



Trabajo de Diploma para optar por el título de
Ingeniero Informático

Sistema de planificación de visitas para el Centro de Soporte

Autor: Yoel Antonio Casado González

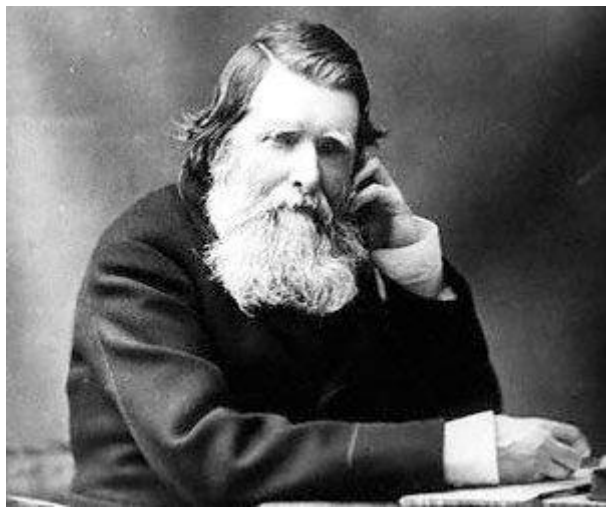
Tutor: Ing. Nelson Sánchez Álvarez

La Habana, 2017

Frase:

*La calidad nunca es un accidente;
siempre es el resultado de un esfuerzo de la
inteligencia.*

John Ruskin.



Sistema de planificación de visitas para el Centro Soporte

Declaración de Autoría

Declaro ser autor de la presente tesis que tiene por título: Sistema de planificación de visitas para Centro de Soporte y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los 29 días del mes de junio del año 2017.

Yoel Antonio Casado González

Firma del Autor

Nelson Sánchez Álvarez

Firma del Tutor

Sistema de planificación de visitas para el Centro Soporte

Datos de Contacto

Nombre: Nelson Sánchez Álvarez.

Correo: nalvarez@uci.cu.

Categoría Científica: Ingeniero en Ciencias Informáticas.

Sistema de planificación de visitas para el Centro Soporte

Agradecimientos

Primeramente a mi país por permitirme tener la posibilidad de graduarme como ingeniero. A todas aquellas personas que creyeron en mí y me brindaron su apoyo incondicional en todo momento. A mi familia, que siempre creyó en mí y supieron estar ahí para decirme tu puedes. A todos los profesores y compañeros que de distintas formas contribuyeron al desarrollo de este trabajo y sin ellos no hubiese obtenido este resultado. A mi tutor que siempre estuvo a mi lado y me trató no solamente como su tesista sino como un hermano. Especialmente a mis compañeros de los cinco años de carrera que hasta hoy estamos juntos y su apoyo no he dejado de existir en ninguna circunstancia. Y por último y no menos importante a la UCI por haber sido mi hogar durante estos maravillosos años de carrera.

Sistema de planificación de visitas para el Centro Soporte

Dedicatoria

A mi padre, pues su valor y sacrificio me ha demostrado que todo es posible, quien me ha demostrado la importancia de superarse y ser mejor persona cada día. El que nunca ha dudado de mí, y siempre estando a mi lado en los momentos buenos y malos. Te dedico este trabajo, y te dedicaré mis días, como lo has hecho tú por mí. A mi abuela del alma, pues siempre ha estado presente de una u otra forma. A mis tíos, en especial a Sonia, que ha sido como una madre para mí. Al compañero Fidel, por hacer posible que todos tengamos esta posibilidad de superación. En general dedico esta tesis a toda mi familia por ser gracias a ellos que este yo aquí en este día.

Sistema de planificación de visitas para el Centro de Soporte

Resumen

La correcta gestión de los recursos en una entidad posee un alto valor estratégico, pues de ella dependen los gastos para la ejecución de la producción. En la Universidad de la Ciencias Informáticas, el Centro de Soporte brinda servicios a las entidades que contratan el soporte técnico a los productos de software, por lo que se hace necesario mantener una estrecha relación con el cliente. Por medio de estos contratos y dependiendo de su tipo, el cliente puede convenir que sus problemas sean resueltos in-situ, generando una gran cantidad de visitas para los especialistas del centro. Sin embargo, no se cuenta con un mecanismo que gestione las visitas generadas por los diferentes problemas presentados en la explotación de esos productos. Con la implementación de un sistema informático que gestione ese proceso, se obtuvo un mayor control de las vistas generadas, una mayor capacidad de operación y se evitó la sobrecarga a especialistas. Además, se obtienen informes de satisfacción y resultados de dichas visitas, facilitando una estadística sobre la cual será posible predecir posibles problemas y actuar de forma proactiva. Para el desarrollo de esta herramienta se efectuó un análisis de sistemas homólogos, que a la postre resultaron no ser efectivos para la problemática presentada. Por ende, se decidió elaborar una solución que estuviera acorde a las políticas de informatización del país, sobre tecnologías libres y que fuera reutilizable en otros ámbitos con situaciones similares.

