Universidad de las Ciencias Informáticas

Facultad 4

Sistema de gestión para el proceso de reserva de transporte de los centros de producción en la Universidad de las Ciencias Informáticas

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

Autor:

Darian Aguila Vega

Tutores:

Ing. Disnayle Jorge Chacón

Ing. Ively Viera Fajardo

La Habana. Junio, 2019

AGRADECIMIENTOS

A mis padres y a mi hermana por estar siempre ahí cuando los necesite, por ser mi fuente infinita de inspiración, por su incondicional apoyo. A mi familia por su comprensión. A mis tutores Ively y Disnayle por apoyarme y ayudarme siempre, por guiarme y estar siempre pendiente de mí. A mi otro tutor Yuriesky por su ayuda incondicional. A mis amigos Javier, Ernesto, Harold, Alberto, Rayner, Oscar, Beatriz y Brian por ser los mejores que podría tener. A todos Gracias.

DEDICATORIA

A mis padres.

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo Darian Aguila Vega, con carné de identidad 95121029708, soy el autor del trabajo titulado **"Sistema de gestión para el proceso de reserva de transporte de los centros de producción en la Universidad de las Ciencias Informáticas"** y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

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

Darian Aguila Vega

_____

Firma del Autor

Ing. Disnayle Jorge Chacón

_____

Firma del Tutor

Ing. Ively Viera Fajardo

_____

Firma del Tutor

RESUMEN

El presente trabajo propone un sistema de gestión para la reserva de transporte de los centros de producción en la UCI. Se analizaron diferentes fuentes bibliográficas relacionadas con las características, funcionalidades de sistemas de reserva de transporte a nivel nacional e internacional. La solución consiste en un sistema capaz de gestionar las solicitudes de reserva de transporte generadas por los líderes de proyectos, gestiona además las rutas creadas por los encargados de la dirección de transporte y notifica a los líderes si fue aceptada o no la solicitud. Emite distintos reportes para llevar un historial donde se pueda consultar la información por medio de filtros de búsqueda. El proceso de desarrollo estuvo guiado por la metodología de *software* AUP en su versión UCI, seleccionándose como principales tecnologías: el marco de trabajo Drupal 7.67, el lenguaje de programación PHP 7.0.2, el sistema de gestión de base de datos MySQL 5.0.12 y Visual *Paradigm* 8.0 como herramienta para el modelado. Las pruebas de *software* aplicadas al sistema de gestión para la reserva de transporte de los centros de producción en la UCI, demostraron que es una solución funcional, segura, con un rendimiento adecuado. Los resultados de la investigación evidenciaron que el sistema desarrollado permite mejorar el proceso de reserva de transporte de los centros de producción en la UCI.

Palabras clave: sistema, reserva, transporte.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

El transporte constituye una de las ramas de mayor impacto en la economía, por tal motivo se le concede una relevancia extraordinaria en la sociedad, porque permiten la circulación de bienes y personas, logrando una integración social que favorece el desarrollo. Por esta razón, con el paso del tiempo se ha visto una mejora en la eficiencia de este, con servicios renovados y una menor utilización de recursos. En este sentido, existe en la actualidad un verdadero interés en lograr que los medios de transporte utilicen menor cantidad de energía, circunstancia en parte relacionada con los problemas que pueden existir en el futuro en lo que respecta a provisión de combustible. La informatización de los sistemas dedicados a la gestión de la transportación, ha sido una estrategia adoptada por gobiernos y administraciones para hacer un uso racional del combustible y de todos los recursos relacionados en este complejo tema de movilizar personal o recursos materiales.

En Cuba, la gestión del transporte y sus respectivas reservaciones tienen un gran auge, por ello se solicita fomentar la expansión de proyectos que agilicen y hagan más eficientes los trámites requeridos, facilitando a su vez el acceso a la información y garantizando su calidad. Por tal razón varias entidades han dedicado tiempo y recursos para impulsar la informatización de los procesos asociados a la gestión del transporte. Una de ellas es la Universidad de las Ciencias Informáticas (UCI), institución educacional creada en el 2002 a partir de una idea del Comandante en Jefe Fidel Alejandro Castro Ruz, la cual tiene como meta la formación de ingenieros informáticos y la potenciación de la industria cubana del *software* (Rodríguez González, et al., 2015).

La UCI tiene como misión producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación (UCI, 2016). Para llevar a cabo lo anterior se han creado en su interior diversos centros de producción de *software*. Todo esto ha posibilitado el desarrollo de varios proyectos, a través de los cuales, se han informatizado una parte de los principales procesos que se ejecutan, tanto del centro educacional como en la sociedad cubana, e incluso a nivel internacional.

Para completar la información necesaria de los proyectos dígase, levantamiento de requisitos, aprobación del diseño, realización de pruebas funcionales, para dar soporte a los sistemas, despliegue de los sistemas desarrollados y chequeos del estado de ejecución, se precisa la transportación del personal que realiza estas tareas, hacia los lugares donde son requerido dichos proyectos.

Los líderes de proyecto son los encargados de realizar la solicitud del transporte a su Dirección del centro y esta a su vez a la Dirección de Transporte.

Actualmente la gestión de la solicitud de reserva se realiza a través de mensajes de correo electrónico enviando un documento *Excel* con toda la información referente a la transportación.

Como parte de las dificultades actuales ocurre que en ocasiones se les olvide a los líderes de proyecto realizar la reservación, o que los encargados de la Dirección de algún centro no se encuentren laborando para garantizar el transporte. Además, no se conoce el estado actual en que se encuentra la solicitud y no se notifica la aceptación o no de esta, provocando atrasos en el cronograma de ejecución de las tareas, pérdida de tiempo del personal que las realiza y el incumplimiento con el cliente al que va destinado el proyecto. No existe un registro con las solicitudes realizadas, ni de los centros con mayores necesidades de transportación, así como la cantidad de kilómetros recorridos, ni una planificación de las visitas en función de optimizar los recursos energéticos. Por lo que, cuando se realiza la planificación del transporte no se tienen en cuenta estos elementos.

Al enviarse la información referente a la reserva de transporte por vía correo electrónico, esto provoca que la bandeja de entrada del correo de quien atiende el proceso llegue al límite de su capacidad. Además, puede ocurrir que el correo no sea leído en tiempo por algunos de los involucrados, o que sea eliminado por error.

También puede ocurrir que después de realizada la reserva de transporte y llegado el día, se cancele la visita, debido a la disponibilidad de tiempo del cliente o por cambios en el cronograma de ejecución de las tareas, provocando que se disponga de esfuerzo y recursos de forma ineficiente, impidiendo la utilización del transporte para otros fines de gran importancia para la Universidad.

Toda la planificación referente a las rutas de transportación para poder agrupar por cercanía, se hace de forma manual. Esto implica que el encargado de esta tarea debe conocer las distancias a cada lugar; apoyándose por un mapa y la cantidad de medios de transporte disponible; lo que produce demora a la hora de realizar esta tarea y a veces se culmina con rutas no enfocadas a optimizar la cantidad combustible.

A partir de la situación problemática descrita anteriormente se plantea como **problema a resolver** ¿Cómo mejorar el proceso de reserva de transporte de los centros de producción en la UCI?

Para dar respuesta a ello se centra el **objeto de estudio** en el proceso de planificación de transporte; enmarcado como **campo de acción** el proceso de reserva de transporte de los centros de producción en la UCI. Con lo anteriormente expuesto y en la búsqueda de una solución al problema se declara como **objetivo general** del trabajo de diploma, desarrollar un sistema para mejorar el proceso de reserva de transporte de los centros de producción en la UCI.

Complementando el objetivo general se definen los siguientes **objetivos específicos:**

- Analizar los principios teóricos de la investigación, enfocados a sistemas para gestionar el proceso de reserva de transporte de los centros de producción en la UCI.
- Realizar análisis y diseño del sistema para gestionar el proceso de reserva de transporte de los centros de producción en la UCI.
- Implementar el sistema para gestionar el proceso de reserva de transporte de los centros de producción en la UCI.
- Validar el sistema mediante pruebas de *software* para comprobar su correcto funcionamiento.

Como **preguntas científicas** se establecen las siguientes:

1- ¿Cuáles son los antecedentes históricos del empleo las tecnologías para gestionar el proceso de reserva de transporte de los centros de producción en la UCI?
2- ¿Cuál es el estado actual del proceso de reserva de transporte de los centros de producción en la UCI?
3- ¿Qué características debe tener un sistema para gestionar el proceso de reserva de transporte de los centros de producción en la UCI?
4- ¿Qué resultados tendrá en la práctica un sistema para gestionar el proceso de reserva de transporte de los centros de producción en la UCI?

Los **métodos científicos** que se emplearon durante la investigación fueron:

Métodos teóricos

- Analítico-Sintético: se empleó en el estudio de la bibliografía, para identificar, analizar y seleccionar los conceptos y las definiciones más importantes relacionadas con el tema, permitiendo generar una propuesta adecuada a la situación planteada.

- Histórico-Lógico: se empleó en el análisis de la base teórica. Además, fue utilizado para el estudio de sistemas de gestión de reservación de transporte similares con la investigación.
- Modelación: Este método se utilizó para la realización de los artefactos correspondientes al análisis y diseño e implementación de las funcionalidades del sistema.

Métodos empíricos

- Entrevista: se utilizó para recopilar y analizar la información referente a la gestión del transporte, mediante la realización de preguntas a directivos y personal involucrado. Para identificar los procesos de reserva de transporte que se llevan a cabo actualmente en la UCI y las deficiencias que presentan; estableciendo criterios en vista al desarrollo del sistema. Las entrevistas fueron realizadas al jefe de piquera de transporte, a una selección de jefes de proyectos de la Facultad 1 y a la directora del Centro de Ideoinformática (CIDI).
- Observación: empleado en el análisis de los procedimientos que se realizan actualmente en la gestión de reservas de transporte, además posibilitó verificar la información obtenida a través de las entrevistas ejecutadas.

El contenido de este documento está estructurado en tres capítulos, conclusiones, recomendaciones, bibliografía y anexos.

Capítulo 1: Fundamentación teórica de la investigación. En este capítulo se expone un análisis de los sistemas actuales destinados a la gestión de la reserva de transporte. Se realiza una investigación sobre las herramientas, tecnologías y tendencias actuales, que sirven para dar soporte a este proceso. Se fundamentan los objetivos propuestos con el presente trabajo y finalmente se justifica la tecnología seleccionada, así como se analizan las fuentes y bibliografías utilizadas.

Capítulo 2: Análisis y diseño del sistema de gestión para la reserva de transporte de los centros de producción en la UCI. En este capítulo se realiza el análisis del estado actual del negocio para un mejor entendimiento del mismo y de esta manera lograr la informatización de todos los procesos involucrados en el negocio. Se hace una descripción de la solución que se propone, Se obtienen y describen los requisitos a desarrollar mediante las Historias de Usuario (HU), se define la arquitectura y patrones de diseño a utilizar para el desarrollo del sistema. Se representa la estructura del sistema por medio de los diagramas de clase del diseño. Se muestra el diseño de la base de datos a través del modelo de datos.

Capítulo 3: Implementación y validación del sistema de gestión para la reserva de transporte de los centros de producción en la UCI. En este capítulo tiene como objetivo la implementación de las diferentes funcionalidades, presentar los estándares de código, la elaboración de los diagramas de componentes y despliegue. Se describen las pruebas realizadas para la validación del sistema y se exponen los resultados obtenidos.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

1.1 Introducción

En el presente capítulo se abordan aspectos sobre los sistemas de gestión de reserva de transporte existentes tanto a nivel nacional como internacional. Se realiza un estudio de las principales tecnologías, metodologías y tendencias de desarrollo sobre las cuales se apoyará la propuesta de solución. Se abordan las herramientas seleccionadas para el análisis y diseño de la aplicación, con el fin de dar cumplimiento a los objetivos trazados.

1.2 Conceptos asociados a la investigación

1.2.1 Sistema

Proviene del latín systema, es un módulo ordenado de elementos que se encuentran interrelacionados y que interactúan entre sí (Definición de sistema, 2016).Otras fuentes de información lo describen como un conjunto de elementos relacionados entre sí y que funcionan como un todo (Significado de Sistema, 2015). Después de analizadas las definiciones anteriores se puede afirmar que un sistema es una relación que se establece entre varios elementos ordenados con características comunes y que funcionan como un todo.

1.2.2 Gestión

Procede del latín gestĭo, el concepto de gestión hace referencia a la acción y consecuencia de administrar o gestionar algo. Por otro lado, gestionar es llevar a cabo diligencias que hacen posible la realización de una operación comercial o de un anhelo cualquiera (Significado de Gestión, 2015). Analizando la definición anterior se puede afirmar que la gestión es la capacidad de una institución de planear, organizar y controlar de manera eficiente sus recursos (humanos y físicos).

1.2.3 Transporte

El término de transporte proviene de los vocablos del latín *trans*, “al otro lado”, y *portare*, “llevar”; es un medio de traslado de personas o mercancías de un lugar a otro, y está considerado como una actividad del sector terciario. El transporte permite el crecimiento económico y las posibilidades de desarrollo de una nación (Definición de transporte, 2016).

1.2.4 Sistema de Gestión de Transporte

También conocido como (*TM-System*) es una herramienta para los agentes de transporte. Con el sistema de gestión de transporte, el agente puede tramitar y trabajar las órdenes de transporte que recibe de la empresa cargadora o productora (El Diccionario de Transporte , 2016). Con ello, los viajes y rutas pueden planearse y organizarse mejor. De esta forma, podrá ver en cualquier momento la disponibilidad de los vehículos. Esta mejora el funcionamiento del área de transporte de las instituciones, ya sea de transporte de mercancías o el capital humano de la empresa.

1.2.5 Sistemas de Reservación Computarizado

Los Sistemas de Reservación Computarizado o *Computarized Reservation Systems* (CRS), son el primer paso de la inclusión de los computadores en los sistemas de reservación de transportes masivos de pasajeros. Estos sistemas se usaron para realizar seguimientos y mantener registros de las agendas de vuelo, reservaciones y asignación de asientos de los pasajeros, carga de la nave, inventario del vuelo, compra de tickets y tarifas de los pasajes principalmente en las aerolíneas de transporte de pasajeros (Travelport, 2014). Con el tiempo los CRS mejoraron, se les agregaron nuevas funcionalidades, surgiendo así los Sistemas de Distribución Global o *Global Distribution Systems* (GDS).

1.2.6 Sistemas de gestión de reservaciones

Son un medio para acceder a los servicios de una empresa o institución en general, durante un tiempo determinado con antelación a que este sea brindado. Las reservaciones son de gran importancia, pues reducen el riesgo de que el cliente no reciba el servicio y disminuyen el tiempo que este debe esperar para tener acceso al mismo (Naranjo, 2014).

1.3 Estudio del proceso para sistemas de reserva de transporte

1.3.1 Proceso de reservación de transporte en el ámbito internacional

Expedia

Expedia es un sitio *web* de reserva de viajes. Permite a sus usuarios realizar búsquedas de vuelos, hoteles y alquiler de coches con el fin de crear un paquete de viaje a medida, con tarifas competitivas. Los usuarios también pueden comprar vacaciones en crucero y otras actividades. Brinda la opción de hacer reservas *on-line* o por teléfono. Posee licencia propietaria. Ofrece información sobre el viaje para que el pasajero sepa

las posibilidades de reservación que tiene de forma general. Este sistema brinda la posibilidad de reservar viajes de ida y vuelta, desde un origen y hacia un destino predefinido en la aplicación. Además, brinda un mejor entendimiento acerca de la gestión de paradas intermedias, donde el pasajero podrá decidir e informar si desea quedarse en un sitio intermedio entre el origen y el destino del viaje (Expedia, 2014).

Logitravel

Logitravel es un sitio *web* de reserva de viajes por excelencia. Su interfaz sencilla y la gran cantidad de viajes y ofertas planteadas, hacen que sea uno de los primeros lugares a los que acuden los visitantes que buscan un destino vacacional. En Logitravel es posible buscar cruceros, circuitos, destinos en costa, destinos de ocio, vuelos, hoteles, ferris, trenes y paquetes. Este incluye ofertas de viaje, especialmente pensadas para públicos concretos, como es el caso de los viajes románticos o los que buscan parques de atracciones. La especialización y la gran cantidad de viajes ofrecidos convierten a este sitio *web* en punto de referencia para la búsqueda de viajes. Posee licencia propietaria y tecnologías que se dificulta su adquisición debido a que son privativas (Logitravel, 2014). Este *software* proporciona información acerca del histórico que se lleva de todos los datos manejados en el sistema.

Después de analizar los sistemas anteriores se puede destacar que son sistemas de licencia propietaria y que utilizan otras tecnologías privativas. Ofrecen un sin número de servicios y sus funcionalidades que no satisfacen las necesidades actuales de la UCI ya que están enfocadas al tipo de cliente y las especificaciones propias de su negocio.

