Cómo desarrollar una extensión a las plataformas Moodle que componen el EVA de la Universidad de Matanzas que contribuya en el proceso notificación de solicitud creación cursos a los gestores de categorías.

Hipótesis: Mediante el uso de las herramientas y tecnologías actuales, es posible desarrollar una extensión a la plataforma Moodle que contribuya en el proceso notificación de solicitud creación cursos a los gestores de categorías.

Teniendo como **objeto de estudio:** El proceso notificación de solicitud creación cursos en la plataforma Moodle.

Objetivo general: Desarrollar una extensión a la plataforma Moodle que permita aumentar los grados del proceso del notificación de solicitud creación cursos a los gestores de categorías.

Enmarcado en el **campo de acción:** El proceso notificación de solicitud creación cursos en las plataformas Moodle del EVA de la Universidad de Matanzas

Posibles resultados:

1. Sistema para la notificación de solicitud creación cursos a los gestores de categorías en la Moodle de la Universidad de Matanzas.

2. Mecanismo para la generación digital y formato duro del estado de solicitudes de creación de cursos en las plataformas Moodle.

3. Herramienta de apoyo a la gestión y administración de los cursos en la plataforma Moodle

Para dar cumplimiento a los objetivos de esta investigación se definieron las siguientes **tareas investigativas**:

1. Elaboración del marco teórico de la investigación a través del estudio del estado de los avances existentes actualmente sobre el tema .
2. Identificación de los principales elementos que componen el proceso notificación de solicitud creación cursos en las plataformas Moodle del EVA de la Universidad de Matanzas.
3. Caracterización de los principales elementos que componen el proceso notificación de solicitud creación cursos en la plataforma Moodle.
4. Caracterización de los principales elementos que componen el proceso notificación de solicitud creación cursos en las plataformas Moodle del EVA de la Universidad de Matanzas.
5. Realización del levantamiento de requisitos funcionales y no funcionales.
6. Implementación del sistema que brinde solución al problema planteado.
7. Realización de pruebas para validar el cumplimiento de los requerimientos.

Durante la investigación se llevan a cabo varios métodos y técnicas en la búsqueda y procesamiento de la información como son: **Métodos Teóricos**

- **Analítico-sintético:** Para el estudio de los conceptos vinculados en los sistemas informáticos para la gestión económica, y para el análisis de la documentación necesaria, permitiendo así, un mejor entendimiento del problema a resolver y realizar la extracción de los elementos más importantes para el desarrollo del trabajo.
- **Histórico-lógico:** Para realizar un análisis de las soluciones similares y las tendencias actuales en los sistemas informáticos enfocados en la gestión económica.
- **Inductivo-Deductivo:** Para realizar el estudio de las principales herramientas existentes para el desarrollo de los sistemas informáticos para la gestión económica y según las características de las mismas, se definieron las cualidades que debe cumplir el sistema que se propone en el presente trabajo de diploma.

- **Modelación:** Para representar gráficamente conceptos y procesos con la finalidad de un mejor entendimiento de la solución que se propone.

Métodos Empíricos

- **Observación científica:** Para conocer el funcionamiento actual del proceso de gestión de contenidos para el aprendizaje utilizados en la Universidad de Matanzas, lo que permitió detectar las dificultades existentes en dicho proceso
- **Consulta bibliográfica:** Para consultar y analizar las fuentes de información relacionadas con los diversos tipos de sistemas informáticos para la gestión de contenidos para el aprendizaje.
- **Generalización:** Permite sistematizar en cada capítulo de la investigación los aspectos más significativos y llegar a conclusiones más objetivas y explícitas.

Técnicas para la obtención de información:

- **La entrevista:** Para la realización de encuentros planificados con los especialistas del proceso para obtener la información necesaria que será utilizada en el desarrollo del trabajo, posibilitando una buena comunicación y una participación activa y directa entre el equipo de desarrollo y el experto.

La estructura del documento se resume en los siguientes acápites:

Capítulo 1. Fundamentación teórica: Abarca la elaboración del marco teórico de la investigación. De igual manera se analizan aplicaciones de gestión de contenidos para el aprendizaje. Además se exponen las características principales de estos sistemas. Se analizan las principales tendencias, tecnologías, metodologías y softwares utilizados en la actualidad para el desarrollo de aplicaciones de gestión de contenidos para el aprendizaje. A su vez se analizan y se fundamenta la selección de estas para el desarrollo de la solución propuesta.

Capítulo 2. Análisis y diseño del sistema: Se reflejan las actividades realizadas en los procesos de análisis y diseño de la solución propuesta; proceso que será guiado por la metodología de desarrollo seleccionada. En el mismo se realiza el modelo de dominio donde se describen las entidades que intervienen con el objetivo de facilitar la comprensión de los principales conceptos que se utilizarán en el proceso de negocio identificado. Se exponen los artefactos más importantes que describen el flujo normal de eventos que ocurren en el sistema, se realiza una descripción de la solución propuesta, planteándose los requisitos funcionales y no funcionales. Se define la arquitectura que tendrá la solución propuesta.

Capítulo 3. Implementación y pruebas: En este capítulo se describe la fase de implementación del sistema, según la metodología propuesta. Se realizan además una serie de pruebas que permiten validar el correcto funcionamiento de la solución, verificándose así, que el mismo cumple con todos los requerimientos y exigencias del cliente.

Finalmente, se presentan las conclusiones y las recomendaciones de la investigación para dejar el camino abierto a futuros estudios relacionados con el tema abordado. De igual forma, quedan recogidas las bibliografías y anexos que fueron utilizados y conformados respectivamente para el desarrollo de la solución.

Capítulo 1: Fundamentación Teórica

En el presente capítulo se engloban conceptos fundamentales asociados al Plugin de notificación de solicitud de creación de cursos en la Moodle a los gestores de categorías. Se analizan las características de la plataforma Moodle sus roles haciendo énfasis en el rol de gestor, el flujo de como funcionaria el proceso de solicitud de cursos dentro de la plataforma. Se investigan y se dan a conocer las principales tendencias, tecnologías, metodologías y softwares utilizados en la actualidad para el desarrollo de sistemas de creación de cursos para gestores de categorías virtuales. A su vez se analizan y se fundamenta la selección de estas para el desarrollo de la solución propuesta.

1.1 Entornos Virtuales de Aprendizaje

En la actualidad, la formación virtual es el uso de software especializado en la enseñanza a traves de dispositivos electrónicos tales como tablets, smartphones y computadoras. Este tipo de formación utiliza un software especifico denominado genéricamente como plataformas de formación virtual. Existen diferentes grupos de entornos de formación según la finalidad de los mismos, de los cuales se destacan los Sistemas de Gestión del Conocimiento (Learning Management System, LMS), también llamados Virtual Learning Environment (VLE) o Entornos Virtuales de Aprendizaje (EVA), los cuales representan una agrupación de las partes más importantes de los demás entornos. Los EVA se podrían describir como entornos que:

- Permiten el acceso a través de navegadores, protegido generalmente por contraseña o cable de acceso.
- Utilizan servicios de la web 1.0 y 2.0.
- Disponen de un interface gráfico e intuitivo.

- Integran de forma coordinada y estructurada las diferentes extensiones.
- Posibilitan la comunicación e interacción entre los estudiantes y el profesor-tutor.
- Presenta diferentes tipos de actividades que pueden ser implementadas en un curso.
- Incorporan recursos para el seguimiento y evaluación de los estudiantes.

Entornos Virtuales de Aprendizaje UM

En la Universidad de Matanzas se cuenta con un departamento especializado en el mantenimiento y atención al sitio http://eva.umcc.cu el cual tiene como principal objetivo brindarle a los estudiantes una plataforma digital en la cual se complementan sus estudios. En el centro de altos estudios mencionado con anterioridad el curso académico 2021-2022 inició en el mes de febrero con las plataformas digitales como escenario, debido a la incidencia de la COVID-19 en esta provincia. «Entre los recursos que se han potenciado en este periodo tenemos dos grupos principales: los asociados a los entornos virtuales de aprendizaje y los vinculados con la biblioteca universitaria o Centro de Información Científico-Técnica (CICT)»,para los cuales «se realizó un convenio con ETECSA en el cual, utilizando Nauta Hogar o las zonas Wi-Fi, los usuarios tendrían exceptos de pago los servicios que las universidades declararan» explicó Jósval Díaz Blanco, director del departamento de Informatización de la Universidad de Matanzas.

Proceso de creación y solicitud de nuevos cursos

Actualmente en La Universidad de Matanzas para realizar una solicitud de curso el usuario docente debe solicitar un curso a través de un formulario que presenta el propio sistema una vez que el usuario docente o el profesor llena los datos del formulario y envía este formulario el sistema registra los datos de este formulario como una solicitud de curso pendiente dicha solicitud le llega en forma de notificación al administrador o el gestor global de la plataforma, este es entonces el encargado de aprobar dicha solicitud,pero para poder realizar la aprobación el administrador debe conocer todos los planes de estudio de toda la carrera es decir todas las asignaturas que se imparten en cada una de las carreras y ademas conocer los profesores que las imparten . Este trabajo se vuelve un tanto lento,engorroso en

ese momento ya que el gestor debe apoyarse en cada jefe de departamento para chequear que la información que les llega en la solicitud en cuanto al profesor y al nombre del curso es la correcta para su aprobación, de ser esta información correcta entonces se aprueba el curso.

1.2 Moodle

Moodle es una plataforma de aprendizaje diseñada para proporcionarle a educadores, administradores y estudiantes un sistema integrado único, robusto y seguro para crear ambientes de aprendizaje personalizados. La plataforma está construido por el proyecto Moodle,dirigido y coordinado por el Cuartel General Moodle, la cual se encuentra soportada financieramente por una red mundial de cerca de 80 compañías de servicio más conocidas como Moodle Partners o Socios de Moodle.

MOODLE es un acrónimo en inglés de Module Object-Oriented Dynamic Learning Environment (Entorno de Aprendizaje Dinámico Modular Orientado a Objeto).Es una de las plataformas más usadas en la educación, es un centro de gestión de aprendizajes con enfoque constructivista permitiendo la comunicación entre los participantes y promoviendo el trabajo cooperativo.

Una de las principales características de MOODLE, radica en que es un software construido para el aprendizaje globalmente diseñado para soportar la enseñanza-aprendizaje, fácil de usar, gratuito, actualizado, flexible, seguro, escalable, robusto, basado en web, entre otras. Esto implica que las personas tienen la posibilidad para estudiar y aprender sin limitaciones de espacio y tiempo, basta con asumir el rol de estudiante virtual para hacer uso de los recursos y herramientas para el proceso de aprendizaje.(Rodríguez, 2018)

- Construido para el aprendizaje: Mundialmente probado y de confianza impulsando a decenas de miles de ambientes de aprendizaje global, MOODLE tiene la confianza de instituciones y organizaciones grandes y pequeñas. El número de usuarios de MOODLE a nivel mundial, de más de 79 millones de usuarios, entre usuarios académicos y empresariales, lo convierten en la plataforma de aprendizaje más ampliamente utilizada del mundo.Diseñado para soportar tanto la enseñanza como el aprendi-

zaje. Con más de 10 años de desarrollo guiado por la pedagogía de constructivismo social, MOODLE proporciona un conjunto poderoso de herramientas centradas en el estudiante y ambientes de aprendizaje colaborativo, que le dan poder, tanto a la enseñanza como al aprendizaje.

- Fácil de usar : Una interfaz simple, que posee características de arrastrar y soltar, y recursos bien documentados, junto con mejoras continuas en usabilidad, hacen a MOODLE fácil de aprender y usar. Gratuito, sin cargos por licenciamiento MOODLE es proporcionado gratuitamente como programa de Código Abierto, bajo la Licencia Pública General GNU (GNU General Public License). Cualquier persona puede adaptar, extender o Modificar MOODLE, tanto para proyectos comerciales como no-comerciales, sin pago de cuotas por licenciamiento, y beneficiarse del costo/beneficio, flexibilidad y otras ventajas de usar MOODLE.

- Siempre actualizado : La implementación de MOODLE en código abierto significa que MOODLE es continuamente revisado y mejorado, para adecuarse a las necesidades actuales y cambiantes de sus usuarios.

1.3 Enfoques existentes

existen varias extensiones de Moodle los cuales realizan resúmenes de determinados aspectos de forma automática de cursos virtuales.