1.3.2 Proceso de reservación de transporte en Cuba

Sistema de reservación en línea de Viazul

Este sistema surge producto al crecimiento y aceptación nacional de la empresa con el mismo nombre. Viazul ofrece transportación interprovincial a todo tipo de personal, extranjero o nacional. Con el desarrollo de este sistema las reservaciones a la empresa se elevaron a escala internacional. Viazul recibe cada día un aumento en la solicitud de sus servicios producto de esta aplicación, que ofrece los horarios de los ómnibus, así como diferentes servicios complementarios con microbuses, logrando así una variada oferta, basados en el mismo género de transportación. (Viazul, 2016).

El sistema muestra la posibilidad de escoger destinos ya predefinidos, lo cual puede ser tomado como referencia en el caso de la transportación general, permitiendo al trabajador encargado realizar las rutas de

las reservas de los centros de producción, la posibilidad de que escoja la opción de viaje más viable para cada pasajero, logrando así optimizar el consumo de combustible.

Por otro lado, está destinado a fines turísticos, por lo que contiene opciones específicas de este tipo de negocios que le permiten diferenciarse de otras empresas de transporte en el país. Un ejemplo de ello es que, para deleite del cliente, permite la inclusión de rutas turísticas en la programación de sus viajes. Características como estas evidentemente no son demandadas en la UCI y a la vez, también es un *software* propietario.

El sistema de reservación de Cubana de Aviación

Este sistema está dedicado exclusivamente a la reservación de billetes de viaje de avión. Brinda al usuario la posibilidad de seleccionar el origen y el destino de su viaje, la clase en la que desea viajar e igualmente la fecha. Las ventas directas comprenden la reserva y el pago, este último deberá efectuarse usando una tarjeta de crédito *Visa* o *Mastercard*, enviando notificación al pasajero vía correo electrónico si la transacción fue exitosa. Posee licencia propietaria (Cubana, 2017).

La utilización del sistema Cubana de Aviación en la UCI, no respondería a las necesidades de informatización del proceso de reserva de transporte de los centros de producción, pues los servicios que presta están en función del tipo de aeronave, de clase de pasajero en dicha aerolínea y de las especificidades de su negocio.

1.3.3 Proceso de reservación de transporte en la UCI

Sistema de Reservación de Pase Masivo

Es en el año 2007 que el primer sistema informático para gestionar las Transportaciones Nacionales de la UCI queda oficialmente puesto en uso. El Sistema de Reservación de Pase Masivo brindó soporte a los procesos de la Dirección de Transporte, incluyendo además una serie de funcionalidades de administración del sistema, que permitían gestionar toda la información con la que esta dirección trabaja. En todas sus acciones consultaba la información en una base de datos central que proporcionaba la información respecto a las personas que podrían reservar.

En el desarrollo de esa aplicación *Web* se utilizó PHP como lenguaje de programación del lado del servidor, PostgreSQL como sistema gestor de base de datos y Apache como servidor de aplicaciones *web*. Se usó JavaScript del lado del cliente para lograr la interactividad con el usuario en el navegador y específicamente

la técnica Ajax. La metodología RUP fue la seleccionada para planificar y controlar todo el proceso de desarrollo. Se usó además la arquitectura Cliente-Servidor (Rabilero, et al., 2016).

Luego de analizar este sistema, se llega a la conclusión de que era necesaria su renovación, debido a una serie de cambios que habían ocurrido durante el tiempo de explotación, pues se implementó teniendo en cuenta un grupo de características que tenía la Universidad en años anteriores y la modificación de estas dio al traste con el funcionamiento de la aplicación. Lo que trajo consigo que cada movilización de la institución, ya sea masiva o de profesores, requiriere de mucho tiempo para preparar las condiciones y tratar de arreglar los problemas que ocurren con el uso de la aplicación.

Sistema de Transportación Nacional para la UCI

Este sistema brindó sus servicios a partir del 2009 en la UCI para la gestión del transporte del personal en diferentes etapas del curso escolar. Está compuesto por un grupo de funcionalidades: gestionar reservaciones, definir bloques de entrada y salida, gestionar viajes, y distribuir el personal por transporte; aunque en reiteradas ocasiones esta actividad debió ser corregida manualmente. De este sistema se identificaron varias funcionalidades y características que serán objeto de un profundo análisis, debido a que deben ser tomadas en cuenta para la solución que se proponga. Ellas son la reservación y la distribución.

El Sistema de Transportación Nacional para la UCI (STN-UCI) es un sistema poco flexible en cuanto a la configuración de sus funcionalidades. Se construyó atado a características y necesidades que poseía la Universidad en el momento de su desarrollo. En consecuencia, el mismo contiene funcionalidades obsoletas, como la gestión del pase de fin de semana y la distribución por semestre del pase de profesores. A su vez, también requería nuevas funcionalidades como la distribución de las personas a viajar por ómnibus. El marco de trabajo usado en el desarrollo fue GUUD en su versión 1.0; integrado por los marcos de trabajo CodeIgniter y jQuery (Rabilero, et al., 2016).

Conclusión del estudio realizado a los sistemas gestores de reservaciones de transporte

Después de analizado los sistemas anteriormente descritos, se puede destacar que la mayoría de los estos poseen licencia propietaria y que utilizan otras tecnologías privativas. Ofrecen un sin número de servicios y funcionalidades que no satisfacen las necesidades actuales de los centros de producción y la Dirección de transporte de la UCI, ya que están enfocadas a satisfacer un tipo de cliente y las especificaciones para las cuales fueron diseñadas. Por lo que se evidencia la necesidad de desarrollar un nuevo sistema, que permita

mejorar el proceso de reserva de transporte de los centros de producción en la UCI. No obstante, se tendrán en cuenta para el nuevo desarrollo, las buenas prácticas que quedaron como resultado del estudio realizado como son la forma de realizar la reservación y distribución, la posibilidad de escoger destinos ya predefinidos y la accesibilidad a la información acerca del histórico, que se lleva de todos los datos manejados en el sistema.

A continuación, se presenta una tabla resumen donde se recogen los datos más significativos de cada uno de los sistemas analizados.

Tabla 1. Características generales de los sistemas estudiados (Elaboración propia).

Sistemas	Expedia	LOGITRAVEL.com	Viazul	Cubana	UCI Universidad de las Ciencias Informáticas
Tecnología	.Net	.Net	PHP	Laravel	Codelgniter
Licencia	Propietaria	Propietaria	Propietaria	Propietaria	Libre
Multiplataforma	No	No	No	No	Si
Servidor	Nginx	Nginx	Apache	Nginx	Apache

1.4 Metodologías de desarrollo de *software*

Las metodologías de desarrollo de *software* en Ingeniería de *Software* (ISW) son un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. Estas se dividen principalmente en 2 grupos: Las metodologías tradicionales de desarrollo como RUP y las metodologías ágiles como XP y AUP. Las metodologías de desarrollo surgen por la necesidad de la industria del *software* de agilizarse y robustecerse al mismo tiempo. Las primeras están pensadas para el uso exhaustivo de documentación durante todo el ciclo de vida del proyecto. Las segundas se centran en la capacidad de respuesta frente a los cambios, la confianza en las habilidades del equipo y en mantener una buena relación con el cliente.

Las metodologías tradicionales (formales) se focalizan en documentación, planificación y procesos. Además, centran su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto, definido todo esto, en la fase inicial del desarrollo del proyecto (Roberth G. Figueroa, 2016).

1.4.1 Metodología de desarrollo ágil

Las metodologías ágiles permiten incorporar cambios con rapidez en el desarrollo de *software*. En muchas ocasiones, los modelos de gestión tradicionales no sirven para afrontar un reto que hoy en día resulta fundamental: incorporar cambios con rapidez y en cualquier fase del proyecto. Estas se envuelven en un enfoque para la toma de decisiones en los proyectos de *software*, que se refiere a métodos de ingeniería del *software* basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto.

XP

Programación extrema, en inglés *Extreme Programming*, es la metodología más destacada de los procesos ágiles de desarrollo de *software*, se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Se basa en las pruebas realizadas a los principales procesos, de tal manera que, adelantándose en algo hacia el futuro, se puedan hacer pruebas de las fallas que pudieran ocurrir. Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores partícipes del proyecto trabajen en una misma estación de trabajo (Sánchez, 2015).

AUP

Constituye una versión simplificada del RUP, desarrollada por *Scott Ambler*. Esta metodología combina procesos propios del concepto unificado tradicional con técnicas ágiles, con el objetivo de mejorar la productividad. Permitiendo así describir de manera simple y fácil de entender la forma de desarrollar aplicaciones de *software* de negocio.

AUP aplica técnicas ágiles, entre las que se incluyen: (Sánchez, 2015)

- El desarrollo dirigido por pruebas.
- El modelado ágil.
- La gestión de cambios ágil.
- La refactorización de bases de datos para mejorar la productividad.

Variación de AUP para la UCI

Proceso ágil unificado, por sus siglas en inglés *Agil Unified Process*, en su versión UCI. Esta metodología es una variante realizada por la UCI a la metodología ágil AUP y está definida por la universidad como el documento rector de la actividad productiva.

Fases de Variación de AUP para la UCI

La metodología Variación de AUP para la UCI está forma por tres fases, (Inicio, Ejecución y Cierre) para el ciclo de vida de los proyectos de la universidad, las cuales contienen las características de las cuatro fases (Inicio, Elaboración, Construcción y Transición) propuestas en AUP. Las características de las fases de la metodología de la universidad son:

- ✓ **Inicio:** durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

- ✓ **Ejecución:** en esta fase se ejecutan las actividades requeridas para desarrollar el *software*, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, se obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto. Durante esta fase el *software* es transferido al ambiente de los usuarios finales o entregado al cliente junto con la documentación. Además, en esta transición se capacita a los usuarios finales sobre la utilización de la aplicación.

- ✓ **Cierre:** en el cierre se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto (Sánchez, 2015).

Como resultado de analizar las metodologías planteadas anteriormente se decidió utilizar AUP-UCI como metodología de desarrollo debido a que está definida por la universidad como el documento rector de la actividad productiva.

1.5 Sistema de gestión de contenidos (CMS)

CMS

Es un sistema de gestión de contenidos (CMS por sus siglas en inglés: *Content Management System*) para páginas *web*, aplicadas principalmente a la publicación. Su interfaz administrativa facilita la creación y publicación de contenidos, además de administrar otras funciones más avanzadas como: la modificación del estilo de la página o agregar funcionalidades extra a través de *plugins*, de una manera simple para facilitar las tareas a usuarios con conocimientos menos avanzados de programación.

Algunas funcionalidades típicas de un CMS son:

- Administración de la estructura del portal: módulos, menús, diseño, configuración general.
- Administración del contenido: distintos tipos de temas, su gestión y publicación.
- Administración de usuarios: políticas de gestión de usuarios y de acceso a los contenidos mediante roles y permisos.
- Informes y gestión del portal: errores, estadísticas de acceso.

Joomla

Joomla es un CMS de código abierto con gran cantidad de temas y aplicaciones. Se basa en un sistema combinado de PHP y MySQL, cuenta con su propia comunidad de usuarios y desarrolladores que contribuyen al desarrollo del proyecto, ya sea trabajando en la mejora del *software* o aportando nuevos módulos para incrementar o mejorar sus funcionalidades. Presenta una curva de aprendizaje elevada. Permite desarrollar sitios *web* dinámicos e interactivos, la generación de código HTML bien formado, la gestión de blogs, vistas de impresión de artículos, la creación de foros y *polls* (encuestas), calendarios, búsquedas integradas al sitio y soporte multi-idioma (Joomla!, 2017).

Drupal

Drupal es un CMS que se distribuye como *software* libre bajo licencia GNU GPL (*General Public License*) versión 2 o superior. El *software* está desarrollado con el lenguaje de programación PHP y utiliza una base de datos MySQL. Está maquetado con hojas de estilo CSS, con lo que es posible construir sitios *web* totalmente accesibles (Gil, 2012). En Drupal es posible desarrollar cualquier tipo de portal o aplicación *web* debido a que es un CMS genérico. Además de las funcionalidades básicas que vienen integradas en el

software, es posible añadir nuevas funcionalidades a través de módulos. Cuenta con una amplia comunidad de usuarios y desarrolladores.

Es modular, multipropósito y muy configurable ya que permite publicar artículos, imágenes y servicios añadidos como foros, encuestas, votaciones, blogs, administración de usuarios y permisos. En la actualidad se encuentra entre las plataformas de uso más frecuente en la construcción de sitios *web* y una de las más fáciles de desplegar debido a sus requerimientos mínimos.

WordPress

WordPress es un sistema de gestión de contenidos (CMS) gratuito y de código abierto*,* por medio del uso de PHP y MySQL, sujeto a la licencia GPLv2. Actualmente hace funcionar 31% de los principales 10 millones de sitios *web* de Internet. La usabilidad, extensibilidad y madura comunidad de desarrollo de *WordPress* hace que sea una elección segura para *web* de todos los tamaños. Es un *software* diseñado para todos, enfatizando en la accesibilidad, rendimiento y facilidad de uso. De igual manera que Joomla y Drupal dispone de gran cantidad de temas y *plugins* tanto gratuitos como de pago (Wordpress, 2018) .

1.5.2 Conclusión sobre los CMS

Se ha optado por un CMS debido a la necesidad de que la *web* sea administrable y que su contenido debe poder gestionarse posteriormente de manera fácil y sencilla. Después de analizar los CMS anteriores se decidió utilizar Drupal en su versión 7.67, basándose en que posee un gran número de temas, aplicaciones y funcionalidades, lo que permite la agilización en el proceso de desarrollo de *software* ya que tributa al aprovechamiento y reutilización del código, creado por otros programadores.

Cuenta con una amplia comunidad de desarrolladores y diseñadores, encargados de crear complementos y plantillas. Permite seguir un flujo de trabajo de forma sencilla. Este CMS ha demostrado ser una solución segura y fuerte debido a que el equipo de seguridad de Drupal publica regularmente recomendaciones de seguridad que describen sus vulnerabilidades e indican soluciones en forma de parches, versiones actualizadas o instrucciones de mitigación. Además, los conocimientos recientemente adquiridos durante las prácticas curriculares fueron en Drupal 7.

1.6 Sistema de gestión de base de datos

MySQL 5.0.12

Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario, utiliza licencia de tipo GPL. Es muy utilizado en aplicaciones *web*, en plataformas (Linux/*Windows*-Apache-MySQL-PHP/*Perl*/*Python*), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad está muy ligada a PHP, que a menudo aparece en combinación con MySQL. El principal objetivo de MySQL es velocidad y robustez. Soporta gran cantidad de tipos de datos para las columnas. Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo. Flexible sistema de contraseñas (*passwords*) y gestión de usuarios, con un muy buen nivel de seguridad en los datos. El servidor soporta mensajes de error en distintas lenguas (MySQL, 2018).

Ventajas de MySQL:

- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos, no solos las basadas en *Windows*.
- Requiere menos recursos de *hardware*.
- Es multiplataforma, puede ser usado sin necesidad de pago bajo los términos de la Licencia Publica General de GNU.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos

Se decide utilizar MySQL debido a las ventajas presentadas anteriormente y por ser el SGBD que utiliza Drupal 7 para el manejo de su base datos.

1.7 Lenguaje del lado del servidor

PHP 7.0.2

PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor* y está publicado bajo la *PHP License.* Es un lenguaje de programación, interpretado, diseñado originalmente para la creación de páginas *web* dinámicas. Se ejecuta en un servidor, que toma el código PHP como su entrada y generando HTML como salida. Puede ser desplegado en todos los sistemas operativos que contengan un intérprete PHP compatible. Posee soporte para la mayoría de los gestores de bases de datos, comúnmente usados (PHP,

2018). Todas estas ventajas hacen de PHP el lenguaje a seleccionar del lado del servidor para el desarrollo de la solución, además de ser el lenguaje utilizado por Drupal 7.

1.8 Tecnologías y herramientas

HTML 5

HTML, sigla en inglés de *HyperText Markup Language* (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas *web*. Es un estándar que sirve de referencia del *software* que conecta con la elaboración de páginas *web* en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página *web*, como texto, imágenes, videos, juegos, entre otros (html-MDN, 2018).

CSS 3

Las hojas de estilo en cascada (en inglés *Cascading Style Sheets*, CSS) constituyen un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).CSS permite el control centralizado de la presentación de un sitio *web* completo con lo que se agiliza de forma considerable la actualización del mismo. Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio *web*, con lo que aumenta considerablemente la accesibilidad (W3C, 2018).

JavaScript

JavaScript es un lenguaje de programación liguero e interpretado, se define como orientado a objetos y basado en prototipos. Es un lenguaje script multi-paradigma y se utiliza principalmente en su forma del lado del cliente (*client-side*), implementado como parte de un navegador *web* permitiendo mejoras en la interfaz de usuario y páginas *web* dinámicas (Sánchez, 2016).

Bootstrap 4

Mediante la combinación de CSS y JavaScript, este *framework* multiplataforma, simplifica el proceso de creación de diseños *web*. Es un *framework* que posee numerosos componentes y permite un ahorro significativo de esfuerzo y tiempo, debido a que facilita la creación de interfaces con diseño *web* adaptativo o *responsive*. Ofrece un conjunto de plantillas CSS y ficheros JavaScript que permiten la integración del

framework de forma sencilla con las aplicaciones *web*. Por otra parte, es capaz de integrarse con las principales librerías *JavaScript*, por ejemplo, jQuery (Bootstrap, 2017).

Visual Paradigm 8.0

Visual Paradigm 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. Ayuda a una rápida construcción de aplicaciones con calidad y a un menor costo. Permite crear todos los tipos de diagramas de clases, ingeniería inversa, generar código desde diagramas y generar documentación (Visual Paradigm, 2017).

IDE de desarrollo

***NetBeans* 8.1**

NetBeans es un IDE de desarrollo muy utilizado para desarrollar aplicaciones de *software* con *Symfony*, Drupal, *WordPress*, Laravel, entre otros marcos. Es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento. Soporta varios lenguajes, tales como: PHP y todos los relacionados con la *web* (HTML, CSS, JavaScript y XML). Se caracteriza por su portabilidad ya que es compatible con la mayoría de los sistemas operativos como Windows, Mac, GNU/Linux y Solaris. De modo que se convierte en una herramienta muy útil para el desarrollo *web* (NetBeans, 2016).

Herramientas de pruebas

Acunetix

Acunetix es una herramienta automatizada de pruebas de seguridad de aplicaciones *web*. Actualmente Acunetix comprueba más de 500 tipos diferentes de vulnerabilidades. Puede escanear cualquier sitio *web* que es accesible a través del protocolo HTTP / HTTPS, básicamente, si el sitio *web* se puede ver en un navegador, Acunetix puede escanearlo. También proporciona herramientas de pruebas de penetración manuales que aumentan y contribuyen a las pruebas automatizadas, así como ayudar con la prueba de vulnerabilidades lógicas. Al ser una solución online no hay necesidad de hospedar el *software* y por lo tanto no hay necesidad de extra *hardware* (Acunetix, 2018).

JMeter

JMeter es una herramienta Java desarrollada dentro del proyecto Jakarta, que permite realizar Pruebas de Rendimiento y Pruebas Funcionales sobre aplicaciones *web*. JMeter permite realizar pruebas *web* clásicas, pero también permite realizar test de FTP, JDBC, JNDI, LDAP, SOAP/XML-RPC, y *WebServices* (en Beta). Permite la ejecución de pruebas distribuidas entre distintos ordenadores, para realizar pruebas de rendimiento. Además, activar o desactivar una parte del test, lo que es muy útil cuando se está desarrollando un test largo, y se desea deshabilitar ciertas partes iniciales que sean muy pesadas o largas. Tiene la forma de generar un caso de prueba a través de una navegación de usuario. Fácil correlación mediante la capacidad de extraer datos de formatos de respuesta más populares, HTML, JSON, XML o cualquier formato textual, portabilidad completa y almacenamiento en caché (Jmeter, 2018).

1.9 Conclusiones parciales

A partir del estudio realizado a los sistemas de reservación de transporte existentes tanto a nivel nacional e internacional, se llega a la conclusión de que ninguno de estos sistemas son ideales a la hora de implementarlos como solución en la UCI, por lo que se hace necesaria la creación de un nuevo sistema que responda a las especificaciones puntuales propias de los centros de producción. Se logró fundamentar las bases teóricas de la investigación y se analizaron los conceptos básicos relacionados al objeto de estudio definido. Se seleccionaron todas las herramientas, tecnologías, metodología y lenguajes de programación a utilizar, para dar solución a la problemática.

CAPÍTULO 2. ANÁLISIS Y DISEÑO DEL SISTEMA DE GESTIÓN PARA LA RESERVA DE TRANSPORTE DE LOS CENTROS DE PRODUCCIÓN EN LA UCI

2.1 Introducción

En este capítulo se brinda una propuesta general del sistema a desarrollar. Se realiza la descripción de los requisitos y del diseño del sistema que se propone para dar respuesta a los objetivos planteados. Se presentan los principales resultados del análisis y el diseño del sistema mediante diagramas UML.

2.2 Descripción de la propuesta de solución

Para darle solución al problema de la presente investigación, se propone el desarrollo de un sistema que permita mejorar el proceso de reserva de transporte de los centros de producción en la UCI. Para ello este sistema debe gestionar las solicitudes de reserva de transporte generadas por los líderes de proyectos, además debe gestionar las rutas creadas por los encargados de la dirección de transporte y notificar a los líderes de proyecto si fue aceptada o no la solicitud. Debe emitir distintos reportes para llevar un historial donde se pueda consultar la información por medio de filtros de búsqueda. También brindará la posibilidad a los encargados de la dirección de transporte escoger rutas ya predefinidas anteriormente, con el fin de optimizar el consumo de combustible.

2.3 Especificación de requisitos

Es el proceso de determinar, documentar y gestionar las necesidades y los requisitos de los interesados para cumplir con los objetivos del proyecto. El beneficio clave de este proceso es que proporciona la base para definir y gestionar el alcance del proyecto, incluyendo el alcance del producto (PMBOK, 2013). En correspondencia a lo antes planteado, en la presente investigación se utilizan solo dos tipos de requisito: requisitos funcionales y no funcionales, detallados a continuación.

2.4 Requisitos funcionales (RF)

Son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas (SOMMERVILLE, 2011). El objetivo principal de los requisitos funcionales es identificar y documentar lo que se espera del sistema, de una forma clara y comprensible para el cliente y los miembros del equipo de desarrollo. Para el sistema a implementar se identificaron las siguientes funcionalidades:

Tabla 2. Requisitos funcionales del sistema (Elaboración propia).

Requisitos	
RF1 : Autenticar usuario	**RF19 :** Editar rol
RF2 : Cancelar usuario	**RF20 :** Eliminar rol
RF3 : Mostrar usuario	**RF21 :** Crear ruta
RF4 : Listar usuario	**RF22 :** Editar ruta
RF5 : Bloquear usuario	**RF23 :** Eliminar ruta
RF6 : Desbloquear usuario	**RF24 :** Mostrar ruta
RF7 : Configurar LDAP	**RF25 :** Listar de ruta
RF8 : Crear solicitud	**RF26 :** Exportar listado de rutas
RF9 : Editar solicitud	**RF27 :** Mostrar reporte de rutas por chofer
RF10 : Eliminar solicitud	**RF28 :** Exportar reporte de rutas por chofer
RF11 : Mostrar solicitud	**RF29 :** Mostrar reporte de rutas por vehículo
RF12 : Listar de solicitud	**RF30 :** Exportar reporte de rutas por vehículo
RF13 : Exportar listado de solicitudes	**RF31 :** Notificar por correo la asignación de rutas
RF14 : Mostrar reporte de solicitudes por fecha	**RF32 :** Crear taxonomía centro
RF15 : Exportar reporte de solicitudes por fecha	**RF33 :** Editar taxonomía centro
RF16 : Mostrar reporte de solicitudes por centro productivo	**RF34 :** Eliminar taxonomía centro
RF17 : Exportar reporte de solicitudes por centro productivo	**RF35 :** Listar términos de taxonomía centro

RF18 : Crear rol	

2.5 Requisitos no funcionales (RNF)

Son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema. Son requerimientos que no se relacionan directamente con los servicios específicos que el sistema entrega a sus usuarios (SOMMERVILLE, 2011). Existen múltiples categorías para clasificar los requisitos no funcionales, para el sistema se definieron los siguientes RNF:

Tabla 3. Requisitos no funcionales del sistema (Elaboración propia).

Categoría	Requisitos	Descripción de los Requisitos
Interfaz	**RNF1**	El sitio deberá mostrarse correctamente en navegadores como Mozilla Firefox versión superior a la 30, Chrome versión superior a la 31 y Opera 11 o superior.
Usabilidad	**RNF2**	El sistema debe garantizar una interfaz amigable e intuitiva para asegurar la fácil navegación del usuario por la misma.
	RNF3	El sistema de gestión reserva de transporte de los centros de producción en la UCI debe ser una aplicación *web*
Portabilidad	**RNF4**	El *software* estará construido con código totalmente portable para ser ejecutado en sistemas operativos (Windows y GNU/Linux).

Diseño e implementación	**RNF5**	Se utilizará Visual Paradigm como herramienta de modelado.
	RNF6	Sistema de Gestión de Contenidos (CMS): El sistema se desarrollará sobre Drupal en su versión 7.67.
	RNF7	Entorno de desarrollo integrado (IDE): El *software* se desarrollará sobre NetBeans 8.1.
Seguridad	**RNF8**	Se garantizará la integridad y confidencialidad de la información mediante mecanismos de control de accesos no autorizados propios de Drupal 7.67.
Legales	**RNF9**	La plataforma y herramientas utilizadas para el desarrollo del sistema deben poseer licencia GNU GPL (*General Public License*).
Hardware	**RNF10**	Para la ejecución del sistema se requiere que la *PC* cliente tenga los siguientes componentes de *hardware*: Pentium 4 o superior, 512 MB RAM como mínimo. Se recomienda para el servidor una memoria RAM de 2 GB.
	RNF11	La comunicación entre la *PC* cliente y el servidor de aplicaciones *web* se realiza a través del protocolo HTTPS.
Eficiencia	**RNF12**	El tiempo de demora de una petición al servidor debe ser menor de (4) segundos aproximadamente.

Software	**RNF13**	Para el despliegue del sistema se debe contar con un servidor de bases de datos con MySQL versión 5.0.12.

2.6 Historias de Usuario

Las historias de usuarios son breves descripciones textuales de la funcionalidad requerida. Describen al interesado que se beneficia con la característica (rol), aquello que el interesado necesita lograr (objetivo) y el beneficio para el interesado (motivación). Las historias de usuarios se utilizan a menudo con métodos ágiles (PMBOK, 2013).

En la metodología AUP-UCI en el escenario 4, las historias de usuarios son una de las formas de describir los requisitos del sistema. Se describen con un lenguaje claro y natural, para que puedan ser comprendidas por el resto del equipo que las vaya a utilizar. Este artefacto sirve como guía para las futuras pruebas de aceptación. Para realizar las historias de usuarios se hace necesario definir los siguientes parámetros.

La prioridad en el negocio:

- **Alta:** Cuando son consideradas por los clientes esenciales para el funcionamiento del negocio.
- **Media:** Cuando el cliente cree que son necesarias, pero estas no intervienen en gran medida en el desarrollo del negocio.
- **Baja:** Cuando constituyen procesos que se deben tener en cuenta, pero su ausencia no perjudica el flujo principal del negocio.

El riesgo en desarrollo:

- **Alto:** Cuando en la implementación de las HU pueden surgir errores que lleven a la inoperatividad el código.
- **Medio:** Cuando en la implementación de las HU pueden existir errores que retrasen la entrega del producto.
- **Bajo:** Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

A continuación, se presentan historias de usuario de mayor importancia, que describen los requisitos necesarios para el sistema.

Tabla 4. Historia de usuario para el requisito funcional “Crear solicitud” (Elaboración propia).

Historia de Usuario	
Numero:HU_8	**Nombre Historia de Usuario:** Crear solicitud
Programador: Darian Aguila Vega	**Iteración Asignada:**1
Prioridad: Alta	**Tiempo Estimado:** 1 día
Riesgo en Desarrollo: Alta	**Tiempo Real:** 1
Descripción: Los usuarios autenticados pueden crear nuevas solicitudes en el sistema, para ello debe llenar los siguientes campos: • **Nombre de jefe de proyecto:** Campo de texto, obligatorio. Admite todos los caracteres. Longitud máxima 20 caracteres. • **Proyecto:** Campo de texto, obligatorio. Admite todos los caracteres. Longitud máxima 30 caracteres. • **Centro:** Campo de selección obligatorio. Valores (ej: FORTES, CEGEL, entre otros). • **Fecha:** Calendario desplegable, obligatorio. Formato DD/MM/AAAA. • **Hora Ida:** Campo de texto, obligatorio. Debe comenzar con los caracteres, (0-9) separado por (:), debe terminar con (0-9) y (am, pm), ej:(10:30 am).	

• **Hora Regreso:** Campo de texto, obligatorio. Debe comenzar con los caracteres, (0-9) separado por (:), debe terminar con (0-9) y (am, pm), ej:(10:30 am).

• **Cantidad de Personas:** Campo de selección, obligatorio. Valores (números de 0-20). Valor por defecto (0).

• **Dirección:** Campo de texto, obligatorio. Long min 5, máximo 60 caracteres. Permiten los caracteres: (a-z, A-Z, 0-9). Debe comenzar con (a-z, A-Z). Permite direcciones estructuralmente válidas (Estructura: Nombre lugar: Calle_principal No. numero e/ calle1 y calle2, Municipio. ej Casa: 198 No. 3440 e/ 324 y 327, Cerro).

• **Estado:** Campo de selección, Oculto solo es visible cuando es editada la solicitud. Valores (pendiente, aceptada, rechazada), Valor por defecto (pendiente).

Observaciones:

1. Si el usuario introduce la información de forma correcta, el sistema emite un mensaje notificando que se ha creado satisfactoriamente el usuario.

2. Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error.

3. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben de llenarse.

Prototipo elemental de interfaz gráfica de usuario: No aplica.

Tabla 5. Historia de usuario para el requisito funcional “Editar solicitud” (Elaboración propia).

Historia de Usuario	
Numero:HU_9	**Nombre Historia de Usuario:** Editar solicitud
Programador: Darian Aguila Vega	**Iteración Asignada:**1
Prioridad: Alta	**Tiempo Estimado:** 1 día
Riesgo en Desarrollo: Alta	**Tiempo Real:** 1
Descripción: Los usuarios autenticados pueden editar sus solicitudes y los usuarios con rol (*webmaster* o dirección de transporte) pueden editar todas las solicitudes existentes en el sistema, para ello debe editar los siguientes campos: • **Nombre de jefe de proyecto:** Campo de texto, obligatorio. Admite todos los caracteres. Longitud máxima 20 caracteres. • **Proyecto:** Campo de texto, obligatorio. Admite todos los caracteres. Longitud máxima 30 caracteres • **Centro:** Campo de selección obligatorio. Valores (ej: FORTES, CEGEL, entre otros). • **Fecha:** Calendario desplegable, obligatorio. Formato DD/MM/AAAA. • **Hora Ida:** Campo de texto, obligatorio. Debe comenzar con los caracteres, (0-9) separado por (:), debe terminar con (0-9) y (am, pm), ej:(10:30 am).	

• **Hora Regreso:** Campo de texto, obligatorio. Debe comenzar con los caracteres, (0-9) separado por (:), debe terminar con (0-9) y (am, pm), ej:(10:30 am).

• **Cantidad de Personas:** Campo de selección, obligatorio. Valores (números de 0-20). Valor por defecto (0).

• **Dirección:** Campo de texto, obligatorio. Long min 5, máximo 60 caracteres. Permiten los caracteres: (a-z, A-Z, 0-9). Debe comenzar con (a-z, A-Z). Permite direcciones estructuralmente válidas (Estructura: Nombre lugar: Calle principal No. numero e/ calle1 y calle2, Municipio. ej Casa: 198 No. 3440 e/ 324 y 327, Cerro).

• **Estado:** Campo de selección, editable solo para los usuarios con rol (*webmaster* o dirección de transporte). Valores (pendiente, aceptada, rechazada), Valor por defecto (pendiente).

Observaciones:

1. Si el usuario introduce la información de forma correcta, el sistema emite un mensaje notificando que se ha creado satisfactoriamente el usuario.

2. Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error.

3. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben de llenarse.

Prototipo elemental de interfaz gráfica de usuario: No aplica.

Tabla 6. Historia de usuario para el requisito funcional “Listar solicitud” (Elaboración propia).

Historia de Usuario	
Numero:HU_12	**Nombre Historia de Usuario:** Listar solicitud
Programador: Darian Aguila Vega	**Iteración Asignada:**1
Prioridad: Media	**Tiempo Estimado:** 1 días
Riesgo en Desarrollo: Alta	**Tiempo Real:** 1
Descripción: Los usuarios que tengan rol jefe de proyecto pueden listar sus solicitudes existentes en el sistema y los otros roles (*webmaster* y dirección de transporte) pueden listar todas las solicitudes.	
Observaciones: 1. Si no existen solicitudes en el sistema el listado se mostrará vacío.	
Prototipo elemental de interfaz gráfica de usuario: No aplica.	

2.7 Arquitectura

El sistema a implementar, al utilizar el CMS Drupal hereda su arquitectura. Drupal presenta una arquitectura n-capas que permite ampliar sus funcionalidades mediante la integración de nuevos módulos. Drupal estructura los contenidos en una serie de elementos básicos. Estos son los nodos (*nodes*), módulos (*modules*), bloques y menús (*blocks* & *menus*), permisos de usuario y plantillas (*templates*).

Los “Nodos” son los elementos básicos en que Drupal almacena la información, los contenidos. Así a medida que el sitio *web* crece, lo va haciendo el número de nodos los cuales van formando un “depósito y nodos” cada vez mayor. Se puede decir que la primera capa de la estructura de Drupal la forma este “depósito” de nodos (Jardinot González, 2015).