- Tal es el caso de *Course Request Manager* es un verdadero ahorro de tiempo para los administradores del sitio que necesitan procesar solicitudes para nuevas configuraciones de cursos.El bloque administrador de solicitudes de curso permite a los administradores diseñar e implementar fácilmente formularios de solicitud de cursos personalizados que pueden utilizar los usuarios con los permisos adecuados. El problema con este sistema es que solo es compatible con Moodle 3.7 y para nuestro sistema estamos utilizando Moodle 4.1.

- *Configurable Reports* o Informes Configurables como se le conoce en español el cual es un generador de informes personalizados de Moodle muy destacado en la categoría

de plugin tipo Bloques. El mismo esta diseñado de forma modular para permitir a los desarrolladores crear nuevos complementos de forma rápida.Es por ello que debido a su diseño de interfaz seleccionamos diversos elementos para el desarrollo de nuestro sistema . El problema con este sistema es que crea informes muy superficiales y el objetivo de la presente investigación es centrado en solicitudes de cursos especificas y con determinados requisitos .

1.4 Metodologías para el desarrollo de software

En la actualidad existen una gran cantidad de metodologías para el desarrollo de software, separadas en dos grandes grupos; las metodologías tradicionales o pesadas y las metodologías ágiles. Las metodologías tradicionales se basan en las buenas prácticas dentro de la ingeniería del software, siguiendo un marco de disciplina estricto y un riguroso proceso de aplicación. Las metodologías ágiles, en cambio, representan una solución a los problemas que requieren una respuesta rápida en un ambiente flexible y con cambios constantes, haciendo caso omiso de la documentación rigurosa y los métodos formales.
Las metodologías se clasifican en Ágiles o Tradicionales. Las metodologías Tradicionales se caracterizan por:

- El proyecto esta planificado de antemano sin posibilidad de cambiar los requisitos/necesidades.
- Supone que el tiempo y el costo son variables pero los requisitos son fijos.
- Factores como el costo, el alcance y el tiempo son importantes.

Por otro lado las metodologías Ágiles se caracterizan por:

- Prioriza el trabajo en equipo, la colaboración con los clientes y la flexibilidad.
- Flexible a cambios y desarrollo de especificaciones.
- Dedica menos tiempo a la planificación inicial y la priorización.

A continuación se muestra una tabla en la cual se comparan ambos grupos:

Metodologías Tradicionales	Metodologías Ágiles
Predictivos	Adaptativos
Orientado a procesos	Orientado a personas
Proceso rígido	Proceso flexible
Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños.
Poca comunicación con el cliente	Comunicación constante con el cliente.
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación
Se basan en normas provenientes de estándares seguidos por el entorno de desarrollo	Se basan en heurísticas provenientes de prácticas de producción de código
Cierta resistencia a los cambios	Preparados para cambios durante el proyecto
Impuestas externamente	Impuestas internamente por el equipo
Proceso muy controlado, numerosas normas	Proceso menos controlado, con pocos principios
Contrato prefijado	Contrato flexible e incluso inexistente
Cliente interactúa con el equipo de desarrollo mediante reuniones	El cliente es parte del desarrollo
Grupos grandes	Grupos pequeños (<10)
Más artefactos	Pocos artefactos
La arquitectura del software es esencial	Menor énfasis en la arquitectura del software

Tabla 1.1: Fuente: Tomada de (Letelier, 2006)

1.4.1 Elección de la metodología de desarrollo de software

La metodología de desarrollo de software seleccionada para realizar la presente investigación es Extreme Programming (XP) debido a que contiene un conjunto de técnicas que dan agilidad y flexibilidad a la investigación. Debido a los constantes cambios que puede proponer el cliente y el corto periodo para el desarrollo de software el uso de la metodología XP es idóneo ya que pone más énfasis en la adaptabilidad que en la previsibilidad, además los cambios de requisitos sobre la marcha son un aspecto natural.

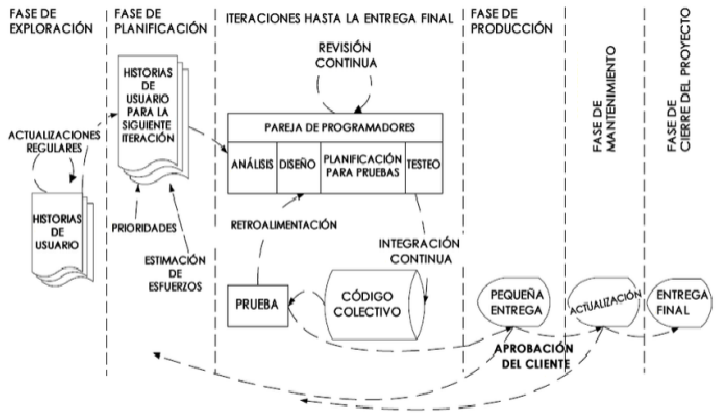

Figura 1.1: Modelo de dominio

1.5 Herramientas y tecnologías de desarrollo

Las herramientas y tecnologías de desarrollo son fundamentales para lograr una correcta planificación, desarrollo, funcionalidad y despliegue de cualquier software que se desarrolle.Una correcta elección de las mismas puede evitar futuros errores y agilizar el proceso de desarrollo del software. A continuación se abordara acerca de las herramientas y tecnologías seleccionadas para el desarrollo del trabajo en cuestión, se profundizará en el lenguaje de modelado, herramienta CASE, bibliotecas de desarrollo, entorno de desarrollo,y lenguajes de programación a utilizar.

1.5.1 Lenguaje de modelado

El lenguaje de modelado es cualquier lenguaje informático gráfico o textual que aprovisione el diseño y la construcción de estructuras y modelos siguiendo un conjunto sistemático de reglas y marcos. El lenguaje de modelado se utiliza principalmente en el campo de la informática y la ingeniería para diseñar modelos de software, sistemas, dispositivos y equipos nuevos. El contexto del lenguaje de modelado es principalmente textual y gráfico, pero según los requisitos y el dominio específico en uso. (Booch et al., 2006)

1.5.1.1 Elección del lenguaje de modelado

Se selecciona como lenguaje el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) debido a que se emplea para documentar, visualizar,construir y documentar los artefactos de la solución propuesta.El UML es una técnica de modelado de objetos y como tal supone una abstracción de un sistema para llegar a construirlo en términos concretos.La formalización de los diagramas del UML permite que cada modelo de sistemas se refine, admitiendo la inclusión y la refinación de las relaciones entre los elementos, chequeando la consistencia interna de cada uno de los elementos, y verificando la interconexión entre los elementos. UML surge como una herramienta de gran aceptación cuando es necesario soportar el diseño y la implementación de una solución automatizada, que subyace en un modelo de gestión de cualquier sistema.(Silva et al., 2018)

1.5.2 Herramienta CASE

CASE es el acrónimo de Computer Aided Software Engineering, ó ingeniería de software asistida por computadora, y se refiere al uso de programas de cómputo paraorganizar y administrar el desarrollo de software, sobre todo en aquellos proyectos donde se involucre una cantidad considerable de recursos y personal, tarea que para el gestor de proyectos representa una inversión considerable de tiempo, por tal motivo estas herramientas tienen como propósito el fungir como apoyo durante el ciclo de vida de todo el proyecto incluyendo todos sus componentes. (González López, González López, Gallud Lázaro et al., 1995)

1.5.2.1 Visual paradigm

Se selecciona Visual Paradigm porque es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue, permite dibujar todos los tipos de diagramas de clases,código inverso, generar código desde diagramas y generar documentación.(Field y Moore, 2005)

Visual Paradigm presenta todos los diagramas UML y herramientas de Diagrama Entidad Relación (ERD por sus siglas en inglés) esencialmente en el diseño de sistemas y bases de datos. Las innovadoras herramientas de modelado como Resource Catalog, Transitor y Nicknamer hacen que el modelado de sistemas sea fácil y rentable. Doc.Composer le permite producir especificaciones de diseño detalladas listas para usar en discusión con solo unos pocos ciclo.(Peña y Baquero, 2016)

1.5.3 Biblioteca de Desarrollo

Una biblioteca de desarrollo o librería es un conjunto de archivos que se utiliza para desarrollar software. Suele estar compuesta de código y datos, y su fin es ser utilizada por otros programas de forma totalmente autónoma, la misma esta compuesta por funciones, clases, tipos predefinidos, constantes, variables globales y macros, etc. Existen dos tipos de librerías en programación, las librerías estáticas y las dinámicas.(Arenas Morales y Brios Guevara, 2019)

- Librerías estáticas Estas se graban en un programa como ejecutables. Sirven exclusivamente para esto; después, podemos borrarlas sin problemas, ya que el programa seguirá funcionando con la función necesaria.

- Librerías dinámicas Son distintas a las estáticas en cuanto a que no se copian en el programa al compilarlas. Las subrutinas son cargadas en tiempo de ejecución, en vez de enlazarse en tiempo de compilación.

1.5.3.1 Elección de la Biblioteca de Desarrollo

Bibliotecas Internas de Moodle Las bibliotecas internas de Moodle son utilizadas para manipular los datos del entorno, es decir, son usadas para obtener cantidad de cursos determinados, nombre de los cursos, datos importantes de los mismos como archivos existentes y nombres de estos archivos. Se emplean principalmente en la creación de extensiones para la plataforma Moodle y la modificación de las mismas. Son conocidas dentro de la plataforma como Interfaz de programación de aplicaciones (API, por sus siglas en inglés, *Application Programming Interface*) y algunos ejemplos de las mismas son:

- ChartJS : Biblioteca encargada de generar gráficas.
- PDF : Biblioteca encargada de generar y manipular PDF.

1.5.4 Entorno de Desarrollo

Un entorno de desarrollo es un espacio de trabajo que permite a los desarrolladores crear una aplicación o realizar cambios en ella sin afectar a la versión real del producto de software. También proporcionan a los desarrolladores una interfaz de usuario común (User Interface o UI por sus siglas en inglés) para desarrollar y depurar en diferentes modos

1.5.4.1 Visual Studio Code

Visual Studio Code es la herramienta seleccionada para desarrollar el software propuesto por el presente trabajo debido a que Visual Studio Code es un editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma, está disponible para Windows, GNU/Linux y macOS. VS Code tiene una buena integración con Git, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación. VS Code tiene una gran variedad de características útiles para agilizar el trabajo tales como(Dubé et al., 2007)

- **Multiplataforma**: Está disponible para Windows, GNU/Linux y macOS.

- **IntelliSense**:Proporciona sugerencias de código y terminaciones inteligentes en base a los tipos de variables, funciones, etc.

- **Depuración**: Incluye la función de depuración que ayuda a detectar errores en el código. De esta manera, se evita tener que revisar línea por línea a puro ojo humano para encontrar errores, además, es capaz de detectar pequeños errores de forma automática antes de ejecutar el código o la depuración como tal.

- **Uso del control de versiones**: Posee compatibilidad con Git, por lo que es posible revisar diferencias o lo que se conoce como git diff, organizar archivos, realizar commits desde el editor, y hacer push y pull desde cualquier servicio de gestión de código fuente.

- **Extensiones**: Las extensiones permiten personalizar y agregar funcionalidad adicional de forma modular y aislada. Por ejemplo, para programar en diferentes lenguajes,agregar nuevos temas al editor, y conectar con otros servicios. Realmente las extensiones permiten tener una mejor experiencia, y lo más importante, no afectan en el rendimiento del editor, ya que se ejecutan en procesos independientes. Por todo lo anteriormente planteado se decide utilizar a Visual Studio Code como IDE para nuestra investigación.

1.5.5 Sistema de Control de Versiones

Un control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante. Dicho sistema te permite regresar a versiones anteriores de tus archivos, regresar a una versión anterior del proyecto completo, comparar cambios a lo largo del tiempo, ver quién modificó por última vez algo que pueda estar causando problemas, ver quién introdujo un problema y cuándo, y mucho más. Usar un VCS también significa generalmente que si arruinas o pierdes archivos, será posible recuperar los fácilmente.

1.5.5.1 Git

Gracias a que Git es uno de los sistemas de control de versiones más utilizados en el mundo y cuenta con un gran apoyo de la comunidad de programadores es seleccionado como herramienta de control de versiones para desarrollar el plugin abordado en el presente trabajo.