Los “Módulos” son los elementos que operan sobre los nodos y otorgan funcionalidad a Drupal permitiendo incrementar sus capacidades o adaptarlas a las necesidades de cada sitio *web*. Son como *Plugins* que se instalan en el sitio *web* proporcionándole nuevas funcionalidades (Jardinot González, 2015).

La siguiente capa de Drupal la constituyen los “Bloques y menús”. Estos permiten estructurar y organizar los contenidos en la página *web.* Es decir que son los elementos que albergan y permiten acceder al usuario a la salida generada y procesada por los módulos a partir de la información almacenada en los nodos (Jardinot González, 2015).

La siguiente capa importante en Drupal es la de “Control de usuarios y permisos”. Actualmente, la mayor parte de sitios *web* son multiusuario, por lo que la seguridad y control de los usuarios es un punto clave para garantizar la integridad de la información almacenada. Con esta finalidad Drupal dispone de un registro de usuarios y de roles que permiten especificar que tareas pueden realizar y a que contenidos puede acceder cada tipo de usuario. Es decir que las operaciones que se pueden realizar sobre los elementos provenientes de las capas inferiores (lectura, modificación, creación…) se encuentran limitadas por la capa de control de usuarios y permisos de Drupal (Jardinot González, 2015).

La última capa, es la capa de “Plantillas” y es la que establece la apariencia gráfica o estilo de la información que se le muestra al usuario. Esta separación entre información y aspecto gráfico permite cambiar el diseño u apariencia del sitio *web* sin necesidad de modificar los contenidos, lo que es muy práctico si lo único que queremos es renovar la apariencia de un sitio *web* (Jardinot González, 2015).

La lógica de Drupal está programada en PHP, siguiendo un modelo de programación estructurada que hace uso de un sistema de bases de datos relacional. El núcleo del sistema se complementa con algunos módulos obligatorios y otros opcionales, que vienen dentro de la distribución de Drupal. El sistema incluye un conjunto de herramientas y pautas que se deben seguir para desarrollar e integrar nuevas funcionalidades a través de módulos adicionales (Pina, 2016).

El sistema Drupal se encarga de ejecutar los módulos solo cuando se necesitan, y para ello es necesario que cada módulo tenga un medio de comunicarle qué hace y cuándo debe hacerse. Esto se realiza mediante un conjunto de funciones llamadas *hooks* o ganchos. Cuando Drupal atiende una determinada solicitud, examina los módulos activados en el sitio buscando funciones cuyos nombres se ajustan a determinados *hooks.* Los *hooks* van a permitir incorporar simples y complejas funcionalidades al sistema Drupal (Pina, 2016).

En siguiente figura se muestran de forma esquemática los elementos que conforman un sistema Drupal.

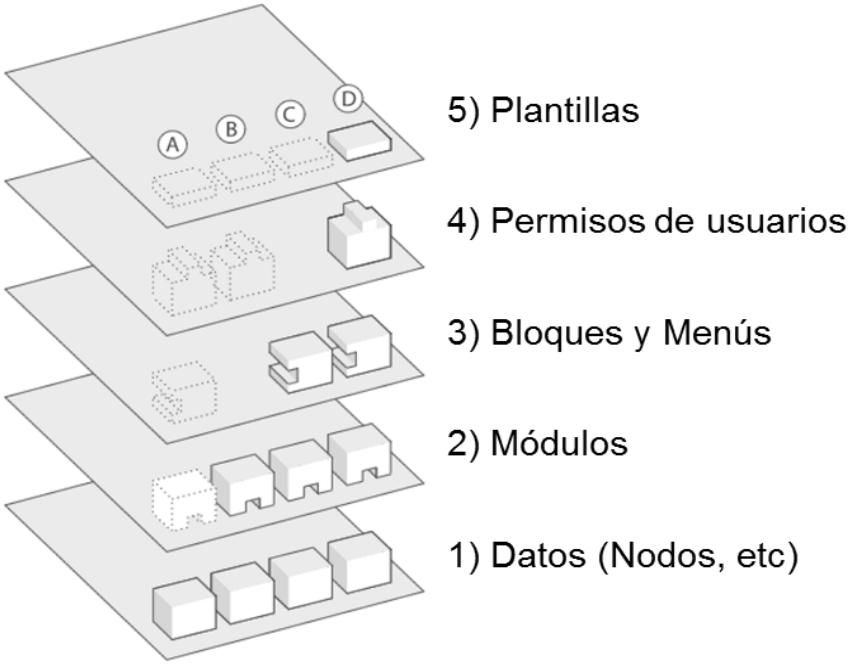


Figura 1. Arquitectura de Drupal (Jardinot González, 2015).

El sistema utiliza una arquitectura n-capas siguiendo un modelo de programación estructurada, que permite ampliar sus funcionalidades a través de métodos de desarrollo e integración de módulos adicionales, consiguiendo de esta forma incorporar nuevas características funcionales.

2.8 Patrones de diseño en Drupal

En el desarrollo de *software* el uso de patrones de diseño constituye una buena práctica, proporcionando una organización estructurada entendible por cualquier programador. El uso de patrones permite construir *software* que sean más fáciles de mantener y extender, representando un ahorro de tiempo (Gama, et al., 2014). En el diseño del sistema se tuvieron en cuenta los patrones de *software* para la Asignación General de Responsabilidad o GRASP (*General Responsabilty Assignment Software Patterns*) y Banda de los Cuatro o GOF (*Gang of Four*). A continuación, se describen los principales patrones empleados en el desarrollo de la propuesta de solución:

2.8.1 *Observer* (Observador)

Los módulos que implementan un *hook* determinado por evento de inserción o actualización de una determinada entidad, son declarados como observadores de dichas entidades con las que interactúan. En el sistema se implementa el *hook* edición_de_solicitud_*node_update*, que se ejecuta al editar una solicitud en el sitio, evidenciando así el patrón observador ya que se encuentra a la espera de un evento de edición, para ejecutar la acción de notificar a los líderes de proyecto las modificaciones realizadas a su solicitud.

2.8.2 *Chain of Responsibility* (Cadena de responsabilidades)

Este patrón en la POO (Programación Orientada a Objeto) proporciona a más de un objeto la capacidad de atender una petición, para así evitar el acoplamiento con el objeto que hace la petición. Se forma entre los objetos una cadena, en la cual cada objeto o satisface la petición o la pasa al siguiente. En el sistema de menús de Drupal se puede identificar el patrón de cadena de responsabilidades. En cada solicitud de la página, el menú del sistema determina si hay un módulo para gestionar la solicitud y si el usuario tiene acceso a los recursos solicitados. Para ello, el mensaje se pasa a la opción del menú correspondiente a la solicitud, si el elemento del menú no puede manejar la petición, se pasa a otro. Esto continúa hasta que un módulo se encarga de la petición, niega el acceso para el usuario, o la cadena se ha agotado (Marimón, et al., 2013).

2.8.3 *Singleton* (Instancia única)

La esencia de este patrón en la POO consiste en tener una sola instancia de un objeto disponible para toda la aplicación que la contiene. Provee una instancia global permitiendo que otros objetos accedan a esta única instancia. En Drupal se puede identificar este patrón si se piensan los módulos como objetos, los cuáles pueden ser pensados como una clase con una única instancia. En general lo que diferencia un módulo en Drupal de otro es el conjunto de funciones que este contiene, garantizando así la existencia de una única instancia para un módulo y la creación de un mecanismo de acceso global único a dicho módulo (Marimón, et al., 2013).

2.8.4 *Decorator* (Decorador)

Permite añadir responsabilidades extra que se pueden añadir y quitar a objetos concretos de manera dinámica y transparente, esto es, sin afectar a otros objetos. Además, proporciona una alternativa flexible a

la herencia para extender funcionalidad. Se aplica cuando la herencia sea impracticable, porque implique crear múltiples subclases para todas las combinaciones posibles (Guerra, et al., 2016).

La evidencia de este patrón se encuentra en la definición de *hooks* por parte de los módulos del núcleo de Drupal y otros módulos contribuidos, que posibilitan que otros puedan extender el comportamiento de dichas funciones. De esta forma se brinda la flexibilidad de que nuevos módulos puedan modificar el comportamiento del núcleo en cuanto al tratamiento de los datos y en cada uno de los eventos del sistema.

2.9 Modelo de Diseño

2.9.1 Diagrama de Clases de Diseño

Los Diagramas de Clases de Diseño permiten modelar clases, incluidos sus atributos, operaciones, relaciones y asociaciones con otras clases. Aporta una visión de la estructura del sistema (Pressman, 2010). Los diagramas mostrados a continuación con su descripción correspondiente, permiten visualizar las clases y relaciones que genera el CMS Drupal en conjunto con las creadas para la propuesta de solución. Permitiendo alcanzar un mayor grado de comprensión del sistema a desarrollar.

Descripción del Diagrama de Clases de Diseño “Crear solicitud”

La página servidora SP_Crear_Solicitud construye la página cliente CP_Crear_Solicitud, la cual se compone por el formulario FR_Crear_Solicitud que permite al usuario insertar la información necesaria para crear una solicitud. Después de ser enviados esto datos por medio del formulario, son procesados por la página servidora y se almacenan en la base de datos. Por último, se muestra al usuario la página CP_Mostrar_Solicitud.

El paquete CMS Drupal almacena un conjunto de ficheros indispensables para el funcionamiento del CMS como son (módulos, plantillas, JavaScript, CSS).

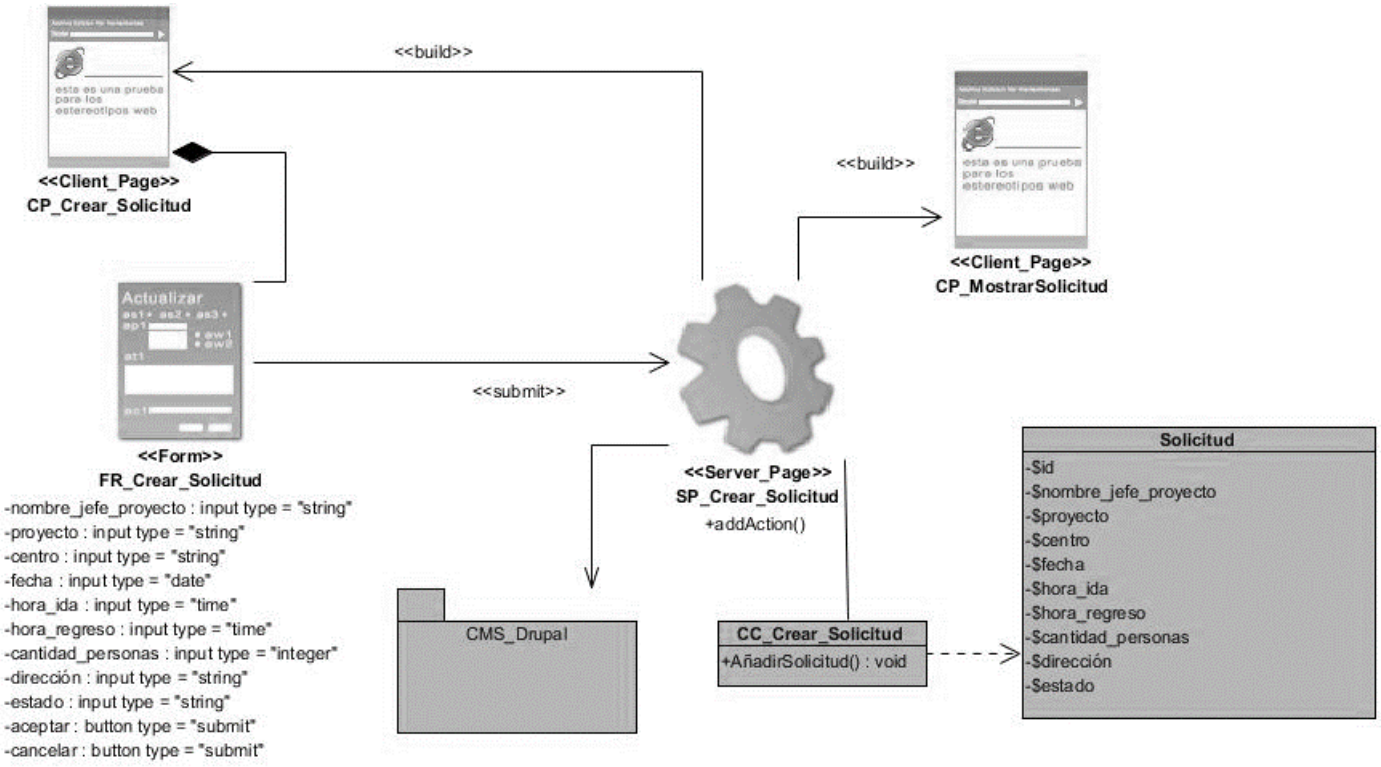


Figura 2. Diagrama de Clases de Diseño para el requisito funcional “Crear solicitud” (Elaboración propia).

Descripción del Diagrama de Clases de Diseño “Editar solicitud”

La página servidora SP_Editar_Solicitud construye la página cliente CP_Crear_Solicitud. A través de la página cliente CP_Listar_Solicitud, la cual se compone por el formulario FR_Listar_Solicitud se puede acceder mediante un enlace a la página cliente CP_Editar_Solicitud, la cual se compone por el formulario FR_Editar_Solicitud que permite al usuario editar la información referente a la solicitud. Después de ser enviados esto datos por medio del formulario, son procesados por la página servidora y se almacenan en la base de datos. Por último, se muestra al usuario la página CP_Mostrar_Solicitud.

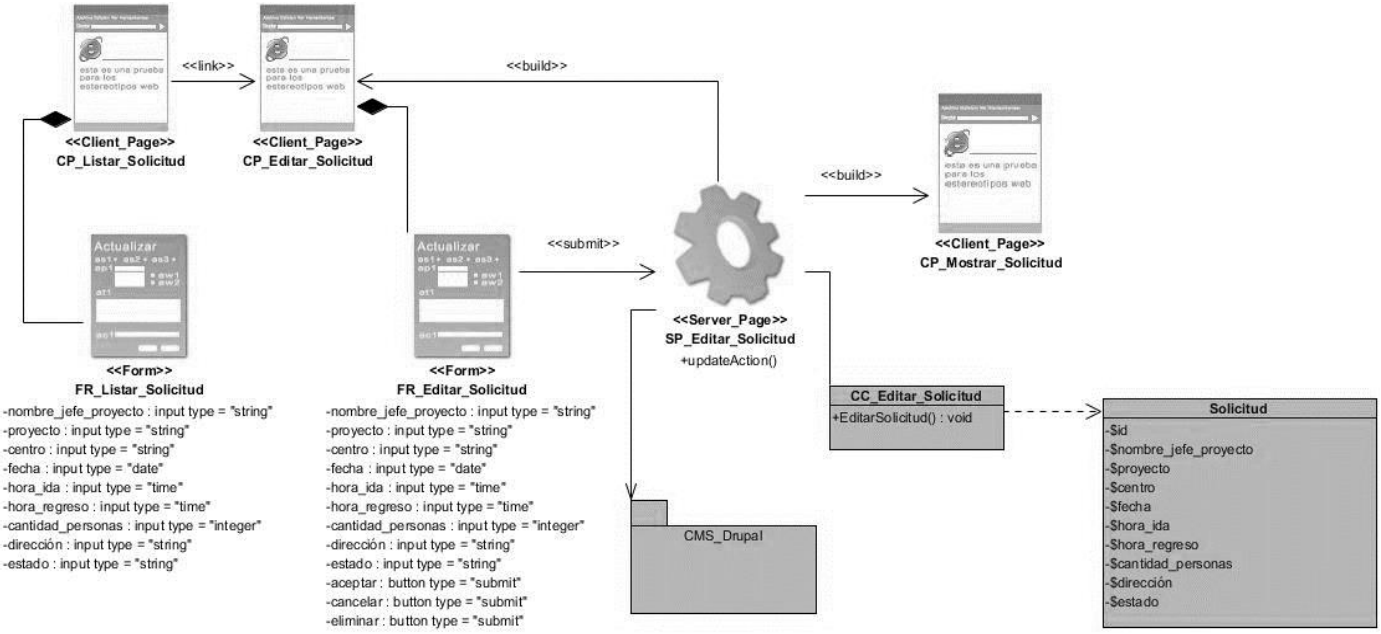


Figura 3. Diagrama de Clases de Diseño para el requisito funcional “Editar solicitud” (Elaboración propia).

Descripción del Diagrama de Clases de Diseño “Listar solicitud”

La página servidora SP_Listar_Solicitud construye la página cliente CP_Listar_Solicitud, la cual se compone por el formulario FR_Listar_Solicitud que permite al usuario listar las solicitudes existentes en el sistema.