1.5.5.2 GitHub

GitHUb es una plataforma para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. Esta plataforma es mantenida por Microsoft la cual es una compañía dedicada al desarrollo de uno de los sistemas operativos más utilizados del mundo, Windows. Esta plataforma es ideal para ir realizando salvas del proyecto a desarrollar.

1.5.6 Lenguajes

Un lenguaje de programación es un lenguaje formal que proporciona a una persona, en este caso el programador, la capacidad y habilidad de escribir (o programar) una serie de instrucciones o secuencias de órdenes en forma de algoritmos con el fin de controlar el comportamiento físico o lógico de un sistema informático, para que de esa manera se puedan obtener diversas clases de datos o ejecutar determinadas tareas. A todo este conjunto de órdenes escritas mediante un lenguaje de programación se le denomina programa informático.

1.5.6.1 Lenguaje de maquetado HTML

El lenguaje HTML (Hypertext Market Language por sus siglas en inglés)es un lenguaje de maquetado compuesto por una serie de etiquetas o marcas utilizado para crear una estructura o esqueleto en las diferentes vistas que componen el plugin a desarrollar.

1.5.6.2 Lenguaje de maquetado XML

XML es un lenguaje de marcado similar a HTML. Significa Extensible Markup Language(Lenguaje de Marcado Extensible) y es una especificación de W3C como lenguaje de

marcado de propósito general. Esto significa que, a diferencia de otros lenguajes de marcado,XML no está predefinido, por lo que debes definir tus propias etiquetas.Será utilizado para la creación de una estructura inicial de la base de datos del plugin.

1.5.6.3 Lenguaje de programacion CSS

La hoja de estilo en cascada o CSS (siglas en inglés de cascading style sheets) es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML y derivados. Es utilizado para definir la correcta presentación de las vistas del plugin.

1.5.6.4 Lenguaje de programacion JavaScript

JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, de esta forma creamos páginas web dinámicas las cuales realizan diferentes funciones en dependencia de tu interacción con la misma, asi como permitir implementar diferentes gráficas las cuales son usadas a través de una librería famosa en Moodle llamada ChartJs.

1.5.6.5 Lenguaje de programacion PHP

El lenguaje de programación PHP conocido en inglés como (Hypertext Pre-Processor) es decir Preprocesador de Hipertexto. PHP es un lenguaje de programación de propósito general que se ejecuta en el lado del servidor y es seleccionado debido a que es el lenguaje en el que se encuentra implementada la plataforma Moodle, lo cual brinda una compatibilidad nativa entre la solución a desarrollar y sobre la plataforma en la cual se va a ejecutar

1.5.7 Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) o DGBA (Data Base Management System) es un conjunto de programas no visibles que administran y gestionan la información que contiene una base de datos Los gestores de base de datos o gestores de datos hacen posible admnistrar todo acceso a la base de datos ya que tienen el objetivo de servir de interfaz entre ésta, el usuario y las aplicaciones.

1.5.7.1 María DB

MariaDB está desarrollada por la fundación con el mismo nombre y es considerada una versión mejorada de MySQL, tiene alta compatibilidad con su antecesora y su mejora radica en ofrecer una mejor seguridad al momento de su implementación (Foundation 2019). Pero una de sus características más destacables es la incorporación de nuevos motores de búsqueda para proporcionar mayor escalabilidad y mejores velocidades al momento de realizar consultas a la base de datos.(Pilicita Garrido, Borja López y Gutiérrez Constante, 2021)

1.6 Conclusiones parciales del capítulo

Una vez finalizado el presente capítulo, se ha llegado a la conclusión de que Moodle es uno de los Entornos Virtuales de Aprendizaje más utilizados en el mundo. Visual Studio Code con los plugins adecuados puede llegar a convertirse de un editor de texto a un excelente Entorno Integrado de Desarrollo para trabajar con el lenguaje PHP. Además se optó por utilizar las siguientes tecnologías y herramientas, eXtreme Programming (XP) como metodología de desarrollo de software, UML y el Visual Paradigm como herramienta CASE para visualizar, construir y documentar los artefactos del sistema. También se utilizará Visual Studio Code como editor de texto y PHP como lenguaje de programación ya que están más acordes a las necesidades y condiciones en la que se realiza la presente investigación.

Capítulo 2: Análisis y diseño del sistema

En el presente capítulo se reflejan las actividades realizadas en los procesos de análisis y diseño de la solución propuesta; proceso que será guiado por la metodología de desarrollo seleccionada previamente. En el mismo se realiza el modelo de dominio donde se describen las entidades que intervienen con el objetivo de facilitar la comprensión de los principales conceptos que se utilizarán en el proceso de negocio identificado. Se exponen los artefactos más importantes que describen el flujo normal de eventos que ocurren en el sistema, se realiza una descripción de la solución propuesta, planteándose los requisitos funcionales y no funcionales. Se define la arquitectura que tendrá la solución propuesta.

2.1 Consideraciones del negocio

Para el desarrollo de este sistema hay elementos que se deben de tener en consideración, tales como que el sistema solo puede ser accesible por los usuarios con roles de Gestor Global o Administrador, además, el sistema debe de ser capaz de integrarse a las plataformas Moodle.

2.2 Modelo del dominio

Un modelo de dominio muestra las clases conceptuales significativas en un dominio del problema, las cuales se centra en las abstracciones relevantes, vocabulario del dominio e información del dominio. Su utilidad radica en ser una forma de "inspiración" para el diseño de los objetos software, además de ser el artefacto clave del análisis orientado a objetos (Vázquez-Ingelmo y García-Peñalvo, 2019). A continuación se presentan las clases del dominio perteneciente a la solución:

2.2.1 Descripción de los conceptos del dominio

- **Usuario:** Persona capacitada para interactuar con el Visualizador.

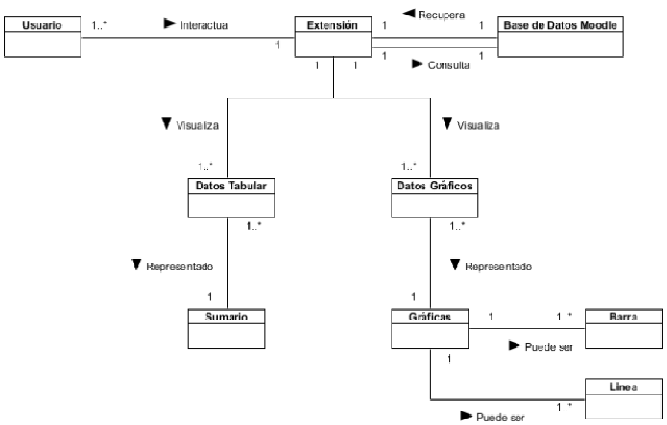

Figura 2.1: Modelo de dominio

- **Módulo**Herramienta encargada de procesar y visualizar los datos.
- **Datos Tabular:** Representación de la información en forma de tablas.
- **Datos Gráficos:** Representación de la información de forma gráfica.
- **Sumario:** Conjunto de elementos representados en tabla.
- **Variables:** Es la representación de las solicitudes de cursos hechas al sistema
- **Gráfica:** Es un tipo de representación de datos, generalmente numéricos, mediante recursos gráficos (líneas, vectores, superficies o símbolos), para que se manifieste visualmente la relación matemática o correlación estadística que guardan entre sí.
- **Acumulativa:**Gráfica que muestra de forma acumulativa el cumplimiento de los diferentes aspectos de todos los cursos en una categoría determinada.

2.3 Propuesta del sistema

Se propone una extensión para las solicitudes de los cursos a los gestores de categoría en los Entornos Virtuales de Aprendizaje de la Universidad de Matanzas, utilizando para su desarrollo herramientas libres, el cual debe de ser capaz de facilitar los resultados finales de una evaluación completa a un curso dado, bajo ciertos parámetros aportados por expertos.

2.4 Fase de exploración y planificación

En la fase de exploración y planificación los clientes describen sus necesidades en las Historias de Usuario que son los requisitos funcionales del sistema y establecen las prioridades de cada una. Al mismo tiempo, se define el tiempo de desarrollo de cada Historia de Usuario y se familiariza con las tecnologías, herramientas y prácticas que se utilizarán en el desarrollo del sistema para verificar la calidad de los cursos de los Entornos Virtuales de Aprendizaje de la Universidad de Matanzas. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema. La fase de exploración toma poco tiempo,dependiendo de la capacidad del programador con la tecnología y el alcance del proyecto.

2.5 Requisitos funcionales

Los requisitos funcionales son aquellos que describen qué debe hacer el sistema, desde el punto de vista de las necesidades del usuario, son capacidades o condiciones que debe cumplir el sistema y que están fuertemente ligados a las opciones del programa (Ruiz, 2012). Los requisitos funcionales (RF) del sistema propuesto se identificaron de acuerdo a las capacidades o condiciones que este debe cumplir. A continuación, se muestran los requisitos funcionales del sistema.

1. **RF-1:** Visualizar el acceso del plugin en la pestaña de curso en el panel de administración.
2. **RF-2:** Visualizar solicitud de cursos pendientes de forma tabulada.
3. **RF-3:** Enviar notificación a los gestores

1. **RF-3.1:** Enviar notificación de forma manual.
2. **RF-3.2:** Enviar notificación de forma automática.

1. **RF-4:** Generar reporte
2. **RF-4.1:** Generar las solicitudes pendientes en formato pdf.

1. **RF-5:** Generar estadísticas.

2. **RF-5.1:** Generar estadísticas de los cursos creados en los últimos años divido por meses.

3. **RF-5.2:** Generar estadísticas de los cursos creados en los últimos cinco años dividido por años.

2.6 Requisitos no funcionales

Los requerimientos no funcionales (RFN) son requisitos que imponen restricciones en el diseño o la implementación. Son propiedades o cualidades que el producto debe cumplir. Sin embargo estos requisitos no definen el éxito del producto, pero influyen considerablemente en la evaluación del mismo (Pytel et al., 2011). Teniendo en cuenta las características del sistema se definieron los siguientes requerimientos no funcionales:

1. **RNF-1 Interfaz:**

   a) La interfaz de usuario del sistema deberá ser diseñada de forma tal que permita el aprovechamiento del espacio.

   b) Clara y concisa. No debe dar lugar a la confusión del usuario.

2. **RNF-2 Estabilidad:** El sistema debe ser capaz de manejar los errores ocurridos durante la ejecución de la misma y avisando de la naturalidad del error.

3. **RNF-3 Rendimiento:** El sistema debe desempeñar su función de una manera fluida. Se debe buscar la experiencia de uso más agradable para el usuario.

4. **RNF-4 Optimización :**El tiempo de ejecución debe ser mínimo, para mejorar los tiempos de respuesta y la experiencia de uso del usuario.

5. **RNF-6: Ayuda y documentación:**Se brindarán manuales de ayuda que documenten cómo trabajar de forma adecuada con el sistema.

2.6.1 Historias de Usuario

Las Historias de Usuario se escriben en el lenguaje del cliente, representa un requisito que se debe satisfacer con la implementación del sistema y debe ser lo suficientemente sencilla para que el programador pueda saber que va a implementar. Si la Historia de Usuario cuando el cliente la escribe el programador entiende que no es lo completamente sencilla como para implementarla como una funcionalidad, entonces se divide en dos o más Historias de Usuario (Menzinsky et al., 2018). A continuación se muestra un ejemplo de las Historias de Usuarios del sistema :

HISTORIA DE USUARIO			
Orden	HU_2	**Nombre**	Visualizar solicitud de cursos pendientes de forma tabular
Riesgo	Bajo	**Prioridad**	Alta
Iteración		**Puntos estimados**	1
Descripción	El usuario debe de ser capaz de visualizar todas las solicitudes de cursos pendientes de aprobación. Se debe de mostrar la información básica de estas solicitudes en forma tabular.		
Continúa en la siguiente página			

Tabla 2.1 Continuación de la página anterior

Observación	■ 1- Nombre del profesor que solicita el curso ■ 2- Nombre corto del curso ■ 3- Nombre completo del curso ■ 4- Categoría ■ 5-Razón de solicitud del curso ■ 6- Usuarios responsables de aprobar el curso

Descripción de los campos que componen las Historias de Usuario:

- **Orden:** Está constituido por dos partes. La primera está referido al nomenclador HU (Historia de Usuario y la segunda corresponde el número de la funcionalidad que representa.
- **Nombre:** Nombre que identificará a la Historia de Usuario.
- **Riesgo:** Es el grado de incertidumbre en el desarrollo que se asocia a la Historia de Usuario. Determina la posibilidad real de implementarse o no con las condiciones previstas por el equipo de desarrollo (tiempo, recursos, personal). Puede ser Bajo, Medio o Alto.
- **Prioridad:** La prioridad la define el cliente, y es el grado de importancia que le concede a la funcionalidad.
- **Iteración:** Es el número de la fase en la cual se define la Historia de Usuario.
- **Puntos estimados:** Es un número entero que representa la cantidad de semanas que se dispone para el desarrollo de la Historia de Usuario. Las Historias de Usuario con

altos puntos estimados deben ser separadas en varias tareas. Un punto es una semana efectiva de desarrollo.