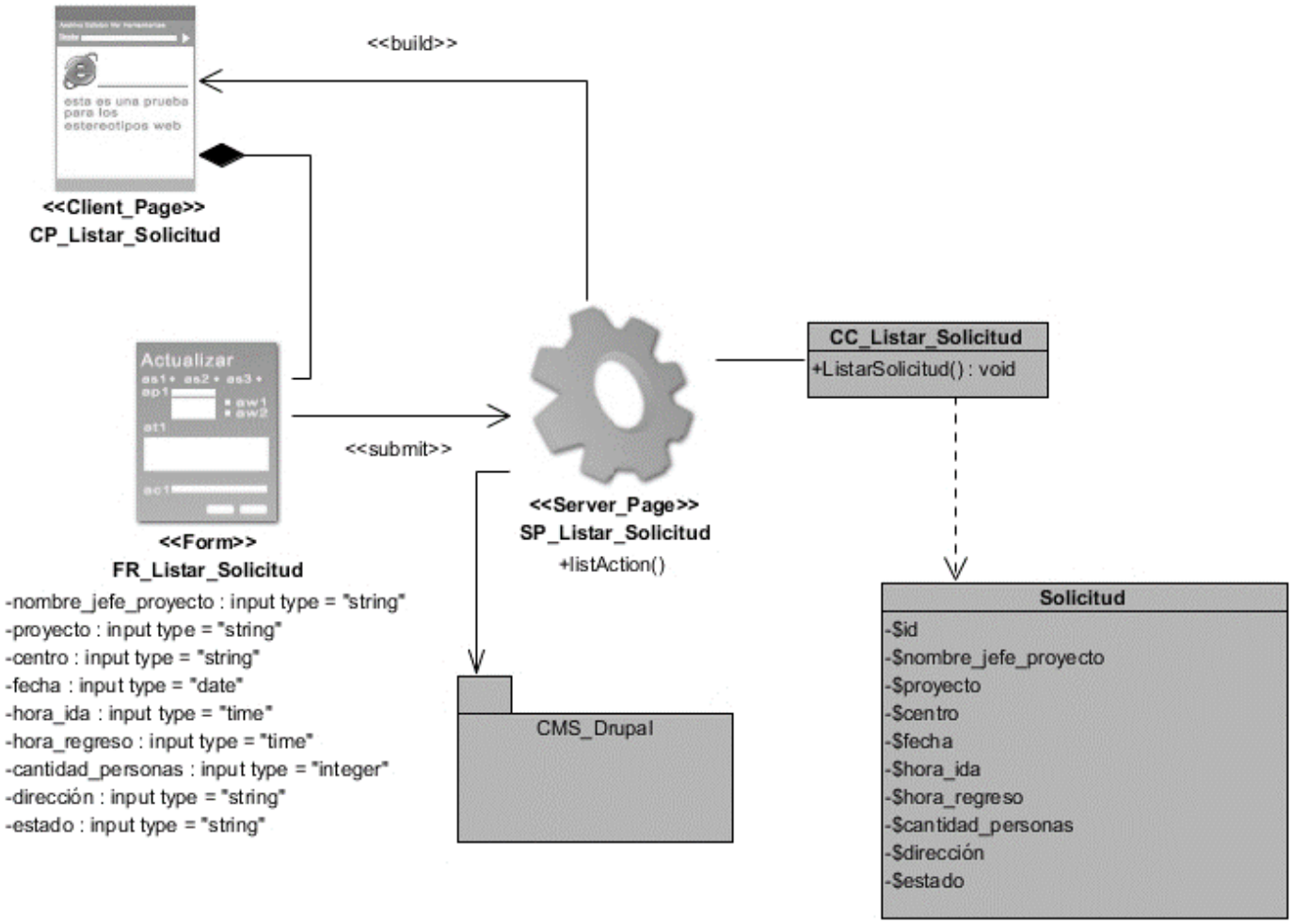


Figura 4. Diagrama de Clases de Diseño para el requisito funcional “Listar solicitud” (Elaboración propia).

2.9.2 Diagrama de secuencia

Los diagramas de secuencia se usan principalmente para modelar las interacciones entre los actores y los objetos en un sistema, así como las interacciones entre los objetos en sí. Muestra la sucesión de interacciones que ocurre durante un caso de uso particular o una instancia de caso de uso (SOMMERVILLE, 2011). A continuación, se muestran los diagramas de secuencia de mayor relevancia empleados en el desarrollo de la propuesta de solución.

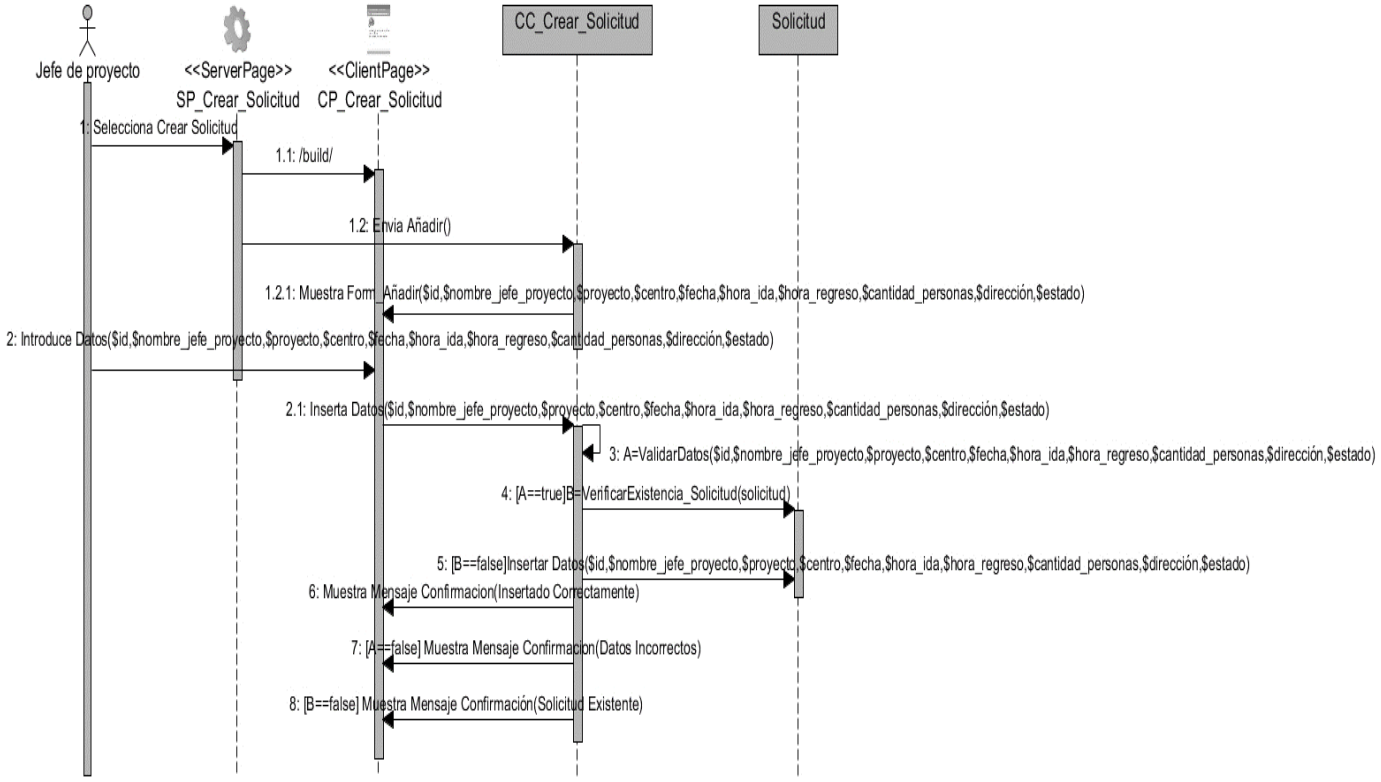


Figura 5. Diagrama de Secuencia para el requisito funcional “Crear solicitud” (Elaboración propia).

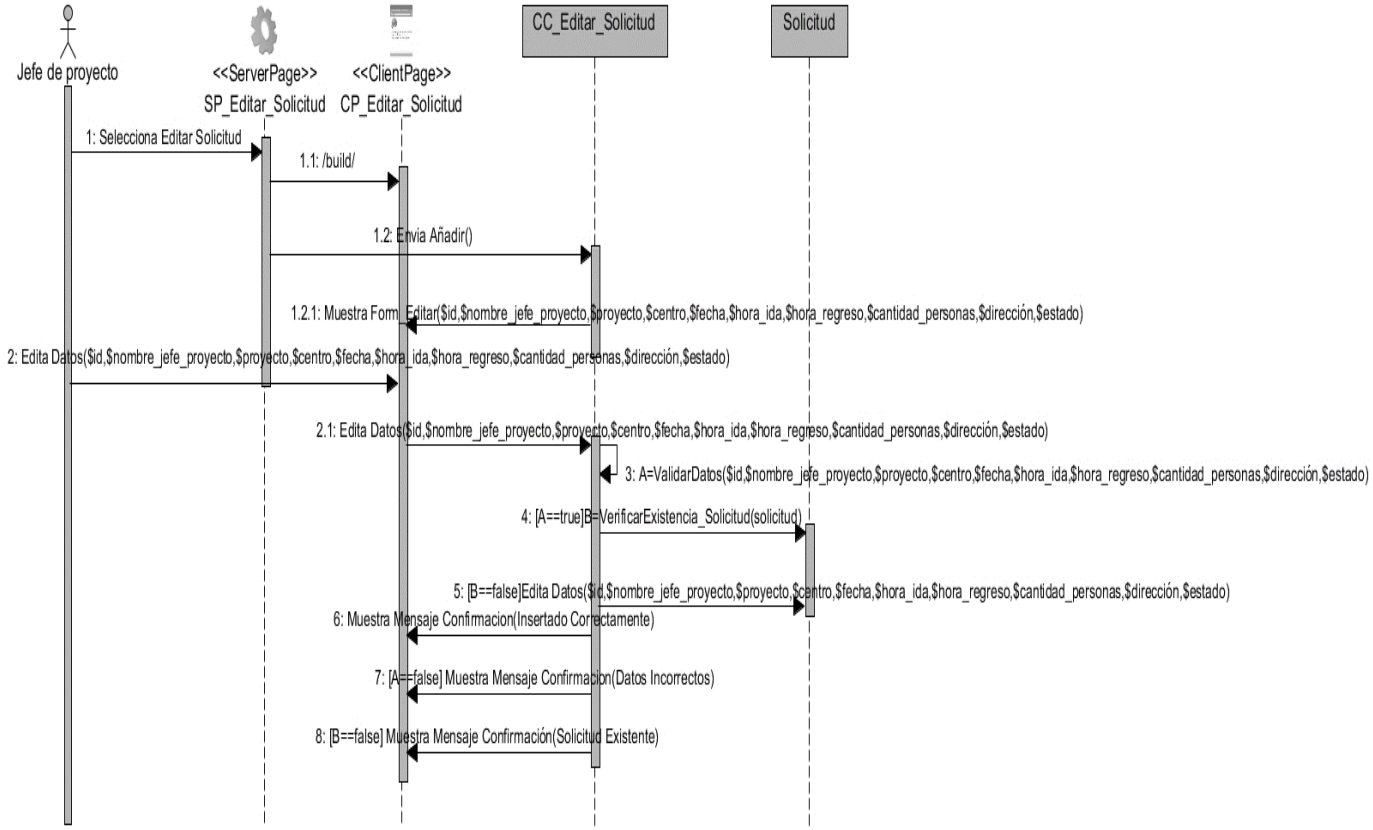


Figura 6. Diagrama de Secuencia para el requisito funcional “Editar solicitud” (Elaboración propia).

2.10 Modelo de datos

El Modelo de datos constituye el principal elemento de diseño de bases de datos que consiste en representar: objetos (entidades que existen y que se manipulan), atributos (características básicas de estos objetos) y relaciones (forma en que se enlazan los distintos objetos entre sí). Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos (Marqués, et al., 2015).

A continuación, se muestra el diseño de la base de datos del sistema a través del modelo de datos.

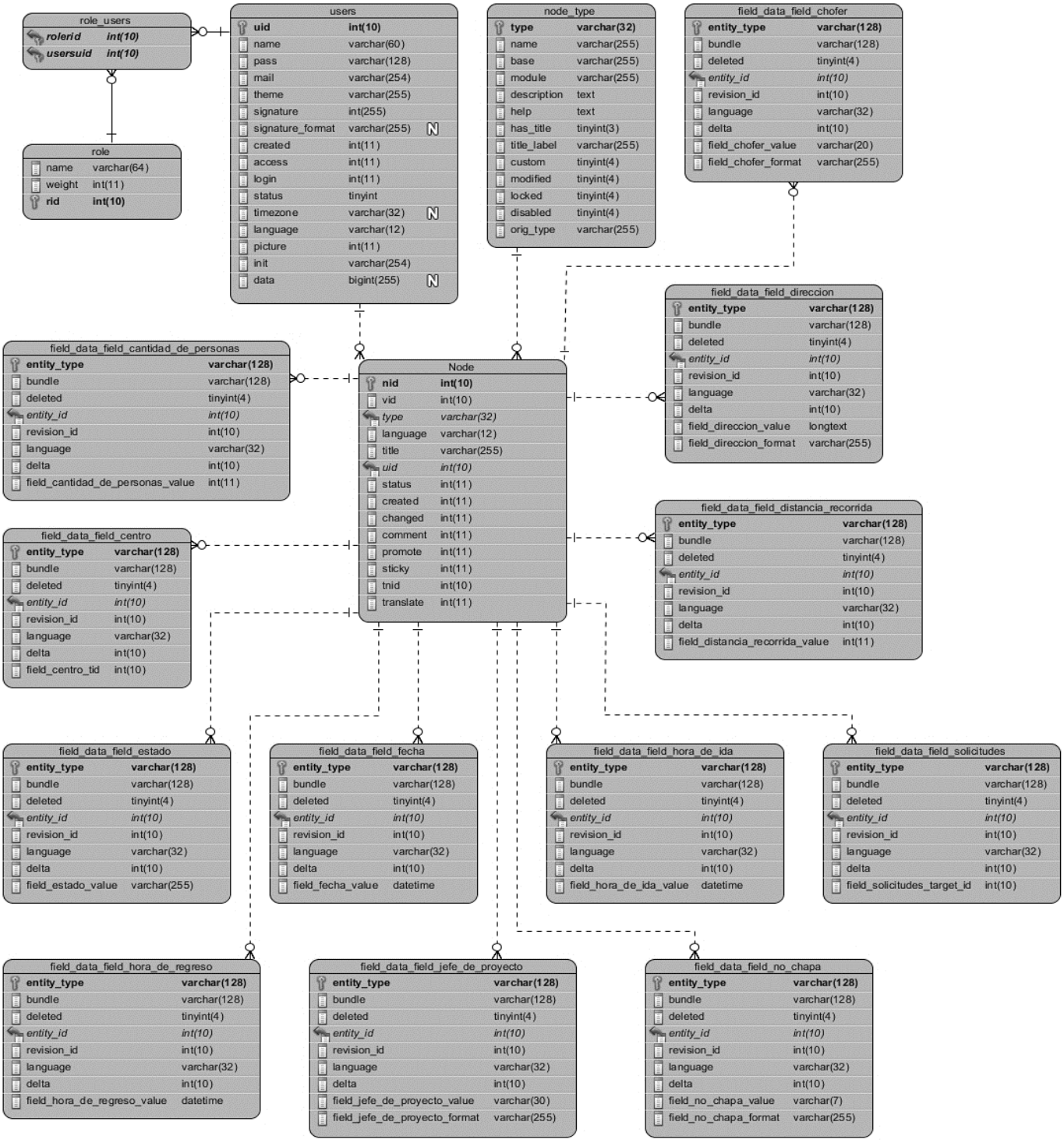


Figura 7. Diagrama del modelo de datos (Elaboración propia).

Descripción del Modelo de Datos

En el diagrama anterior se muestra la relación entre el esquema de Drupal y las tablas del sistema que fueron incorporadas a este. Desde la liberación de la versión 7 de Drupal, cada tipo de contenido creado o generado se almacena en la tabla node_type (como son las rutas y las solicitudes) y cada atributo de este, genera dos tablas: field_data y field_revision (son los valores resultantes de la revisión a los datos almacenados en la tabla node_type). Por esta razón se generan las tablas field_data_field_**campo** (Ej: field_data_field_centro) que almacenan los datos correspondientes a los atributos de cada contenido. La tabla users almacena los datos, registro y autenticación de los usuarios, se relaciona directamente con la tabla role que almacena la información referente a permisos de acceso a las diferentes funcionalidades.

2.11 Conclusiones parciales

En el presente capítulo se describió la propuesta de solución del sistema a partir de las deficiencias identificadas en el proceso de reserva de transporte de los centros de producción de la UCI. Se identificaron los requisitos funcionales y no funcionales con que deberá cumplir el sistema. Se describieron las Historias de Usuarios, lo cual proporcionó conocimiento de la prioridad y el tiempo que conllevará la implementación de cada requisito. A través de la arquitectura y el uso de los patrones de diseño definidos, se garantizó una mayor organización, reutilización de funciones y un código más legible. La elaboración de los diagramas de secuencia y clases de diseño con estereotipos *web* permitieron definir las relaciones entre las clases del *software,* así como las funcionalidades y atributos que deben presentar cada una de estas, logrando así un enfoque más detallado de la estructura y funcionamiento del sistema. Con la realización del modelo de datos se hizo posible materializar la visión de la base de datos a utilizar para el desarrollo del sistema.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA DEL SISTEMA DE GESTIÓN PARA LA RESERVA DE TRANSPORTE DE LOS CENTROS DE PRODUCCIÓN EN LA UCI

3.1 Introducción

En este capítulo se describen los aspectos referentes a las fases de implementación y pruebas de la propuesta de solución, se expone el diagrama de componentes, el cual describe como los elementos del modelo de diseño que se implementan en términos de componentes, se presenta el diagrama de despliegue permitiendo conocer el *hardware* y *software* necesario para el despliegue del sistema. Además, son realizadas las pruebas de *software* para comprobar el correcto funcionamiento del sistema y para detectar y corregir las posibles no conformidades.

3.2 Diagrama de despliegue

El diagrama de despliegue se utiliza para mostrar la estructura física del sistema, incluyendo las relaciones entre el *hardware* y el *software* que se despliega, estas relaciones son representadas por los protocolos de comunicación que se utilizan para acceder a cada uno. En la siguiente figura puede visualizarse el diagrama de despliegue definido para la solución propuesta:

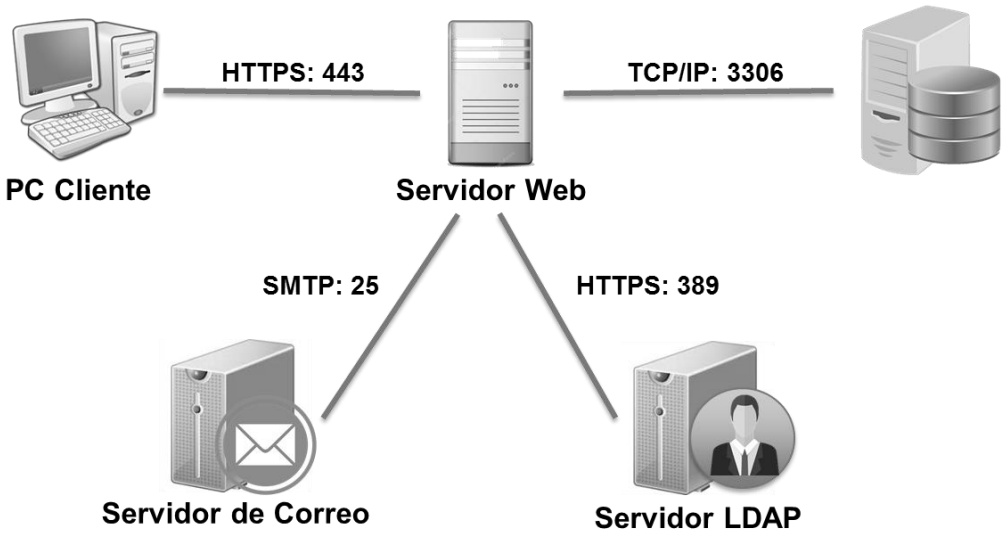


Figura 8. Modelo de despliegue (Elaboración propia).

El diagrama de despliegue representado muestra la siguiente distribución:

Cliente: computadora cliente capaz de conectarse al sistema a través de un navegador *web*, realizando peticiones al Servidor *Web*, mediante el protocolo de comunicaciones HTTPS y el puerto 443.

Servidor *web*: computadora en que se encuentra el servidor *web* Apache, lugar donde estará alojado el sistema. El mismo establecerá comunicación con los ordenadores clientes mediante protocolo HTTPS y con el servidor de base de datos por medio del protocolo TCP/IP y el puerto 3306 (Stallings, 2014).

Servidor de base de datos: ordenador en que se encuentra el gestor de base de datos MySQL capaz de mantener persistente la información generada.

Servidor de correo: servidor encargado del envío de correos electrónicos, el cual se comunica través del protocolo SMTP y el puerto 25.

LDAP: (Protocolo Ligero/Simplificado de Acceso a Directorio) es un protocolo de tipo cliente-servidor para acceder a un servicio de directorio. Los servicios de directorio facilitan el acceso a información organizada a una gran variedad de aplicaciones. Estos servicios le permiten a los usuarios y aplicaciones autorizadas buscar información de personas, computadoras, aplicaciones y dispositivos red. se comunica través del protocolo HTTPS y el puerto 389 (Stallings, 2014).

HTTPS: protocolo de transferencia de hipertexto seguro, por sus siglas en inglés, *Hypertex*t *Transfer Secure Protocol* (HTTPS), es un protocolo de red basado en HTTP por lo que está orientado a transacciones sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores y sigue el esquema petición-respuesta entre un cliente y un servidor (Stallings, 2014).

TCP/IP: base de Internet y sirve para enlazar computadoras que utilizan diferentes sistemas operativos. Familia de protocolos utilizada para la conexión entre el servidor *web* y el servidor donde se encuentra ubicada la base de datos (Stallings, 2014).

SMTP: (Protocolo para Transferencia Simple de Correo) es un protocolo de red utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos (Stallings, 2014).

3.3 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus dependencias. Determina la organización lógica de la implementación del sistema, mostrando todos los tipos de elementos de *software* que entran en la fabricación de las aplicaciones informáticas, estas pueden ser paquetes, bibliotecas cargadas dinámicamente o simples archivos, entre otros (SOMMERVILLE, 2011). A continuación se presenta el diagrama de componentes para la solución propuesta.

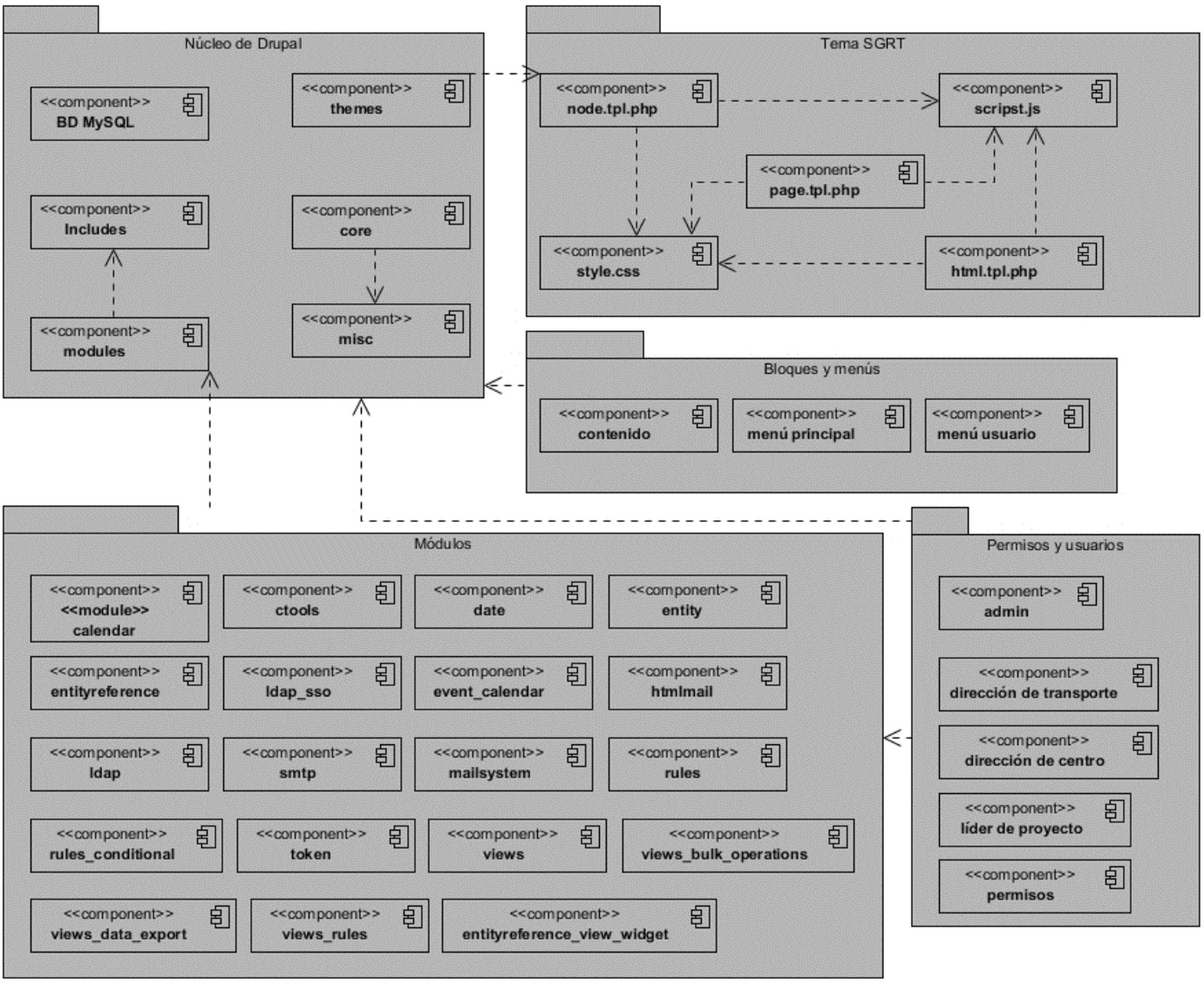


Figura 9. Diagrama de componentes (Elaboración propia).

A continuación, se describe cada componente representado en el diagrama anterior:

Núcleo de Drupal: Contiene los principales componentes básicos con los que cuenta Drupal, los cuales se crean desde que se instala este.

***Themes*:** Capa de presentación y diseño de Drupal.

***Modules*:** Aquí se encuentran todos los módulos de Drupal.

Bloques y menús: Es donde se encuentran los elementos que permiten estructurar y organizar los contenidos del sistema.

Permisos y usuarios: Contiene un registro de usuarios y de roles que permiten especificar que tareas pueden realizar y a que contenidos puede acceder cada tipo de usuario, garantizando así la seguridad del sistema.

Includes: Contiene un conjunto de librerías en forma de archivos PHP con extensión .inc, que incluyen funciones comunes del sistema (ajax.inc, batch.inc, cache.inc, date.inc, form.inc).

Core: Es el punto de inicio del sistema, a partir de esta entrada se invocan los diferentes módulos del CMS.

BD MySQL: Es la base de datos del sistema.

Misc: Incluye archivos CSS y JavaScript para el diseño y funcionamiento de Drupal.

Tema Sistema de Gestión de Reserva de Transporte (SGRT): Tema diseñado y desarrollado para el sistema.

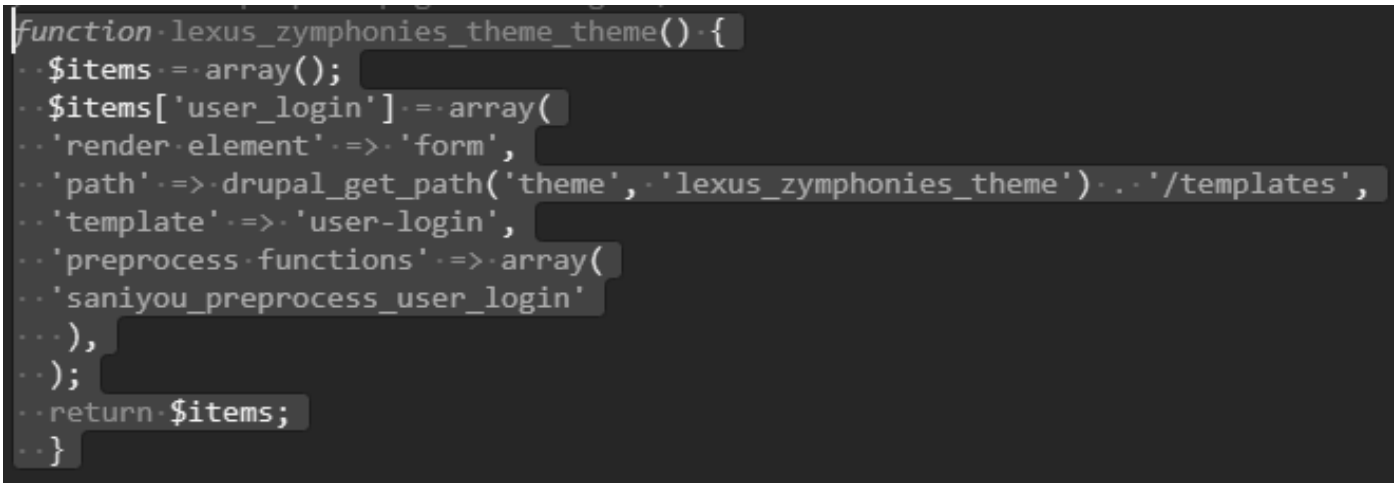
Módulos: Contiene todos los módulos que fueron instalados y configurados para el sistema.

3.4 Estándar de codificación

Los estándares de código resultan importantes en cualquier proyecto de desarrollo, ya que determinan normas de estilos, orden y estructura del lenguaje. Los estándares de código ayudan a asegurar que el código tenga una alta calidad, menos errores, y pueda ser mantenido fácilmente. A continuación, se ofrecen una serie de estándares para la codificación PHP en Drupal.

3.4.1 Indentación

La indentación consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. En programación se utiliza la indentación para anidar elementos. En Drupal debemos indentar con 2 espacios, nunca con tabuladores. Además, no se debe dejar espacios en blanco al final de cada línea (Gil, 2012). En la siguiente figura se puede apreciar un fragmento de código el cual evidencia el uso de la indentación del código.



```
function lexus_zymphonies_theme_theme() {
  $items = array();
  $items['user_login'] = array(
  'render element' => 'form',
  'path' => drupal_get_path('theme', 'lexus_zymphonies_theme') . '/templates',
  'template' => 'user-login',
  'preprocess functions' => array(
  'saniyou_preprocess_user_login'
    ),
  );
  return $items;
  }
```

Figura 10. Ejemplo de uso de indentación (Elaboración propia).

3.4.2 Etiquetas de apertura y cierre de PHP

Cuando estemos escribiendo en PHP, siempre se deben utilizar las etiquetas **<?php y ?>**, y en ningún caso la versión corta <? y ?>. En general se omite la etiqueta de cierre de PHP (?>) al final de los archivos (**.module** y **.inc)** (Gil, 2012). En la siguiente figura se puede apreciar un fragmento de código el cual evidencia el uso de las etiquetas de apertura de PHP.

```
<?php
/**
 * @file
 * Default theme implementation to display a node.
 */
?>
<?php if (!$page): ?>
  <article id="node-<?php print $node->nid; ?>" class="<?php print $classes; ?>
  clearfix"<?php print $attributes; ?>>
<?php endif; ?>
    <?php if (!$page): ?>
      <header>
  <?php endif; ?>
```

Figura 11. Ejemplo del uso de etiquetas de apertura y cierre de PHP (Elaboración propia).

3.4.3 Operadores

Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio. Por ejemplo, $logged_in == " ", en lugar de $logged_in==" ". Esto se aplica a operadores como +, -, *, /, =, ==, !=, >, <, . (concatenación de cadenas), .=, +=, -=, etc (Gil, 2012). En la siguiente figura se puede apreciar un fragmento de código el cual evidencia el uso de los operadores.

```
<div class="menu-navigation-container">
    <?php $main_menu_tree = menu_tree(variable_get('menu_main_links_source', 'main-menu'));
        if (!$logged_in == "") {?>
         <div class="<?php print $rl; ?>">   <?php
            print drupal_render($main_menu_tree);?>
         </div>
        <?php }?>
</div>
```

Figura 12. Ejemplo del uso de operadores (Elaboración propia).

3.4.4 Uso de comillas

Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres. Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto. Por ejemplo, "<?php print $front_page; ?>" .También se recomienda el uso de comillas

dobles cuando el texto puede incluir alguna comilla simple (Gil, 2012). En la siguiente figura se puede apreciar un fragmento de código el cual evidencia el uso de las comillas.

```
<h1 id="site-title">
    <a href="<?php print $front_page; ?>" title="<?php print t('Home'); ?>">
        <?php print $site_name; ?></a>
    <div id="site-description"><?php print $site_slogan; ?></div>
</h1>
```

Figura 13. Ejemplo del uso de comillas (Elaboración propia).

3.4.5 Uso de punto y coma (;) en código PHP

Aunque PHP permite escribir líneas de código individuales sin el terminador de línea (;), como por ejemplo <?php print $user->name ?>. En Drupal es siempre obligatorio: <?php print $user->name; ?> (Gil, 2012). En la siguiente figura se puede apreciar un fragmento de código el cual evidencia el uso de punto y coma (;) en código PHP.

```
<div class="content user-sesion" >
    <span> <?php if($user->name){ print $user->name;} ?></span>
    <ul class="menu"><a class="enlace-sesion" href="/sgrt/user/logout"
    title=""><i class="fa fa-sign-out "></i> Cerrar sesión</a>
    </ul>
</div>
```

Figura 14. Ejemplo del uso de punto y coma en código PHP (Elaboración propia).