- **Descripción:** Se escribe una fundamentación de lo que hace la funcionalidad.
- **Observación:** Se escribe los elementos o detalles que se deben tener en cuenta para la implementación de la misma.

A partir de la solución propuesta se identificaron 7 requisitos funcionales agrupados en las siguientes Historias de Usuario:

- 1- Visualizar el acceso del plugin en la pestaña de curso en el panel de administración .
- 2- Visualizar las solicitudes de cursos pendientes de forma tabulada .
- 3- Enviar notificación a los gestores de forma manual .
- 4- Enviar notificación a los gestores de forma automática.
- 5- Generar las solicitudes pendientes en formato pdf.
- 6- Generar estadística de los cursos creados en los últimos años divididos por meses.
- 7- Generar estadística de los cursos creados en los últimos cinco años divididos por años.

La descripción del resto de las Historias de Usuarios que definen el sistema están adjunta en los anexos del documento.

2.6.2 Estimación de esfuerzo por Historias de Usuario

Las estimaciones de esfuerzo asociado a la implementación de las Historias de Usuario se realizan con el objetivo de lograr una planificación real en el desarrollo del sistema Visualizador y llevar un registro de la velocidad de desarrollo, basándose principalmente en la suma de puntos correspondientes a las Historias de Usuario.

La planificación se puede realizar basándose en el tiempo. La velocidad de desarrollo es utilizada para establecer cuántas Historias de Usuario se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de éstas. Se establece como medida el punto estimado. Un punto estimado, equivale a una semana ideal de programación. Las Historias de Usuario generalmente valen de 1 a 3 puntos. Teniendo en cuenta lo expuesto anteriormente la estimación de esfuerzo de las Historias de Usuario queda como se muestra:

Tabla 2.2: Estimación de esfuerzo por Historia de Usuario

Historia de Usuario	**Estimación de esfuerzo**
Visualizar el acceso del plugin en la pestaña de curso en el panel de administración	3 semana
Visualizar las solicitudes de cursos pendientes de forma tabulada .	3 semanas
Enviar notificación a los gestores de forma manual .	3 semana
Enviar notificación a los gestores de forma automática.	3 semana
Generar las solicitudes pendientes en formato pdf.	3 semana
Generar estadística de los cursos creados en los últimos años divididos por meses.	3 semana
Generar estadística de los cursos creados en los últimos cinco años divididos por años.	3 semana
Total de Historias de Usuario: 7	**Total de esfuerzo:** 21 semanas

A partir de la suma de los puntos de estimación de esfuerzo por cada Historia de Usuario, se calcula que el desarrollo del sistema tendrá una duración de 21 semanas.

2.6.3 Plan de iteraciones

Para lograr una mejor organización del trabajo y proporcionar un desarrollo iterativo e incremental, se crea el plan de iteraciones donde se planifica el orden de desarrollo de las

Historias de Usuario. Se definió realizar 10 iteraciones, su orden está determinada según las prioridades de las Historias de Usuario y las dependencias existentes entre ellas. La duración total de cada iteración dependerá de los puntos estimados de las Historias de Usuario que en él se desarrollan.

Tabla 2.3: Plan de iteraciones

Iteración	Historia de Usuario
1	Visualizar el acceso del plugin en la pestaña de curso en el panel de administración
2	Visualizar solicitud de cursos pendientes de forma tabulada
3	Enviar notificación a los gestores Enviar notificación a los gestores de forma manual Enviar notificación a los gestores de forma automática
4	Generar reporte Generar las solicitudes pendientes en formato pdf
5	Generar estadística Generar estadísticas de los cursos creados en los últimos años divididos por meses Generar estadísticas de los cursos creados en los últimos cinco años divididos por años

2.6.4 Plan de entrega

El plan de entrega es un documento que especifica con exactitud qué Historias de Usuario serán implementadas en cada entrega del sistema y sus prioridades, de modo que también permita conocer con claridad qué Historias de Usuario serán implementadas en la próxima iteración. Debe ser negociado y elaborado en forma conjunta entre el cliente y el equipo de desarrollado durante las reuniones de planificación de entregas, la idea es hacer entregas frecuentes para obtener una mayor retroalimentación. A continuación se muestra en el plan de entrega definido para el ciclo de desarrollo.

Tabla 2.4: Plan de entregas

Iteración	Historia de Usuario	Fecha de entrega

1	2	11 de septiembre del 2023
2	3	2 de octubre del 2023
3	1	23 de octubre del 2023
4	1	13 de noviembre del 2023
5	2	4 de diciembre del 2023

2.7 Estimación del costo

Entre los aspectos que se abordaran en este capítulo de análisis esta el del análisis económico de la solución propuesta. A continuación se realizará un desglose del coste de los elementos necesarios en esta investigación. Dichos elementos incluyen costes de personal,de hardware y de software. La investigación se realizará entre el 3 de julio del 2022 al 25 de noviembre de 2023 por lo tanto han sido 5 meses de trabajo. Teniendo en cuenta una jornada laboral de 8 horas tendremos un total de 1152 horas de trabajo, distribuidas entre diferentes tareas y diferentes roles profesionales que las llevan a cabo.

2.7.1 Coste de personal

La metodología de software escogida propone un equipo de desarrollo pequeño donde cada integrante tiene su rol y funciones bien definidas. Para determinar el coste del personal involucrado se va desglosar el equipo de acuerdo a la categoría de cada uno así como en la fase donde participa quedando el desglose del coste como se aprecia en la siguiente tabla.

Tabla 2.5: Coste de personal

Fase	Categoría	Horas	Coste/hora	Coste
Documentación	Analista	121	200 CUP	24200 CUP
Análisis	Analista	142	200 CUP	28400 CUP
Diseño	Diseñador	222	150 CUP	33300 CUP
Implementación	Programador	571	100 CUP	57100 CUP
Pruebas	Programador	96	100 CUP	9600 CUP

Total	**1152**	**152600 CUP**

2.7.2 Coste de hardware

Para el hardware calcularemos el coste según el período de amortización teniendo en cuenta una duración del proyecto de 7 meses. El equipo esta formado por ordenadores y tablet, necesitando uno de cada uno de estos dispositivos para el desarrollo.

Tabla 2.6: Coste de hardware

Equipo	Coste	Coste de amortizado
CPU	12000.00 CUP	1200.00 CUP
Monitor	3720.00 CUP	465.00 CUP
Teclado	108.00 CUP	17.90 CUP
Mouse	84.00 CUP	13.90 CUP
Computadora portatil	19200.00 CUP	1920.00 CUP
Total	35115.00 CUP	3616.80 CUP

2.7.3 Coste de software

En la realización de esta investigación se ha optado por utilizar software libre por lo que no tenemos ningún coste asociado al software.

2.7.4 Coste total

A partir del coste de cada de los elementos necesarios para la investigación se puede llegar al coste total, como se aprecia en la siguiente tabla.

Tabla 2.7: Coste total

Tipo de coste	Total
Coste de personal	152600 CUP
Coste de hardware	3616.80 CUP

Coste de software	0 CUP
Total	156216.80 CUP

Por tanto el coste total para la presente investigación asciende a: 156216.80 CUP

2.8 Diagrama de paquetes

Los diagramas de paquetes son diagramas estructurales que se emplean para mostrar la organización y disposición de diversos elementos de un modelo en forma de paquetes. Un paquete es una agrupación de elementos UML relacionados, como diagramas, documentos, clases o, incluso, otros paquetes. Cada elemento está anidado dentro de un paquete, que se representa como una carpeta de archivos dentro del diagrama, y que luego se organiza jerárquicamente dentro del diagrama. Los diagramas de paquetes se usan con frecuencia para proporcionar una organización visual de la arquitectura en capas dentro de cualquier clasificador UML, por ejemplo, un sistema de software (Besares y SÁNCHEZ, 2004). Los diagramas de paquetes bien diseñados ofrecen numerosos beneficios como:
Los plugins o extensiones creados para Moodle siguen un estándar determinado por la comunidad de programadores el cual se describe a continuación:

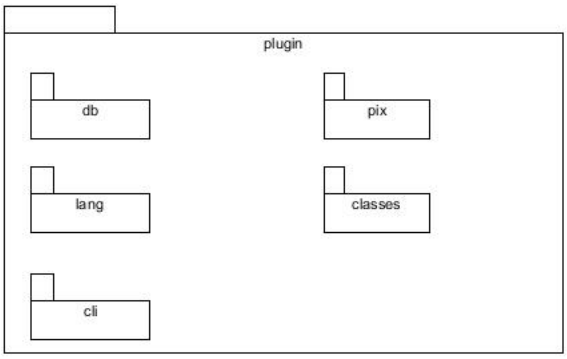

Figura 2.2: Diagrama de paquetes estándar de Moodle

Toda esta estructura es contenida dentro de la carpeta del proyecto que en nuestro caso es *Real-TimeViewer* La misma se organiza de la siguiente manera:

- *db:*Contiene toda la información que interactúa con la Data Base o Base de Datos de Moodle.

- *classes:* Es la encargada de almacenar todas las clases que conforman la lógica de la extensión en general.
- *lang:* : Contiene toda la información referente al idioma soportado por la extensión.
- *pix:* Almacena todos los elementos de tipo PNG, JPEG, etcétera.
- *cli:* Contiene los elementos que interactúan directamente con la linea de comandos.

La estructura del código fuente del sistema está organizado dentro la carpeta report debido a que el mismo se considera como un plugin tipo reporte. La estructura de sus carpetas se muestran a continuación:

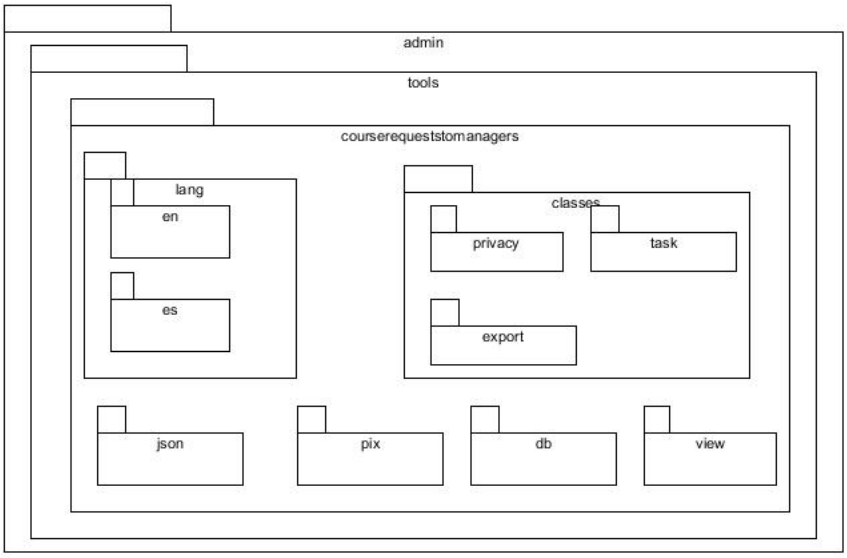

Figura 2.3: Diagrama de paquetes del sistema.

La misma se organiza de la siguiente manera:

- Paquetes *report*,*evaluation.quality* encierran toda la solución. La primera hace referencia al tipo de plugin y la segunda hace referencia al nombre de la extensión.
- Paquete *lang:*Como se describe anteriormente contiene todas las configuraciones del idioma.
  - Paquete *es:* Contiene todo lo referente a la traducción en el lenguaje Español.

    - Paquete *en:* Contiene todo lo referente a la traducción en el lenguaje Inglés. contenidos...