3.4.6 Funciones

Los nombres de las funciones deben estar escritos en minúsculas y las palabras separadas por guion bajo. Además, se debe incluir siempre como prefijo el nombre del módulo, tema, etc., para evitar así duplicidad de funciones. En su declaración, después del nombre de la función, el paréntesis de inicio de los argumentos debe ir sin espacio. Cada argumento debe ir separado por un espacio, después de la coma del argumento anterior (Ej: function: lexus_zymphonies_theme_theme()) (Gil, 2012). En la siguiente figura se puede apreciar un fragmento de código el cual evidencia el uso de funciones.

```php
function lexus_zymphonies_theme_theme()
{
    $items = array();
    // create custom user-login.tpl.php
    $items['user_login'] = array(
        'render element' => 'form',
        'path' => drupal_get_path('theme', 'lexus_zymphonies_theme') . '/templates',
        'template' => 'user-login',
        'preprocess functions' => array(
            'saniyou_preprocess_user_login'
        ),
    );
}
```

Figura 15. Ejemplo del uso de funciones (Elaboración propia).

3.4.7 Nombres de módulos

Como norma general, el nombre de un módulo nunca debería incluir guiones bajos, aunque se componga de varias palabras. De esta forma será más fácil identificar el módulo al que pertenece una función, ya que el prefijo o nombre del módulo será todo aquello que esté antes del primer guion bajo. Por ejemplo, es aconsejable utilizar mimodulo en lugar de mi_modulo. Esta regla no es obligatoria, y es muy común encontrar, entre los módulos contribuidos, nombres conteniendo guiones bajos (Gil, 2012).

3.4.8 Comentar el código

Los comentarios suelen escribirse al principio de un archivo o de cada función y se utilizan para generar documentación de ayuda a través de aplicaciones que extraen la información a partir de las etiquetas empleadas. En el primero de los casos se pueden utilizar las etiquetas /* */ para comentarios en varias líneas y // para comentarios de una única línea. Se deben escribir frases completas, comenzándolas con mayúscula y terminándolas con un punto. En caso de que en el comentario se haga referencia a una constante, ésta deberá escribirse en mayúsculas (Ejemplo TRUE o FALSE) (Gil, 2012). En la siguiente figura se puede apreciar un fragmento de código comentado.

```php
<?php
/*
Comprueba si hay un usuario logeado
*/
if (!$logged_in && current_path() != "user") {
    //Para redireccionar a la paguina de login
    drupal_goto('login');
} ?>
```

Figura 16. Ejemplo de comentario de código (Elaboración propia).

3.5 Pruebas de *software*

Luego de implementado el sistema se hace necesario realizar un conjunto de pruebas que permitan certificar la calidad del sistema. Las pruebas descubren los defectos en el programa antes de usarlo. Permiten demostrar al desarrollador y al cliente que el *software* cumple con los requerimientos. A través de ellas se encuentran situaciones donde el comportamiento del *software* sea incorrecto, indeseable o no esté de acuerdo con su especificación (SOMMERVILLE, 2011). A continuación, se detallarán las pruebas funcionales, pruebas de rendimiento y seguridad realizadas a la aplicación.

3.5.1 Pruebas de Rendimiento

Estas pruebas se usan para descubrir problemas de rendimiento que pueden ser resultado de: falta de recursos en el lado servidor, red con ancho de banda inadecuada, capacidades de base de datos inadecuadas, capacidades de sistema operativo deficientes o débiles, funcionalidad del sistema pobremente diseñada y otros conflictos de *hardware* o *software* que pueden conducir a rendimiento cliente-servidor degradado. Las pruebas de rendimiento se diseñan para simular situaciones de carga del mundo real. Conforme aumenta el número de usuarios simultáneos del sistema o el número de transacciones en línea o la cantidad de datos (descargados o subidos) (Pressman, 2010).

3.5.2 Resultados de las pruebas de rendimiento

Para realizar estas pruebas se utiliza la herramienta JMeter en su versión 2.10, la cual es diseñada para realizar Pruebas de Rendimiento sobre aplicaciones *web*. Para ello se definen las propiedades de las *PC* implicadas.

***Hardware* de prueba (*PC* cliente):**

- Sistema Operativo: Windows v.8
- Microprocesador: Intel(R) Core (TM) i3-5020U CPU @2.20GHz 2.20GHz
- Memoria RAM: 4.00 GB
- Disco Duro: 1TB

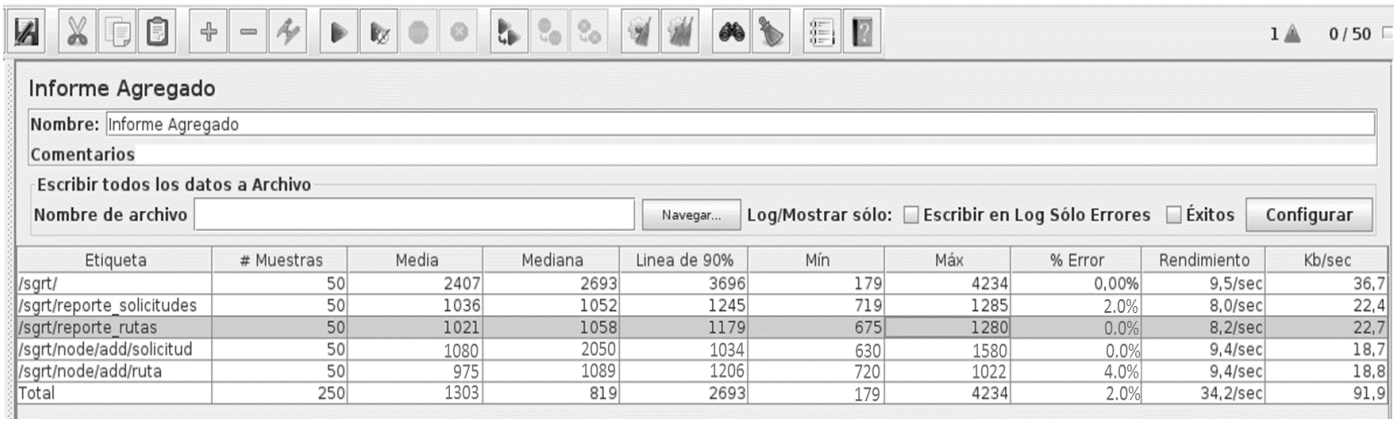
***Hardware* de prueba (*PC* servidor):**

- Sistema Operativo: Linux Mint v.18
- Microprocesador: Intel(R) Core (TM) i3-7100U CPU @2.40GHz 2.40GHz
- Memoria RAM: 8.00 GB
- Disco Duro: 1024 GB

***Software* instalado en ambas *PC*:**

- Tipo de servidor *web*: Apache 2.4.
- Plataforma: SO Linux (*PC* servidor) y SO *Windows* (*PC* cliente).
- Servidor de BD: *MySQL* 5.0.12

Después de definido el *hardware*, se configuran los parámetros del Apache JMeter logrando un ambiente de simulación con un total de 50 usuarios conectados concurrentemente, se realizan peticiones a diferentes páginas del sistema. En la siguiente figura se puede observar los resultados obtenidos.



Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/sgrt/	50	2407	2693	3696	179	4234	0,00%	9,5/sec	36,7
/sgrt/reporte_solicitudes	50	1036	1052	1245	719	1285	2.0%	8,0/sec	22,4
/sgrt/reporte_rutas	50	1021	1058	1179	675	1280	0.0%	8,2/sec	22,7
/sgrt/node/add/solicitud	50	1080	2050	1034	630	1580	0.0%	9,4/sec	18,7
/sgrt/node/add/ruta	50	975	1089	1206	720	1022	4.0%	9,4/sec	18,8
Total	250	1303	819	2693	179	4234	2.0%	34,2/sec	91,9

Figura 17. Resultados obtenidos por el sistema (Elaboración propia).

3.5.3 Análisis de los resultados de las pruebas de rendimiento

A continuación, se explica cada parámetro, para un mejor entendimiento de las pruebas:

- **Usuarios:** total de usuarios.
- **# Muestras:** El número de peticiones.
- **Media:** El tiempo medio transcurrido en milisegundos para un conjunto de resultados.
- **Mín:** El mínimo tiempo transcurrido en milisegundos para las muestras de la URL dada.
- **Máx:** El máximo tiempo transcurrido en un milisegundo para las muestras de la URL dada.
- **% Error:** Porcentaje de las peticiones con errores.
- **Rendimiento:** Rendimiento medido en base a peticiones por segundo/minuto/hora.
- **Kb/s Recibidos:** Rendimiento medido en Kbytes por segundos.

Tabla 7. Resultados de pruebas de rendimiento (Elaboración propia).

Usuarios	# Muestras	Media	Mín	Máx	% Error	Rendimiento (peticiones/segundos)	Kb/s Recibidos
50	250	1303	179	4234	2 %	34.2	91.9

El sistema desarrollado, para un total de 50 usuarios conectados de forma concurrente respondió 250 peticiones al servidor en un promedio de 1.303 segundos, lo que equivale a 34.2 peticiones por segundo, con un porcentaje de error de 2% y un total de 91.9 Kb/s recibidos. Atendiendo a la cantidad de peticiones por cada segundo que se enviaron y a las prestaciones del *hardware* donde se realizaron las pruebas, se considera que constituye un resultado satisfactorio que debe ser mejorado en futuras iteraciones de pruebas para el sistema.

3.5.4 Pruebas funcionales

Las pruebas funcionales tienen como objetivo asegurar que el *software* desarrollado cumpla con las especificaciones requeridas y eliminar los posibles defectos que este pudiera tener. Para la realización de estas pruebas se utilizó el método de caja negra las cuales son realizadas desde el punto de vista de interfaz de usuario. Con el objetivo de realizar este tipo de pruebas al sistema, se diseñó un conjunto de casos de pruebas, mostrados a continuación.

Tabla 8. Variables empleadas en el diseño del caso de prueba: Crear solicitud (Elaboración propia).

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre de jefe de proyecto	Campo de texto	no	Se deben escribir palabras comenzando por mayúscula y sin caracteres extraños, que no excedan un total de 20 caracteres.
2	Proyecto	Campo de texto	no	Se deben escribir palabras comenzando por mayúscula y sin caracteres extraños, que no excedan un total de 30 caracteres.
3	Centro	Campo de selección	no	Se debe seleccionar uno de los centros mostrados
4	Fecha	Calendario desplegable	no	Se debe seleccionar una fecha de las preestablecidas.
5	Hora Ida	Campo de selección	no	Se debe seleccionar una sola opción de las mostradas.
6	Hora Regreso	Campo de selección	no	Se debe seleccionar una sola opción de las mostradas.
7	Cantidad de Personas	Campo de selección	no	Se debe seleccionar una sola opción de las mostradas.
8	Dirección	Campo de texto	no	Se debe insertar una cadena de texto, que coincida con la estructura válida de un dirección.
9	Estado	Campo de selección	no	Se debe seleccionar una sola opción de las mostradas.

Tabla 9. Caso de prueba: Crear solicitud (Elaboración propia).

Escenario	Descripción	V1	V2	V3	V4	V5	V6	V7	V8	V9	Respuesta del sistema	Flujo central
EC 1.1 Insertar datos correctamente	El usuario introduce de manera correcta los datos en el sistema	V	V	V	V	V	V	V	V	N/A	El sistema registra la solicitud.	El usuario rellena los campos de forma correcta y da clic en el botón Guardar.
		Mario	SGRT	FORTES	12/06/2019	8:00 am	12:00pm	7	Fiscalía General de la República	vacío		
EC 1.2 Campos vacíos	El usuario deja uno o más campos vacíos	V	I	I	V	V	V	V	V	N/A	El sistema muestra un mensaje: “El campo es obligatorio”.	El usuario llenando los campos deja uno o más vacíos y da clic en el botón Guardar.
		Mario	vacío	vacío	12/06/2019	8:00 am	12:00pm	7	Fiscalía General de la Re	vacío		

									pública			
		I	V	V	I	V	V	V	V	N/A		
		vacío	SGRT	FORTES	vacío	8:00 am	12:00pm	7	Fiscalía General de la República	vacío		
EC 1.3 Insertar datos incorrectos	El usuario introduce uno o más valores incorrectos	V	V	V	I	V	V	V	V	N/A	El sistema muestra un mensaje: “El campo es inválido”.	El usuario introduce uno o más datos incorrectos y da clic en el botón Guardar
		Mario	SGRT	FORTES	aa	8:00 am	12:00pm	7	Fiscalía General de la Repú	vacío		

									blica			
		I	V	V	V	V	V	V	V	N/A		
		0	SGRT	FORTES	12/06/2019	8:00 am	12:00pm	7	Fiscalía General de la República	vacío		

3.5.5 Resultado de las pruebas funcionales

Luego de realizados los casos de prueba descritos anteriormente, se realiza una primera iteración donde se detectaron 15 no conformidades, que provocaban que el sistema no se comportara de la forma esperada, de las cuales 6 eran errores ortográficos, 6 de validación y 3 opciones que no funcionaban. De estas no conformidades se solucionaron las 15. Se realizó una regresión para comprobar si se habían solucionado correctamente y que no hubieran surgido nuevas no conformidades, pasando así satisfactoriamente a una segunda iteración. Después de realizada la segunda iteración se encontraron 5 no conformidades, caracterizadas en errores ortográficos, las cuales fueron solucionadas en su totalidad para luego hacer una regresión donde no se encontraron nuevas no conformidades. Se ejecutó una tercera iteración obteniéndose un resultado satisfactorio para cada una de las combinaciones de datos por escenario. La siguiente figura muestra estos resultados en forma de gráfica:

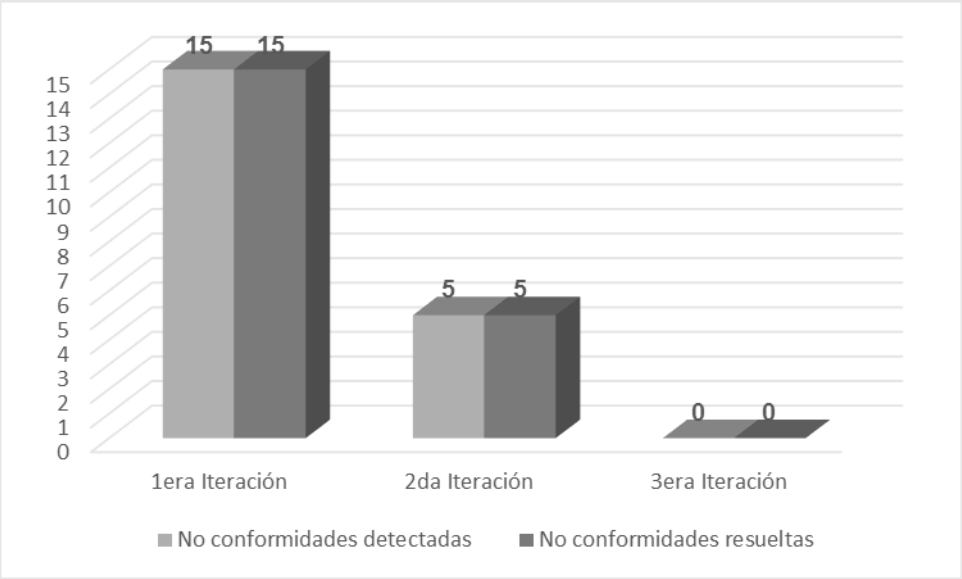


Figura 18. Comportamiento de las no conformidades por iteración (Elaboración propia).

3.5.6 Pruebas de seguridad

Las pruebas de seguridad intentan verificar que los mecanismos de protección que se construyen en un sistema en realidad lo protegerán de cualquier penetración impropia. Se diseñan para sondear las vulnerabilidades que ocurren en la comunicación establecida entre cliente y el servidor (Pressman, 2010).

3.5.7 Resultados de las pruebas de seguridad

Con el objetivo de evaluar la seguridad de la solución propuesta se emplea la herramienta Acunetix WVS v8 la cual arrojó los siguientes resultados.

Tabla 10. Resultados de la prueba de seguridad (Elaboración propia).

Categorías de vulnerabilidades	Nivel
Cross Site Scripting (Secuencias de comandos entre sitios)	Alto
User credentials are sent in clear text (Las credenciales de usuario se envían en texto claro)	Medio

Clickjacking: X-Frame-Options header missing (Falta el encabezado de opciones de *X-Frame-Options*)	Baja
Broken Links (Enlaces rotos)	Información

La prueba realizada mediante la herramienta Acunetix WVS v8, arrojó que en la primera iteración se encontraron 4 alertas en total, de ellas una de clasificación alta, una de clasificación media, una de clasificación baja y una informacional. Todas las vulnerabilidades fueron resueltas mediante la instalación de los módulos *Security Kit* y *Login Security* en el sistema. Se realizó una segunda iteración donde no se encontraron vulnerabilidades, por lo que se puede afirmar, que el sistema desarrollado es seguro y está en condiciones de ser usado por el cliente.

3.6 Evaluación del objetivo de investigación

Para la validación del resultado obtenido se utilizó la técnica de Iadov. Esta técnica es un instrumento que ayuda a conocer el grado de satisfacción de los potenciales usuarios. Constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios utilizados se fundamentan en las relaciones que se establecen entre tres preguntas cerradas, las cuales se intercalan dentro de un cuestionario y cuya relación el sujeto desconoce. Esta técnica incluye también dos preguntas abiertas que indagan sobre lo que más les gusta o disgusta a los encuestados de la actividad objeto de estudio. Las tres preguntas cerradas se relacionan a través de lo que se denomina "Cuadro Lógico de lADOV", el cual permite ubicar a cada encuestado, según el cuadro lógico en una escala de satisfacción, para luego calcular el ISG (Índice de Satisfacción Grupal).

Las respuestas a cada una de estas preguntas permiten determinar la posición de cada sujeto en la escala de satisfacción que toma valores desde 1 hasta 6, distribuido de la siguiente manera: 1-Clara satisfacción, 2-Más satisfecho que insatisfecho, 3-No definida, 4-Más insatisfecho que satisfecho, 5-Clara insatisfacción y 6-Contradictoria (Diaz, et al., 2018). La tabla a continuación muestra el cuadro lógico utilizado en la investigación.

Tabla 11. Cuadro lógico de Iadov utilizado (Elaboración propia).

¿Qué tan satisfecho se siente con los resultados del sistema de gestión para la reserva de transporte de los centros de producción en la UCI?	¿Estima beneficioso el sistema desarrollado para los centros de producción y la dirección de transporte en la UCI?								
	Sí			No sé			No		
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me satisface	1	2	6	2	2	6	6	6	6
Me resulta más satisfactorio que insatisfactorio	2	2	3	2	3	3	6	3	6
Me son indiferentes	3	3	3	3	3	3	3	3	3
Me resulta más insatisfactorio que satisfactorio	6	3	6	3	4	4	3	3	4
No me satisfacen en lo absoluto	6	6	6	6	4	4	6	6	5
No sé decir	2	3	6	3	3	3	6	6	4

Para medir el grado de satisfacción de los usuarios respecto al sistema desarrollado, se tomó como muestra de siete especialistas y profesores del centro CIDI. La selección se realizó teniendo en cuenta que muchos de estos trabajadores serán clientes del sistema, lo que puede ser útil para conocer las debilidades y fortalezas del mismo. Los resultados obtenidos para la satisfacción de forma individual se exponen a continuación.

A partir de la cantidad de respuestas por categoría es posible calcular el Índice de Satisfacción Grupal (ISG) siguiendo la siguiente fórmula:

$$ISG = \frac{A\ (+\ 1) + B\ (+\ 0{,}5) + C\ (0) + D\ (-\ 0{,}5) + E\ (-\ 1)}{N}$$

Las variables representan las cantidades de participantes agrupados por las escalas del índice de satisfacción individual. La cantidad de participantes que expresaron tener una clara satisfacción son representados por A, la cantidad que se sienten más satisfechos que insatisfechos se expresan mediante B, no definido y contradicción se evidencia mediante C, los que se sienten más insatisfechos que satisfechos mediante D, E es la cantidad de participantes que expresan una clara insatisfacción. El valor de N representa el total de participantes.

El valor del ISG permite identificar las siguientes categorías grupales:

- Máxima insatisfacción: -1
- Más insatisfecho que satisfecho: -0.5
- No definido y contradictorio: 0
- Más satisfecho que insatisfecho: 0.5
- Máximo de satisfacción: +1

Esta técnica permite determinar el ISG, que representa los niveles de satisfacción en una escala numérica que abarca el intervalo desde -1 hasta 1. Los valores que se encuentran comprendidos entre -1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que están entre 0,5 y 1 indican que existe satisfacción.

Resultados obtenidos

Los resultados obtenidos de la aplicación de la encuesta se presentan a continuación:

Tabla 12 Escala del índice de satisfacción individual (Elaboración propia).

Categorías grupales de satisfacción	Escala	Participantes en la escala
Clara satisfacción	A	5
Más satisfecho que insatisfecho	B	2
No definido	C	0
Más insatisfecho que satisfecho	D	0

Clara insatisfacción	E	0
Contradictorio	C	0

Cálculo del Índice de Satisfacción Grupal

ISG= (5(+1) +2(+0.5)) / 7 = 0.86

Interpretación del resultado del ISG

En esta investigación el Índice de Satisfacción Grupal obtenido fue de ISG = 0,86 utilizando el proceso de validación mediante la técnica de Iadov de la consulta a los usuarios. Como se puede apreciar el valor del Índice es alto, lo que refleja aceptación de la propuesta y un reconocimiento a su utilidad donde se ha implementado el sistema de gestión para la reserva de transporte de los centros de producción en la UCI. Confirmando su factibilidad de uso, con los criterios emitidos donde evidencian su satisfacción por la contribución del sistema. Los beneficios que traerá la utilización del sistema propuesto se reafirman en las respuestas a las preguntas abiertas brindadas por los encuestados. Se puede afirmar que se cumplió el objetivo de la investigación por todo lo anteriormente planteado.

A continuación, se muestran las dos preguntas abiertas realizadas a los participantes.

1. ¿Qué importancia le concede al sistema de gestión para la reserva de transporte de los centros de producción en la UCI?
2. ¿Qué aspectos a su juicio fortalecen o limitan el uso del sistema de gestión para la reserva de transporte de los centros de producción en la UCI?

Sobre la primera pregunta, los participantes manifestaron que resulta de gran importancia porque permite brindar información en tiempo real de las solicitudes realizadas. Permite generar históricos de las solicitudes y rutas realizadas, lo que proporciona información relevante a la hora de planificar la transportación. Todo esto posibilita organizar y agilizar el trabajo de los involucrados en el proceso de reserva de transporte.

Sobre la segunda pregunta los participantes opinaron que la creciente necesidad de agilizar y confirmar la entrega de las solicitudes de reserva, potencia el uso de esta herramienta. En cuanto a las limitaciones, consideran que todas se encuentran en el orden técnico, puede verse limitado su uso por las posibles fallas o interrupciones en el funcionamiento del sistema en los servidores o su estabilidad por cortes en el respaldo energético.

3.7 Conclusiones Parciales

Con la realización de este capítulo quedó definido el modelo de implementación del sistema de gestión de reservas de transporte de los centros de producción en la UCI. A través del diagrama de componentes, se describieron cada uno de los componentes y su influencia en la solución obtenida. Mediante el diagrama de despliegue se establece el *hardware* y su relación con el *software* necesario para el correcto funcionamiento del sistema. Se le realizaron pruebas de *software* al sistema para validar su calidad mediante pruebas de rendimiento, seguridad y funcionalidad, esta última mediante el apoyo de los diseños de casos de prueba.

CONCLUSIONES GENERALES

- ✓ El estudio de los referentes teóricos y el análisis de las diferentes herramientas y tendencias para la gestión de reservas de transporte permitió determinar la no existencia de un sistema informático que responda a las necesidades actuales de los centros de producción en la UCI, no obstante, sirvieron de base para el desarrollo del sistema de gestión de reserva de transporte de los centros de producción en la UCI.
- ✓ El diseño de la propuesta de solución permitió generar los artefactos más significativos de acuerdo con la metodología de desarrollo de *software* AUP-UCI tomándose como referencia los requisitos detectados.
- ✓ Como resultado de la implementación se obtuvo un sistema funcional que cumple con los requerimientos especificados.
- ✓ La solución fue debidamente probada y las deficiencias detectadas en cada una de las iteraciones de prueba, fueron corregidas en el tiempo establecido, lo que posibilitó la efectividad de la solución propuesta. Estas pruebas demostraron que es una solución funcional, segura y con un rendimiento adecuado.
- ✓ La validación del objetivo mediante la técnica Iadov demostró que el sistema contribuye a mejorar el proceso de reserva de transporte de los centros de producción en la UCI.

REFERENCIAS BIBLIOGRÁFICAS

1. **Cubana. 2017.** Aviacion, C.D. Cubana. [En línea] 2017. [Citado el: 11 de 8 de 2018.] http://www.cubana.cu/home/.
2. **El Diccionario de Transporte . 2016.** Un sistema de gestión de transporte | El Diccionario de Transporte . [En línea] 2016. [Citado el: 4 de 12 de 2018.] http://www.timocom.es/?lexicon=1503241125198229%7CUn%20sistema%20de%20gesti%C3%B3n%20de%20transporte%7CEl%20Diccionario%20de%20Transporte.
3. **acunetix. 2018.** acunetix. [En línea] 2018. https://www.acunetix.com/.
4. **Bootstrap. 2017.** Bootstrap. [En línea] 2017. [Citado el: 4 de 8 de 2018.] http://getbootstrap.com/.
5. **Definición de sistema. 2016.** Definición de sistema - Qué es, Significado y Concepto. [En línea] 2016. [Citado el: 21 de 11 de 2018.] http://definicion.de/sistema/.
6. **Definición de transporte. 2016.** Definición de transporte - Qué es, Significado y Concepto. [En línea] 2016. [Citado el: 5 de 11 de 2018.] http://definicion.de/transporte/..
7. **Diaz, MSc. Aymara Marin y Casañola, DrC. Yaimí Trujillo. 2018.** *Marco de Trabajo para gestionar actividades de calidad.* 2018. ISSN: 2227-1899 | RNPS: 2301.
8. **Expedia. 2014.** Expedia. [En línea] 2014. [Citado el: 24 de 8 de 2018.] https://Expedia.es./.
9. **Gama, Erich, y otros. 2014.** *Patrones de diseño. Elementos de software orientado a objetos reutilizable.* Madrid : Pearson Education, 2014.
10. **Gil, Fran. 2012.** *Experto en Drupal 7 Nivel Avanzado Curso de creación y gestión de portales web con Drupal 7.* 2012. 978-84-939410-5-5.
11. **—. 2012.** *Experto en Drupal 7 Nivel Inicial Curso de creación y gestión de portales web con Drupal 7.* 2012. 978-84-939410-3-1.
12. **Guerra, Sánchez y Esther. 2016.** *Patrones de diseño: Patrón de comportamiento Decorator.* 2016.
13. **html-MDN. 2018.** html-MDN. [En línea] 2018. [Citado el: 3 de 1 de 2019.] http://www.html-MDN.com/.
14. **Jardinot González, Raúl Rolando . 2015.** *Módulo de registro de la actividad de los usuarios en aplicaciones web desarrolladas con Drupal 7.* s.l. : UCI, 2015.

15. **jmeter. 2018.** jmeter. [En línea] 2018. https://jmeter.apache.org.

16. **Joomla! 2017.** Joomla! The Free Software Powering Millions of Websites. [En línea] 2017. [Citado el: 7 de 12 de 2018.] https://www.joomla.org/about-joomla.html.

17. **Logitravel. 2014.** Logitravel. [En línea] 2014. [Citado el: 5 de 10 de 2018.] https://Logitravel.com/.

18. **Marimón, Mezquía y Diosbel. 2013.** *Perfil de instalación de portal intranet para el Sistema Gestor de Contenidos Drupal 7.* La Habana : s.n., 2013.

19. **Marqués, Andrés y María. 2015.** *Modelos de datos.* 2015.

20. **MySQL. 2018.** MySQL. Sitio Web de MySQL. [En línea] 2018. [Citado el: 7 de 2 de 2019.] http://dev.mysql.com.

21. **Naranjo, D. L. N. 2014.** *Aplicaciones Web Integrales. Breves Apuntes.* Holguín : s.n., 2014. ISBN 1818-3018..

22. **NetBeans. 2016.** NetBeans. Sitio oficial del IDE Netbeans. [En línea] 2016. [Citado el: 22 de 10 de 2018.] http://netbeans.org].

23. **PHP. 2018.** The PHP Group. [En línea] 2018. [Citado el: 22 de 11 de 2018.] http://us.php.net/manual/en/preface.php.

24. **Pina, Legón, Leonar. 2016.** *Perfil de instalación de Drupal 7 para el Departamento de Servicios Informáticos para Internet. UCI.* La Habana : s.n., 2016.

25. **PMBOK. 2013.** *Guía de los fundamentos para la dirección de proyectos (guía del pmbok).* s.l. : 5 edición, 2013. ISBN 978-1-62825-009-1.

26. **Pressman, Roger S. 2010.** *Ingeniería del software. Un enfoque práctico (7ª. ed.).* México : McGraw-Hill Interamericana, 2010.

27. **Rabilero, Ramón Fernández y Varona Carmenates, Arian. 2016.** *Sistema para la Gestión de las Transportaciones Nacionales de la Universidad de las Ciencias Informáticas.* LA HABANA : s.n., 2016.

28. **Ramírez, Marlon. 2018.** Aplicación de la técnica de Iadov. 2018.

29. **Roberth G. Figueroa, Armando A. Cabrera. 2016.** *Metodologías Tradicionales VS. Metodologías Ágiles .* s.l. : Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación, 2016.

30. **Rodríguez González, Isied y Miras González, Luis Andrés. 2015.** *Gestión de reservaciones de las transportaciones nacionales de la Universidad de las Ciencias informáticas.* La Habana : s.n., 2015.

31. **Sánchez, Ing. María A. Mendoza. 2015.** *Metodologías de desarrollo de software.* Perú : s.n., 2015.

32. **Sánchez, Jorge. 2016.** *Java Script manual de referencia.* Ciudad de Mexico, D.F : ECMA, 2016.

33. **Sánchez, Tamara Rodríguez. 2015.** *Metodología de desarrollo para la actividad productiva de la UCI.* La Habana : s.n., 2015.

34. **Significado de Gestión. 2015.** Significado de Gestión - Qué es, Concepto y Definición. [En línea] 2015. [Citado el: 7 de 10 de 2018.] http://www.significados.com/gestion/..

35. **Significado de Sistema. 2015.** Significado de Sistema - Qué es un, Concepto y Definición. [En línea] 2015. [Citado el: 23 de 11 de 2018.] http://www.significados.com/sistema/.

36. **Silega, Noris. 2017.** ¿Qué es la técnica de Iadov? [En línea] 16 de 02 de 2017. [Citado el: 16 de 03 de 2019.]

37. **SOMMERVILLE, Ian. 2011.** *Ingeniería del software.* México : Pearson Educación, 2011. ISBN: 978-607-32-0603-7.

38. **Stallings, William. 2014.** *Comunicaciones y redes de computadoras. VI Edición.* Barcelona : Prentice Hall, 2014.

39. **Travelport. 2014.** Estadísticas de los GDS de Travelport, Sitio Web oficial de Travelport. [En línea] 2014. [Citado el: 20 de 11 de 2018.] https://es.travelportservices.com/extranet.

40. **UCI. 2016.** UCI. [En línea] 2016. [Citado el: 2 de 12 de 2018.] http://uci.cu.

41. **Viazul. 2016.** Sitio Web Viazul. [En línea] 2016. [Citado el: 5 de 9 de 2018.] http://viazul.com.

42. **Visual Paradigm. 2017.** Visual Paradigm. [En línea] 2017. [Citado el: 26 de 11 de 2018.] http://www.visualparadigm.com/aboutus/newsreleases/vpuml80.jsp..

43. **W3C. 2018.** W3C. [En línea] 2018. [Citado el: 5 de 12 de 2018.] http://www.w3.org/html/.

44. **Wordpress. 2018.** Wordpress.org. WordPress.org–Español. [En línea] 2018. [Citado el: 4 de 12 de 2018.]

ANEXOS

Anexo 1: Entrevista realizada a Sahilyn Delgado Pimentel directora del centro (CIDI) y a Braydi Ramos Enríquez jefe de piquera de la dirección de transporte en la UCI.

1. ¿Quiénes son los encargados de aceptar la solicitud de transporte?
2. ¿Qué datos son necesarios para realizar la solicitud?
3. ¿Hasta qué día de la semana los líderes de proyecto pueden realizar la solicitud de transporte?
4. ¿Se notifica a los líderes de proyecto la aceptación o no de las solicitudes de transporte?
5. ¿Cuántas solicitudes se realizan diariamente?
6. ¿Qué datos contiene el Excel donde se agrupan las solicitudes del centro CIDI?

GLOSARIO DE TÉRMINOS

CMS: Sistema Gestor de Contenidos. CMS es un *software* para el manejo de contenidos de sitios *web* directamente desde el servidor. Sistema Gestor de Contenidos. CMS es un *software* para el manejo de contenidos de sitios *web* directamente desde el servidor.

CSS: Hojas de Estilo en Cascada, del inglés *Cascading Style Sheets.*

GOF: Patrones de diseño de la banda de los cuatro, del inglés *Gang Of Four*.

GPL: Licencia publica general*, del inglés General Public License.*

GRASP: Patrones de Asignación de Responsabilidades.

***Hooks* (ganchos):** Se utilizan en Drupal como unas funciones que permiten la comunicación entre los distintos módulos del sistema

HTML: Lenguaje de Marcado de Hipertexto, del inglés *HyperText Markup Language.*

HU: Historia de Usuario.

PDF: Documento en formato portable, *del inglés Portable Document Format.*

PHP: Pre-procesador de Hipertexto, del inglés *HyperText Pre-Processor.*

UML: Lenguaje Unificado de Modelado, del inglés *Unified Modeling Language.*

***Web*:** es un sistema de distribución de documentos de hipertexto o hipermedios interconectados y accesibles vía Internet. Con un navegador *web*, un usuario visualiza sitios *web* compuestos de páginas *web* que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de esas páginas usando hiperenlaces.