- Paquete *classes:* Contiene todas las clases que conforman la lógica de la aplicación.
    - Paquete*privacy:* Contiene las clases con protección privadas de la solución.
    - Paquete *task:* Almacena algunos archivos con requisitos específicos para guiar la solución del plugin.
    - Paquete *export:* Contiene las clases relacionadas con la exportación de documentos tipo PDF.
- Paquete *json:* Contiene todos los archivos json que intervienen en la evaluación de los parámetros.
- Paquete *view:* Contiene todas las clases que le permiten al usuario interactuar con los datos administrados y arrojados por el sistema
- Paquete *extra:* Contiene algunos archivos extras como los log o trazas del sistema.
- Paquete *pix:* Como se dio a conocer anteriormente contiene todos los elementos de tipo gráfico estáticos como iconos e imágenes en formatos PNG, JPEG, etcétera.
- Paquete *db:* Contiene todas las clases relacionadas con la interacción con la Base de Datos.

2.9 Diagrama de base de datos

Un modelo lógico de base de datos es una etapa intermedia de las que componen el proceso de desarrollo de una base de datos, el cual contiene representaciones de entidades y atributos, relaciones, identificadores exclusivos, subtipos y supertipos y restricciones entre relaciones (**gomez2013bases**). Para la realización del plugin se le agregarán tres tablas a la base de datos que componen el sistema Moodle las cuales se describen a continuación:

- *course request*: Es la entidad donde el plugin visualiza las solicitudes de curso.

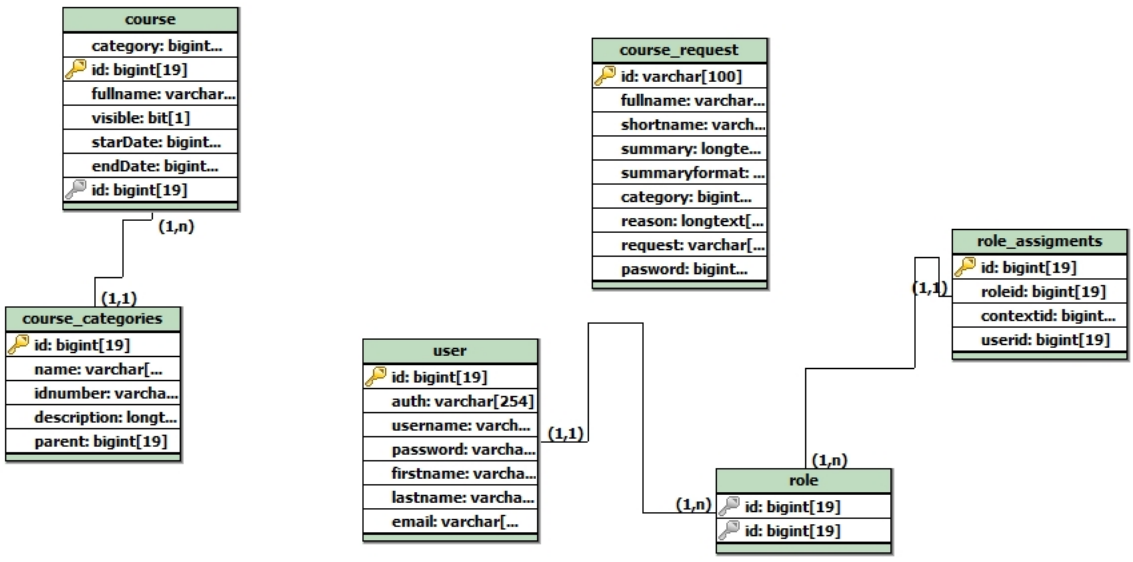

Figura 2.4: Diagrama de base de datos.

- *course category* : Almacena todo lo relacionado con las categorías de los cursos.
- *user*: Almacena todos los datos de los usuarios.
- *role assigments* :Almacena todos los roles y sus gestores correspondientes.

La tabla *Course* es nativa de Moodle solo se usa para representar la conexión entre las tres tablas y la base de datos por defecto de Moodle.

2.10 Tarjetas CRC

Las características más sobresalientes de las tarjetas CRC son su simpleza y adaptabilidad. Se define a una tarjeta CRC como una ficha de papel o cartón que representa a una entidad del sistema, las cuales permiten también que el equipo completo contribuya en la tarea del diseño. Estas tarjetas se utilizan para estructurar las clases y a su vez definir las responsabilidades sobre las mismas, así como la simulación de escenarios en el sistema (Goytia y González, 2014). El nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan en la parte izquierda, y las clases que se implican en cada responsabilidad en la parte derecha.

Clase: Es cualquier persona, evento, concepto, pantalla o reporte.

Responsabilidades: Las responsabilidades de una clase son las entidades que conoce y las que realizan sus atributos y métodos.

Colaboradoras: Los colaboradoras de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades.

A continuación se muestra unas de las tarjetas CRC obtenidas del sistema

Nombre de Clase: ***Presentation***	
Superclase: *View*	**Subclases:**
Responsibilidades	**Colaboradoras**
Vista que representa la pantalla inicial de la extensión de verificación de los parámetros de calidad.	

La descripción del resto de tarjetas CRC que definen el sistema están adjunta en los anexos del documento.

2.11 Tarea de Ingeniería

Las Tareas de Ingeniería son actividades que los programadores conocen que el sistema debe hacer. Deben ser estimables, su tiempo de implementación debe ser corto, aproximadamente entre uno y tres días, su objetivo es resolver las Historias de Usuario. Una Historia de Usuario puede tener una o varias Tareas de Ingeniería en dependencia de la funcionalidad a desarrollar. Pueden existir también tareas de ingeniería técnicas, que son aquellas que aunque no derivan directamente de una Historia de Usuario, es necesaria su consideración para que el sistema funcione.

En la siguiente tabla se muestra el formato utilizado para la confección de las tareas de ingeniería:

Tabla 2.9: Ejemplo de Tarea de Ingeniería.

Tarea de Ingeniería	
Número tarea: 01	**Número de Historia de Usuario:** 02
Nombre tarea: Visualizar solicitud de cursos pendiente de forma tabular.	
Tipo de tarea: Desarrollo	**Puntos estimados:** 1
Fecha inicio:11 de septiembre del 2023	**Fecha fin:** 2 de octubre del 2023
Programador responsable: Ana Mónica Rodríguez García	
Descripción: Implementar un mecanismo para cuando se realice una solicitud de curso se muestre una tabla donde muestre toda la informacion que se necesita para poder aceptar dicha solicitud	

Los campos de la tarjeta de las Tareas de Ingeniería reflejan lo siguiente:

- **Número tarea:** Representa el número por el que se identifica a la tarea. Cada tarea tiene un único número que la identifica.
- **Número Historia de Usuario:** Es el número de la Historia de Usuario a la que responde la tarea.
- **Nombre de tarea:** Define el nombre o funcionalidad concreta a la que se dedica la tarea, debe estar expresado en forma infinitiva.
- **Tipo de tarea:** Información del tipo de tarea a realizar, la misma puede ser:
  - **Desarrollo:** Tarea que se realizará por primera vez.
  - **Corrección:** Tarea que se realiza a partir de una anterior que no se realizó correctamente, es decir, que no pasó los casos de prueba satisfactoriamente.
  - **Mejora:** Tarea que se realiza a partir de una anterior incorporándole nuevos requerimientos.
  - **Otra:** Tarea que no corresponde con una de las anteriores, en este caso es necesario especificar el tipo de tarea o realizar una descripción más profunda de esta.

- **Puntos estimados:** Tiempo de duración de la tarea. El tiempo estimado es reflejado en días. La suma de los puntos estimados de las tareas de ingeniería de una Historia de Usuario no puede superar la cantidad de puntos estimados definidos para la Historia de Usuario.
- **Fecha inicial:** Fecha en la que se inicia el desarrollo de la tarea de ingeniería.
- **Fecha final:** Fecha en la que se concluye el desarrollo de la tarea de ingeniería.
- **Programador responsable:** Nombre del responsable de la realización de la tarea.
- **Descripción:** Es una breve descripción sobre lo que la tarea debe hacer o resolver.

La descripción del resto de Tareas de Ingeniería que definen el sistema están adjunta en los anexos del documento.

2.12 Pruebas

La metodología XP enfatiza en la realización de pruebas a lo largo de todo el desarrollo del software, con el fin de lograr un producto con calidad, reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su corrección. En este proceso no solo participa el desarrollador, también es importante la colaboración del cliente, sobre todo en las pruebas de aceptación. En XP las pruebas se dividen en dos grupos: pruebas unitarias encargadas de verificar el código y pruebas de aceptación orientadas a probar las funcionalidades del sistema

2.12.1 Pruebas de aceptación

Las pruebas de aceptación son las especificaciones para el comportamiento deseado y la funcionalidad de un sistema. Muestra por una Historia de Usuario dada, cómo el sistema se encarga de ciertas condiciones y con qué tipo de resultados. Los clientes junto a un miembro del equipo de desarrollo son los responsables de verificar que los resultados de estas pruebas sean los correctos para así tomar decisiones acerca de las mismas. Una Historia de Usuario no se puede considerar terminada hasta que no pase las pruebas de aceptación. Es

recomendable publicar los resultados de las pruebas de aceptación, para que todo el equipo de desarrollo esté al tanto de esta información. Al sistema además de las pruebas de funcionalidad, se le realizaron Pruebas de Regresión para comprobar que las no conformidades detectadas habían sido resueltas y que no se había afectado otras funcionalidades.

2.12.2 Casos de prueba

Los casos de prueba son evidencias de pruebas funcionales o unitarias que se realizan al sistema para comprobar su funcionamiento. Las pruebas funcionales son validaciones escritas desde la perspectiva del cliente, y las pruebas unitarias son validaciones desde la perspectiva del programador. El objetivo general es tener una forma para decirle al cliente que la Historia de Usuario está lista. Las pruebas funcionales o pruebas de aceptación, son las más importantes, ya que representan la medida de satisfacción del cliente para una funcionalidad que el sistema debe tener. Los casos de pruebas fueron definidos para cada Historia de Usuario establecida. Se comprueba el funcionamiento de cada una de las funcionalidades implementadas que responden a la misma. El formato utilizado para la confección de casos de pruebas se muestra a continuación en la tabla.

Tabla 2.10: Ejemplo de los casos de prueba.

Caso de prueba de aceptación	
Código: PA01_HU03	**Historia de Usuario:** HU03
Nombre: Enviar notificación a los gestores de forma manual .	
Descripción: Se intenterá enviar una notificación a los gestores de la solicitud del curso de forma manual.	
Condiciones de ejecución: Debe de existir al menos un gestor de categoría para aceptar la solicitud.	
Continúa en la siguiente página	

Tabla 2.10 Continuación de la página anterior

Pasos de ejecución: Una vez que el usuario realiza una solicitud el sistema lleva todos los datos referentes en una tabla. Se presiona el botón ENVIAR NOTIFICACIÓN A LOS GESTORES y luego aparecerá el mensaje en la campanita que aparecen en el sistema para todas las notificaciones.
Resultados esperados: Se debe visualizar un cuadro notificación que alerte de que existen campos vacíos. Se debe señalar de forma gráfica aquellas caja de texto que están vacías.
Evaluación de la prueba: Satisfactoria

Campos del caso de prueba

- *Código:* Identificador del caso de prueba. Dividido en dos partes. La primera representa la inicial del artefacto y la segunda representa el número con que se identifica la prueba.
- *Historia de Usuario:* Es el número de la Historia de Usuario a la que responde el caso de prueba.
- *Descripción:* Es una breve descripción del propósito de la prueba.
- *Condiciones de ejecución:* Condiciones especiales que deben tenerse en cuenta para ejecutar el caso de prueba.
- *Entradas / pasos de ejecución:* Entradas o funciones que deben ejecutarse para realizar el caso de prueba.
- *Resultado esperado:* Salida u objetivo que debe cumplir la funcionalidad a la que se le realiza el caso de prueba.
- *Evaluación:* Evaluación de éxito del caso de prueba. Prueba satisfactoria en caso de éxito o prueba insatisfactoria en caso de fallo.

Los casos de prueba son agregados a los artefactos de entrega que se realiza al cliente al terminar cada fase o iteración del proyecto. Las Historias de Usuario con evaluación insatisfactoria, serán corregidas en la próxima iteración a partir de nuevas tareas de ingeniería. Los casos de pruebas realizados al sistema se encuentran adjuntos en los anexos del documento.

2.12.3 Pruebas de compatibilidad

Las pruebas de compatibilidad son pruebas del funcionamiento del sistema con los diferentes navegadores, plataformas de hardware, etcétera, con los que puede interactuar el programa (Naik y Tripathy, 2010). La extensión a desarrollar es un complemento de un sitio web lo que garantiza su visualización en cualquier dispositivo además se utilizarán principios responsive para garantizar una correcta visualización e interacción con el usuario final.

2.12.4 Pruebas de usabilidad

Las pruebas de usabilidad son un servicio de aseguramiento de calidad que consiste en invitar a profesionales, cuyo perfil se adapta al de su público objetivo, a probar el producto y proporcionar comentarios valiosos sobre su facilidad de uso y eficiencia. El objetivo principal de las pruebas de usabilidad es identificar los problemas de usabilidad, recolectar comentarios pertinentes y mejorar la satisfacción de sus usuarios.Algunos de los atributos a tener en cuenta a lo largo de las pruebas de usabilidad son: (García, 2015).

1. **Facilidad de Aprendizaje** : Indica qué tan fácil es aprender la funcionalidad básica del sistema, como para ser capaz de realizar correctamente las tareas que desea llevar a cabo cualquier tipo de usuario.

2. **Eficiencia:** La eficiencia se determina por el número de transacciones por unidad de tiempo que el usuario puede realizar usando el sistema. Lo que se busca es la máxima velocidad de realización de tareas del usuario.

3. **Manejo de Errores:** Este atributo se refiere a la manera en que el sistema puede manejar los errores cometidos por el usuario mientras está realizando una tarea.

4. **Presentación visual apropiada:** El concepto de sistema se materializa al realizar el diseño de la parte visual de la interacción, es decir, la "interfaz gráfica de usuario".La misma es una parte importante del sistema, y un buen diseño gráfico puede hacer que un sistema aumente su nivel de usabilidad.

5. **Satisfacción:** Es el atributo más subjetivo. Muestra la impresión subjetiva que el usuario obtiene del sistema. Para ello se utilizan cuestionarios, encuestas y entrevistas, diseñados especialmente para recabar un cierto "grado de satisfacción" en función de aspectos predefinidos.

2.12.5 Pruebas de satisfacción de usuarios

Otro factor a analizar para saber si el sistema cumple con los requisitos necesarios de cara al cliente es realizar pruebas para determinar el agrado del usuario. La población utilizada para estas pruebas constaba de un equipo de 6 personas que enviaban comentarios y sugerencias para el sistema. Para este cometido se ha realizado una encuesta a estos usuarios, donde se preguntaban sobre las siguientes cuestiones:

1. El sistema es fácil de entender para el usuario.
2. La interfaz del sistema es atractiva y amigable.
3. El tiempo de respuesta es correcto bajo ciertas condiciones.
4. En términos generales, el sistema cumple su cometido y con buen rendimiento.
5. ¿Qué es lo que más le ha gustado de la aplicación?.
    - Interfaz.
    - Facilidad de uso.
    - Funcionalidades.
    - Tiempo de respuesta / Rendimiento

2.13 Conclusiones parciales del capítulo

En el capítulo presentado se definió el modelo del dominio, el cual refleja el punto de partida de la solución propuesta. Se especificaron además los requisitos del sistema que permitieron identificar las funcionalidades con las que este contará y que darán respuesta a las necesidades del usuario. Se realizó un estudio de los principales artefactos que guían la metodología de desarrollo, lo que permitió definir los aspectos necesarios para el desarrollo del sistema. Se obtuvo el Plan de entrega y el Plan de iteraciones confeccionado este último en 7 iteraciones y las tarjetas CRC que definen las entidades del sistema.

Capítulo 3: Implementación y prueba

3.1 Resultados de las pruebas

3.1.1 Pruebas de aceptación

Como resultado de las pruebas de aceptación Se detectaron un total de 12 no conformidades. A medida que se fue avanzando en las iteraciones disminuyeron el número de no conformidades hasta no quedar ninguna, demostrándose de esta manera que el sistema estaba listo para ser utilizado.

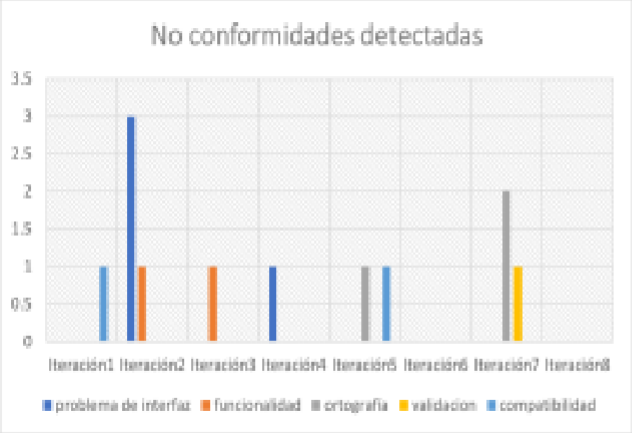

Figura 3.1: Resultados de las pruebas de aceptación

3.1.2 Pruebas de compatibilidad

Al contar con un sistema web con diseño responsive se respetaron las diferentes dimensiones de dispositivos computadoras de escritorios, laptops y tablets componentes se deben adaptan automáticamente a dimensiones mayores de 1134p exigida por la configuración de la resolución de cada uno de los dispositivos mencionados.

3.1.3 Pruebas de usabilidad

Para obtener los resultados de las pruebas de usabilidad nos basamos en los diferentes atributos planteados en la sección de Pruebas de usabilidad, los cuales fueron analizados uno a uno de la siguiente manera:

- **Facilidad de Aprendizaje:** El sistema posee un conjunto de tooltips en diferentes áreas lo cual proporciona un rápido aprendizaje
- **Manejo de Errores:** El sistema posee un conjunto de alertas para en caso de que ocurra algún error avisarle al usuario del mismo
- **Presentación visual apropiada:** Todos los botones, textos y tablas se encuentran distribuidos de forma simétrica para facilitar el uso de los diferentes apartados que presenta la extensión
- **Satisfacción:** Los diferentes usuarios y clientes que han probado la extensión se llevaron una buena impresión y reconocieron la facilidad de uso del mismo

3.1.4 Pruebas de satisfacción de usuarios

Para las cuatro primeras pruebas de satisfacción se les solicitó a los usuarios involucrados en estas pruebas que dieran una valoración de 1 a 5. Donde 1 significa estar en desacuerdo, 5 si esta completamente de acuerdo.

3.1.4.1 La aplicación es fácil de entender para el usuario

Tabla 3.1: Resultados de la pregunta 1 de las pruebas de satisfacción de usuarios **Fuente:** Elaboración propia

Valoración	Votos	Por ciento
Valoración de 1	0	0.00
Valoración de 2	0	0.00
Valoración de 3	0	0.00

Valoración de 4	3	50.00
Valoración de 5	3	50.00

Como se puede observar la mayoría de los usuarios que probaron el sistema quedaron satisfechos con la disposición de sus funcionalidades y su facilidad de uso.

3.1.4.2 La interfaz del sistema es atractiva y amigable

Tabla 3.2: Resultados de la pregunta 1 de las pruebas de satisfacción de usuarios **Fuente:** Elaboración propia

Valoración	Votos	Por ciento
Valoración de 1	0	0.00
Valoración de 2	0	0.00
Valoración de 3	0	0.00
Valoración de 4	2	33.33
Valoración de 5	4	66.66

Como se puede observar, los usuarios consideran el diseño de la interfaz de la aplicación atractiva y amigable.

3.1.4.3 El tiempo de respuesta es correcto bajo ciertas condiciones

Tabla 3.3: Resultados de la pregunta 1 de las pruebas de satisfacción de usuarios **Fuente:** Elaboración propia

Valoración	Votos	Por ciento
Valoración de 1	0	0.00
Valoración de 2	0	0.00

Valoración de 3	0	0.00
Valoración de 4	1	16.66
Valoración de 5	5	83.33

En esta pregunta pudimos comprobar como los usuarios consideraban notable el tiempo de respuesta del sistema.

3.1.4.4 En términos generales, el sistema cumple su cometido y con buen rendimiento

Tabla 3.4: Resultados de la pregunta 1 de las pruebas de satisfacción de usuarios **Fuente:** Elaboración propia

Valoración	Votos	Por ciento
Valoración de 1	0	0.00
Valoración de 2	0	0.00
Valoración de 3	0	0.00
Valoración de 4	2	33.33
Valoración de 5	4	66.66

Observando esta pregunta podemos ver que los usuarios están contentos con la funcionalidad obtenida en la aplicación. Se podría decir que se ha obtenido un rendimiento y una cantidad de funcionalidades que todos los usuarios esperaban. Han valorado positivamente el tiempo de respuesta y el rendimiento por lo tanto podemos concluir que el sistema cumple con los requisitos no funcionales expuestos en la fase de exploración.

3.1.4.5 ¿Qué es lo que más le ha gustado de la aplicación?

A modo de buscar un punto fuerte del sistema, se preguntó a los usuarios que creían más atractivo del sistema. Esto sirve al equipo de desarrollo para determinar que punto del

sistema causa más impacto en los usuarios y mejorarlo.

Tabla 3.5: Resultados de la pregunta 4 de las pruebas de satisfacción de usuarios **Fuente:** Elaboración propia

Valoración	Votos	Por ciento
Interfaz	1	16.66
Facilidad de uso	2	33.33
Funcionalidades	3	50.00
Rendimiento	0	00.00

Como se puede ver, las opciones más marcadas en la encuesta fueron las funcionalidades y la facilidad de uso. Por lo tanto se debe de trabajar más los aspectos de interfaz y rendimiento de la extensión ya que fueron los indicadores menos votados.

3.2 Interfaces del sistema

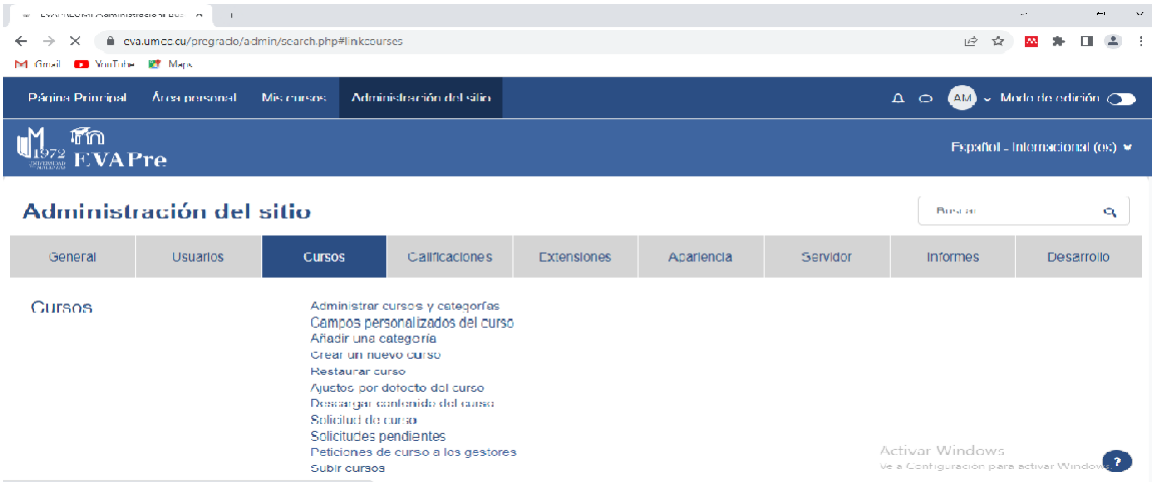

Figura 3.2: Imagen del sistema en la cual se muestra donde debemos acceder para poder ver las peticiones de los cursos

Figura 3.3: Tabla la cual muestra todo lo referido con la solicitud del curso

Figura 3.4: imagen que muestra como se realiza la notificación a os gestores ya sea a través del sistema o de correo electrónico

3.3 Conclusiones parciales del capítulo

Una vez finalizado el presente capítulo, se ha podido arribar a las siguientes conclusiones parciales:

Con la finalización de este capítulo se logró poder obtener los resultados de las pruebas

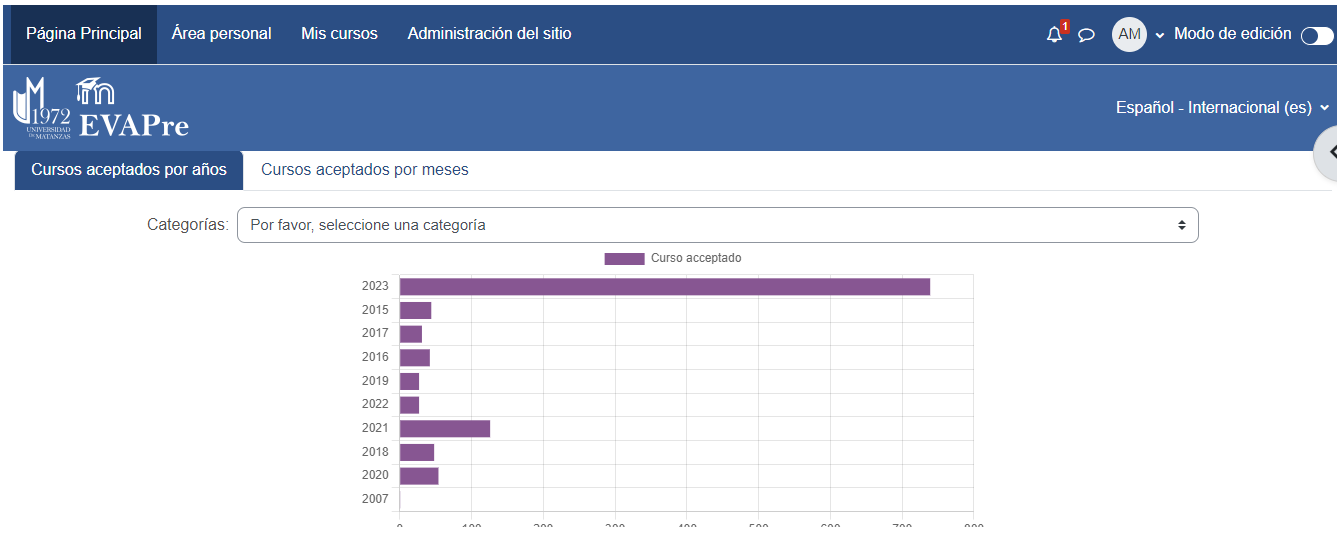

Figura 3.5: Imagen que muestra gráfica de información de las solicitudes de cursos generales

hechas al sistema así como una serie de interfaces que muestran todos y cada uno de los funcionamientos del software.

Conclusiones

- Se optó por utilizar las siguientes tecnologías y herramientas, eXtreme Programming (XP) como metodología de desarrollo de software, UML y el Visual Paradigm como herramienta CASE para visualizar, construir y documentar los artefactos del sistema. También se utilizará Visual Studio Code como editor de texto y PHP como lenguaje de programación ya que están más acordes a las necesidades y condiciones en la que se realiza la presente investigación.

- Se definió el modelo del dominio, el cual refleja el punto de partida de la solución propuesta. Se especificaron además los requisitos del sistema que permitieron identificar las funcionalidades con las que este contará y que darán respuesta a las necesidades del usuario. Se realizó un estudio de los principales artefactos que guían la metodología de desarrollo, lo que permitió definir los aspectos necesarios para el desarrollo del sistema. Se obtuvo el Plan de entrega y el Plan de iteraciones confeccionado este último en 7 iteraciones y las tarjetas CRC que definen las entidades del sistema.

- Se logró poder obtener los resultados de las pruebas hechas al sistema así como una serie de interfaces que muestran todos y cada uno de los funcionamientos del software.

Destacar que todo el desarrollo del trabajo se utilizó herramientas y tecnologías libres.

Recomendaciones

Dentro de las posibles mejoras y recomendaciones que se le hacen al sistema obtenido como parte de la solución expuesta en la presente investigación están:

1. Crear un esquema o tabla que permita guardar el envío de las notificaciones que se le hace a los gestores para poder tener un registro de que si se le envió o no la notificación al gestor y porque vía se hizo.

2. Lograr graficar la cantidad de cursos que han sido rechazados por categoría .

Referencias Bibliográficas

Arenas Morales, V. J. y L. Y. Brios Guevara (2019). "Desarrollo de un sistema informático para agilizar la atención y mejorar la administración en la biblioteca especializada de la facultad de Ciencias Físicas y Matemáticas-UNPRG, Lambayeque-2016." En.

Besares, J. G. V. y D. M. L. SÁNCHEZ (2004). "Ambiente visual para la integración de clases utilizando diagramas de paquetes". Tesis doct.

Booch, G. et al. (2006). *El lenguaje unificado de modelado: guía del usuario*. Addison-Wesley.

Dubé, C. et al. (2007). "The use of aspirin for primary prevention of colorectal cancer: a systematic review prepared for the US Preventive Services Task Force". En: *Annals of internal medicine* 146.5, págs. 365-375.

Field, A. P. y A. C. Moore (2005). "Dissociating the effects of attention and contingency awareness on evaluative conditioning effects in the visual paradigm". En: *Cognition and Emotion* 19.2, págs. 217-243.

García, X. (2015). *UF1843 - Aplicaciones técnicas de usabilidad y accesibilidad en el entorno cliente*. Ediciones Paraninfo, S.A. ISBN: 9788428397810.

González López, P., A. A. González López, J. A. Gallud Lázaro et al. (1995). "Herramientas CASE:¿ cómo incorporarlas con éxito en nuestra organización?" En: *Ensayos: revista de la Escuela Universitaria de Formación del Profesorado de Albacete*.

Goytia, L. y Á. González (2014). *Programación Orientada a Objetos C++ y Java*. Ingeniería y Ciencia Básicas. Grupo Editorial Patria. ISBN: 9786074389333.

Letelier, P. (2006). "Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)". En.

Menzinsky, A. et al. (2018). "Historias de usuario". En: *Ingeniería de requisitos ágil*.

Naik, K. y P. Tripathy (2010). *SOFTWARE TESTING AND QUALITY ASSURANCE: THEORY AND PRACTICE*. Wiley India Pvt. Limited. ISBN: 9788126525935.

Peña, D. M. y L Baquero (2016). "Extensión de la herramienta Visual Paradigm for UML para la evaluación y corrección de Diagramas de Casos de Uso". En: *Univ. las Ciencias Informaticas*.

Pilicita Garrido, A., Y. Borja López y G. Gutiérrez Constante (2021). "Rendimiento de MariaDB y PostgreSQL". En.

Pytel, P. et al. (2011). "Ingeniería de requisitos basada en técnicas de ingeniería del conocimiento". En: *XIII Workshop de Investigadores en Ciencias de la Computación*.

Rodríguez, M. R. (2018). "Aprendizaje con MOODLE". En: *Revista Multi-Ensayos* 4.8, págs. 18-25.

Ruiz, F. (2012). "Ingeniería del software I". En: *Ingeniería Del Software I*.

Silva, A et al. (2018). "Utilidad del Lenguaje Unificado de Modelado (UML) en el desarrollo de software profesional dentro del sector empresarial y educativo". En: *Ciencia-Cierta revista de divulgación científica* 56.

Vázquez-Ingelmo, A y F. García-Peñalvo (2019). "Modelo de Dominio". En.

Anexo A: Historias de Usuarios

A continuación la descripción de cada una de las Historias de Usuarios definidas para el sistema.

HISTORIA DE USO			
Orden	HU_01	**Nombre**	Visualizar el acceso del plugin en la pestaña de curso en el panel de administración
Riesgo	Medio	**Prioridad**	Alta
Iteración	1	**Puntos estimados**	1
Descripción	Solo los administradores generales del sistema es por ello que para poder realizar un envio de solicitud ya sea como correo electrónico o a través de una notificación del sistema el administrador debe de autenticarse para poder realizar dicha acción.		
Observación			

HISTORIA DE USO			
Orden	HU_02	**Nombre**	Visualizar las solicitudes de cursos pendientes de forma tabulada.

Riesgo	Alto	**Prioridad**	Alta
Iteración	1	**Puntos estimados**	2
Descripción	Una vez que se realice la solicitud del curso el sistema debe presentar todos los datos del usuario ya sea su nombre completo junto a su correo electrónico,el nombre de la categoría,la razón por la cual quiere solicitar dicho curso y el nombre completo y el correo electrónico de los gestores encargados de aceptar dicha solicitud		
Observación			

HISTORIA DE USO			
Orden	HU_03	**Nombre**	Enviar notificación a los gestores de forma manual .
Riesgo	Baja	**Prioridad**	Baja
Iteración	5	**Puntos estimados**	1
Descripción	Una vez que se muestran los datos de forma tabulada en el sistema entonces el administrador general realiza de forma manual es decir a través de una notificación del sistema el envio de la notificación a los gestores.		
Observación			

HISTORIA DE USO			
Orden	HU_04	**Nombre**	Enviar notificación a los gestores de forma automática.

Riesgo	Medio	**Prioridad**	Baja
Iteración	6	**Puntos estimados**	1
Descripción			
Observación			

HISTORIA DE USO			
Orden	HU_05	**Nombre**	Generar las solicitudes pendientes en formato pdf.
Riesgo	Medio	**Prioridad**	Alta
Iteración	2	**Puntos estimados**	1
Descripción	Una vez que el sistema muestra toda la información de la solicitud del curso se presiona el botón exportar pdf para que así se quede guardado todo lo referente a esa solicitud .		
Observación			

HISTORIA DE USO			
Orden	HU_06	**Nombre**	Generar estadística de los cursos creados en los últimos años divididos por meses.
Riesgo	Medio	**Prioridad**	Alta

Iteración	2	**Puntos estimados**	1
Descripción	El sistema debe ser capaz de a través de un gráfico de barra todas las solicitudes de curso que se hicieron en los últimos meses en cada categoría.		
Observación			

HISTORIA DE USO			
Orden	HU_07	**Nombre**	Generar estadística de los cursos creados en los últimos cinco años divididos por años.
Riesgo	Bajo	**Prioridad**	Baja
Iteración	2	**Puntos estimados**	1
Descripción	El sistema debe ser capaz de mostrar a través de un gráfico de barra las solicitudes de cursos que se han realizado en los últimos cinco años.		
Observación			

Anexo B: Tarjetas CRC

A continuación la descripción de cada una de las tarjetas CRC definidas para el sistema. Entidades pertenecientes al paquete *cim*.

Nombre de Clase: ***Data***	
Superclase: *Object*	**Subclases:** *Measurement*,*Serie*,*Variable*
Responsibilidades	**Colaboradoras**
Entidad genérica para representar cualquier dato que gestione el sistema.	

Nombre de Clase: ***Item***	
Superclase: *Object*	**Subclases:**
Responsibilidades	**Colaboradoras**
Entidad modelo que representa un elemento de acceso a las funcionalidades, visualizado lo mismo en el menú de navegación o en la ventana principal.	

Nombre de Clase: ***Measurement***	
Superclase: *Data*	**Subclases:**
Responsibilidades	**Colaboradoras**
Entidad encargada de representar una medición que esta asociada a una variable monitoreada por el sistema.	

Nombre de Clase: ***Serie***

Superclase: *Data*	**Subclases:**
Responsibilidades	**Colaboradoras**
Entidad que representa una serie visualizada en la gráfica de tendencia y que esta asociada a una variable monitoreada por el sistema.	

Nombre de Clase: ***Variable***	
Superclase: *Data*	**Subclases:**
Responsibilidades	**Colaboradoras**
Entidad que representa una variable monitoreada por el sistema.	

Entidades pertenecientes al paquete *communication*.

Nombre de Clase: ***BuildUrlRequest***	
Superclase:	**Subclases:**
Responsibilidades	**Colaboradoras**
Entidad encargada de construir el identificador de recursos uniforme perteneciente a una determinada petición que se le realizará al servidor.	

Nombre de Clase: ***ControllerRequest***	
Superclase:	**Subclases:**
Responsibilidades	**Colaboradoras**
Entidad encargada de controlar las peticiones realizadas desde el sistema al servidor.	*Request* *HttpRequestAsyncTask* *Session*

Anexo C: Tareas de Ingeniería

A continuación la descripción de cada una de las Tareas de Ingeniería definidas para el sistema.

Tarea de Ingeniería	
Número tarea: 01	**Número de Historia de Usuario:** 01,19
Nombre tarea: Incorporar los permisos de Internet al sistema	
Tipo de tarea: Desarrollo	**Puntos estimados:** 1
Fecha inicio: 1 de enero del 2018	**Fecha fin:** 1 de enero del 2018
Programador responsable: Ana Mónica Rodríguez García	
Descripción: Incorporar los permisos pertinentes al sistema para que el mismo pueda acceder al estado de conexión del dispositivo. Los permisos a incorporar son los siguientes: ▪ android.permission.INTERNET ▪ android.permission.ACCESS_NETWORK_STATE ▪ android.permission.ACCESS_WIFI_STATE ▪ android.permission.CHANGE_WIFI_STATE ▪ android.permission.CHANGE_NETWORK_STATE ▪ android.permission.CHANGE_WIFI_MULTICAST_STATE	

Tarea de Ingeniería	
Número tarea: 02	**Número de Historia de Usuario:** 01,19

Nombre tarea: Implementar mecanismo de consulta de conexión	
Tipo de tarea: Desarrollo	**Puntos estimados:** 4
Fecha inicio: 2 de enero del 2018	**Fecha fin:** 5 de enero del 2018
Programador responsable: Ana Mónica Rodríguez García	
Descripción: Implementación de un mecanismo que permita consultar el estado conexión del dispositivo y que informe de forma visual de dicho estado.	

Tarea de Ingeniería	
Número tarea: 03	**Número de Historia de Usuario:** 19
Nombre tarea: Iconografía referente a la HU_19	
Tipo de tarea: Diseño	**Puntos estimados:** 2
Fecha inicio: 2 de julio del 2018	**Fecha fin:** 3 de julio del 2018
Programador responsable: Ana Mónica Rodríguez García	
Descripción: Diseñar los íconos que represente los diferentes estados de conexión por los que puede presentar el dispositivo. Los iconos deben ser de las siguientes resoluciones 192x192, 144x144, 96x96, 72x72, 48x48 y 36x36 pixeles.	

Tarea de Ingeniería	
Número tarea: 04	**Número de Historia de Usuario:** 19
Nombre tarea: Implementar mecanismo de actualización del estado de conexión del dispositivo.	
Tipo de tarea: Desarrollo	**Puntos estimados:** 3
Fecha inicio: 4 de julio del 2018	**Fecha fin:** 6 de julio del 2018
Programador responsable: Ana Mónica Rodríguez García	
Descripción: Implementar un mecanismo que cada un segundo actualice el estado de conexión del dispositivo y lo muestre gráficamente en la barra de título utilizando el ícono que se corresponde con el estado.	

Tarea de Ingeniería	
Número tarea: 05	**Número de Historia de Usuario:** 15
Nombre tarea: Iconografía referente a la HU_15	
Tipo de tarea: Diseño	**Puntos estimados:** 2
Fecha inicio: 28 de mayo del 2018	**Fecha fin:** 29 de mayo del 2018
Programador responsable: Ana Mónica Rodríguez García	
Descripción: Diseñar el ícono que represente graficamente la funcionalidad de eliminar todas las series que han sido añadidas a la gráfica de tendencia.	

Tarea de Ingeniería	
Número tarea: 06	**Número de Historia de Usuario:** 15
Nombre tarea: Implementación mecanismo de eliminación de todas series.	
Tipo de tarea: Desarrollo	**Puntos estimados:** 3
Fecha inicio: 30 de mayo del 2018	**Fecha fin:** 1 de junio del 2018
Programador responsable: Ana Mónica Rodríguez García	
Descripción: Diseñar e implementar un mecanismo que permita eliminar todas las series adicionadas a la gráfica de tendencia.	

Tarea de Ingeniería	
Número tarea: 07	**Número de Historia de Usuario:** 13
Nombre tarea: Iconografía referente a la HU_13	
Tipo de tarea: Diseño	**Puntos estimados:** 2
Fecha inicio: 14 de mayo del 2018	**Fecha fin:** 15 de mayo del 2018
Programador responsable: Ana Mónica Rodríguez García	
Descripción: Diseñar el ícono que represente graficamente la funcionalidad de eliminar de las series añadidas a la gráfica de tendencia aquellas que el usuario seleccione.	

Anexo D: Resumen de las Tareas de Ingeniería por Historias de Usuarios

A continuación el resumen de las tareas de ingenierías implementadas por cada Historia de Usuario.

Tabla D.1: Resumen de Tareas de Ingeniería por Historias de Usuario

Historia de Usuario	No. Tarea	Tarea de Ingeniería
Comprobar conexión del dispositivo	2	01, 02
Cargar configuración definida para la terminal	5	72, 73, 74, 75, 76
Mostrar fecha y hora actual	2	68, 69
Mostrar estado actual de la batería del dispositivo	3	65, 66, 67
Autenticar terminal	4	46, 47, 48, 49
Cerrar sesión	2	50, 51
Visualizar datos de la terminal	2	70, 71
Visualizar sumario de variables	8	57, 58, 59, 60, 61, 62, 63, 64
Visualizar detalles de una variable	5	52, 53, 54, 55, 56
Visualizar comportamiento de una variable en un gráfico de tendencia	6	40, 41, 42, 43, 44, 45
Visualizar gráfico de tendencia	6	36, 37, 38, 39, 40, 41
Adicionar serie a la gráfica de tendencia	4	15, 16, 17, 18
Eliminar serie de la gráfica de tendencia	4	07, 08, 09, 10

Continúa en la siguiente página

Tabla D.1 Continuación de la página anterior

Mostrar u ocultar serie de la gráfica de tendencia	4	11, 12, 13, 14
Limpiar gráfico de tendencia	2	05, 06
Invertir los colores de texto y fondo de la gráfica de tendencia	2	23, 24
Exportar el gráfico de tendencia a formato de imagen	4	25, 26, 27, 28
Mostrar u ocultar leyenda de las series representadas en la gráfica	3	33, 34, 35
Visualizar estado actual de conexión del dispositivo	4	01, 02, 03, 04
Editar propiedades de una serie	4	29, 30, 31, 32

Anexo E: Casos de pruebas

A continuación la descripción de cada uno de los casos de pruebas realizados al sistemas.

Caso de prueba de aceptación	
Código: PA01_HU05	**Historia de Usuario:** HU05
Nombre: Auntenticar terminal con errores I	
Descripción: Se intenterá autenticar la terminal dejando los campos de anfritión y puerto del servidor vacío.	
Condiciones de ejecución: El dispositivo no debe haber iniciado sesión o ternerla cerrada en el servidor.	
Pasos de ejecución: Se selecciona la opción de auntenticar ya sea por el menú de navegación o en la vista principal del sistema. Se presiona el botón Aceptar del cuadro de díalogo con los campos anfitrión y puerto vacíos.	
Resultados esperados: Se debe visualizar un cuadro notificación que alerte de que existen campos vacíos. Se debe señalar de forma gráfica aquellas caja de texto que están vacías.	
Evaluación de la prueba: Satisfactoria	

Caso de prueba de aceptación	
Código: PA02_HU01	**Historia de Usuario:** HU01
Nombre: Comprobar conexión del dispositivo conectado a una red wifi	
Descripción: Se ejecutará el sistema en el dispositivo, él cual debe estar conectado a red wifi previamente y se comprobará que el sistema es capaz de notificar del estado de conexión del dispositivo.	

Condiciones de ejecución: El dispositivo debe estar conectado a una red wifi e instalado el sistema entre sus aplicaciones.
Pasos de ejecución: Se busca el ícono del sistema dentro del menú de aplicaciones del dispositivo y se da click sobre este.
Resultados esperados: Una vez iniciado el sistema el mismo debe visualizar mediante una notificación emergente el estado de conexión del dispositivo. Para este caso debe notificar que esta conectado a una red wifi.
Evaluación de la prueba: Satisfactoria

Caso de prueba de aceptación	
Código: PA03_HU03	**Historia de Usuario:** HU03
Nombre: Comprobar visualización de la fecha y hora del dispositivo	
Descripción: Se ejecutará el sistema en el dispositivo y se comprobará que el sistema es capaz de visualizar la fecha y hora que posee el dispositivo y actualizar dichos valores cada un segundo.	
Condiciones de ejecución: El dispositivo debe tener instalado el sistema entre sus aplicaciones.	
Pasos de ejecución: Se busca el ícono del sistema dentro del menú de aplicaciones del dispositivo y se da click sobre este.	
Resultados esperados: Una vez iniciado el sistema el mismo debe visualizar en la barra de título del sistema en su parte derecha superior la fecha y hora del dipositivo el cual debe cumplir con el siguiente formato *d-M-Y hh:mm:ss*. Cada un segundo estos datos deberán actualizarse.	
Evaluación de la prueba: Satisfactoria	

Caso de prueba de aceptación	
Código: PA04_HU04	**Historia de Usuario:** HU04

Nombre: Comprobar visualización del estado de la batería del dispositivo.
Descripción: Se ejecutará el sistema en el dispositivo y se comprobará que el sistema es capaz de visualizar el estado de la batería que posee el dispositivo y actualizar dicho estado cuando ocurra un cambio.
Condiciones de ejecución: El dispositivo debe tener instalado el sistema entre sus aplicaciones. El dispositivo no debe estar conectado a ninguna fuente de alimentación ajena a su batería.
Pasos de ejecución: Se busca el ícono del sistema dentro del menú de aplicaciones del dispositivo y se da click sobre este. Una vez hecho esto se deja en ejecucción el sistema por un tiempo superior a los 15 minutos.
Resultados esperados: Una vez iniciado el sistema el mismo debe visualizar en la barra de título del sistema en su parte derecha inferior el estado de la batería del dipositivo el cual debe cumplir con lo especificado en la HU_04. El estado de la batería debe cambiar el porciento de carga después de 15 minutos y el sistema debe actualizar su representación
Evaluación de la prueba: Satisfactoria

Caso de prueba de aceptación	
Código: PA05_HU06	**Historia de Usuario:** HU06
Nombre: Cerrar sesión DCRSA	
Descripción: Se cerrará la sesión desde el dispositivo el cual está conectado a la misma red del servidor que se encuentra en ejecucción.	
Condiciones de ejecución: El dispositivo debe haber iniciado sessión en el servidor y este debe estar en ejecucción y ambos conectados a la misma red.	
Pasos de ejecución: Se selecciona la opción de cerrar sesión ya sea por el menú de navegación o en la vista principal del sistema.	

Resultados esperados: La visualización de un cuadro emergente donde se informa al usuario que se pudo cerrar la sesión en el servidor. Desaparece del menú de navegación y de la vista principal del sistema la opción de cerrar sesión y aparece la de autenticarse.
Evaluación de la prueba: Satisfactoria

Caso de prueba de aceptación	
Código: PA06_HU07	**Historia de Usuario:** HU07
Nombre: Visualización de los datos de la terminal	
Descripción: Se comprobará la visualización de los datos de la terminal I	
Condiciones de ejecución: El dispositivo debe tener instalado el sistema entre sus aplicaciones. El sistema debe haber sido iniciado.	
Pasos de ejecución: Se busca el ícono del sistema dentro del menú de aplicaciones del dispositivo y se da click sobre este. Una vez hecho esto se despliega el menú de navegación.	
Resultados esperados: En la parte superior del menú de navegación se visualizará la MAC del dispostivo.	
Evaluación de la prueba: Satisfactoria	

Caso de prueba de aceptación	
Código: PA07_HU16	**Historia de Usuario:** HU16
Nombre: Invertir colores de texto y fondo en la gráfica de tendencia.	
Descripción: Se comprobará el correcto funcionamiento de la funcionalidad de invertir los colores de texto y fondo de la gráfica de tendencia.	

Condiciones de ejecución: El dispositivo debe tener instalado el sistema entre sus aplicaciones. El sistema debe haber sido iniciado. Se debe estar visualizando la vista de gráfica de tendencia.
Pasos de ejecución: Se busca el ícono del sistema dentro del menú de aplicaciones del dispositivo y se da click sobre este. Una vez hecho esto se despliega el menú de navegación o en la vista principal se da click sobre la opción de *Tendencia*. Dar click sobre el ícono en la barra de herramienta que representa la funcionalidad en la vista de la gráfica de tendencia.
Resultados esperados: Si el fondo es negro y los textos en blanco una vez dado un click sobre la funcionalidad los colores deben invertirse, el fondo blanco y los textos de negro, en caso de un nuevo click los colores se vuelven a invertirse y así sucesivamente tras cada click sobre el ícono de la funcionalidad.
Evaluación de la prueba: Satisfactoria