Sistema de planificación de visitas para el Centro de Soporte

Abstract

The correct management of the resources in an entity has a high strategic value, since it depends on the expenses for the production execution. AT the University of Computer Science, the Support Center provides services to entities that contract technical support for software products, so it is necessary to maintain a close relationship with the client. The client can agree that their problems are solved on-site, through these contracts and depending on their type, which generates a lot of visits for the specialists of the center. However, there is not a mechanism that manages the visits generated due to different problems presented in the exploitation of these products. The implementation of a computer system that managed this process, was obtained a greater control of the generated views, a greater capacity of operation and was avoided the overloading to specialists. In addition, satisfaction reports and results of such visits are obtained. This provides a statistic on which it will be possible to predict possible problems and act proactively. For the development of this tool was carried out an analysis of homologous systems, which, finally proved they are not effective for the problems presented. Therefore, it was decided to develop a solution that was in line with the country computerization policies, on free technologies and reusable in other areas with similar situations.

Sistema de planificación de visitas para el Centro de Soporte

Índice

<i>Introducción</i>	1
<i>CAPÍTULO I: Fundamentación Teórica</i>	6
1.1 Introducción	6
1.2 Conceptos fundamentales.....	6
1.2.1 Cliente.....	6
1.2.2 Servicio al cliente	7
1.2.3 Gestión de Servicio TI.....	7
1.2.4 Planificación.....	8
1.2.5 Proceso.....	8
1.3 Estudio del estado del Arte de los Sistemas de Planificación.....	9
1.3.1 Diagrama de Gantt.....	9
1.3.2 Sistemas ERP	10
1.3.3 Sistemas de Planificación de la Producción	11
1.3.4 Sistema de Planificación de Actividades	12
1.4 Metodología de Desarrollo	12
1.4.1 Proceso Unificado Ágil (AUP).....	13
1.5 Herramientas y tecnologías.....	16
1.5.1 Lenguajes de programación.....	16
1.5.2 Entorno de Desarrollo Integrado	18
1.5.3 Marco de Trabajo (Framework).....	19
1.5.4 Herramienta para el modelado.....	20
1.5.5 Sistema Gestor de Base de Datos	22
1.6 Conclusiones parciales	22
<i>CAPÍTULO II: Análisis y Diseño del sistema</i>	24
2.1 Introducción	24
2.2 Modelo de Dominio	24
2.3 Solución Propuesta.....	25
2.4 Concepción del Sistema.....	25
2.4.1 Visión y alcance del sistema	26
2.5 Especificación de requisitos	26
2.5.1 Lista de reserva del producto	26
2.5.2 Historias de usuario	28
2.6 Arquitectura del sistema y Diagrama de clases.....	35
2.7 Patrones del Diseño.....	36
2.8 Diagrama Entidad - Relación	40

Sistema de planificación de visitas para el Centro de Soporte

2.9 Conclusiones parciales	41
<i>CAPÍTULO III - Implementación y Pruebas</i>	<i>42</i>
3.1 Introducción.....	42
3.2 Implementación	42
3.2.1 Diagrama de Componentes.....	42
3.2.2 Diagrama de despliegue.....	43
3.2.3 Prototipo de Interfaz de Usuario	44
3.2.4 Estándares de Codificación	46
3.3 Validación de la investigación.....	46
3.4 Pruebas aplicadas al Sistema	47
3.5 Conclusiones parciales.....	54
<i>Conclusiones</i>	<i>55</i>
<i>Referencias bibliográficas</i>	<i>56</i>
<i>Anexos</i>	<i>60</i>

Sistema de planificación de visitas para el Centro de Soporte

Índice de Figuras

<i>Ilustración 1. Tipos de Servicios que ofrece el Centro de Soporte UCI.</i>	2
<i>Ilustración 2. Tipos de Clientes empresariales.</i>	6
<i>Ilustración 3. Fases e iteraciones de Proceso Unificado Ágil.</i>	14
<i>Ilustración 4. Fases e iteraciones de la variación AUP-UCI.</i>	16
<i>Ilustración 5. Diagrama de Dominio del Sistema.</i>	25
<i>Ilustración 6. Diagrama de Clases y arquitectura.</i>	36
<i>Ilustración 7. Patrón de Diseño Controlador.</i>	37
<i>Ilustración 8. Patrón de Diseño Experto.</i>	37
<i>Ilustración 9. Patrón de Diseño Creador.</i>	38
<i>Ilustración 10. Patrón de Diseño Bajo Acoplamiento.</i>	38
<i>Ilustración 11. Patrón de Diseño Alta Cohesión.</i>	39
<i>Ilustración 12. Diagrama Entidad- Relación.</i>	40
<i>Ilustración 13. Diagrama de Componentes.</i>	43
<i>Ilustración 14. Diagrama de Despliegue.</i>	44
<i>Ilustración 15. Prototipo Gestión de visitas.</i>	45
<i>Ilustración 16. Prototipo Interfaz Planificación.</i>	45
<i>Ilustración 17. Interfaz de Usuario HU Gestionar visita.</i>	51
<i>Ilustración 18. Pruebas unitarias AppControllerTest.</i>	54
<i>Ilustración 19. Pruebas unitarias TipoVisitaControllerTest.</i>	54

Sistema de planificación de visitas para el Centro de Soporte

Índice de Tablas

<i>Tabla 1. Disciplinas de la metodología AUP, según la variación UCI.</i>	15
<i>Tabla 2. Requisitos funcionales y no funcionales.</i>	28
<i>Tabla 3. Historia de Usuario “Autenticar usuario”</i>	29
<i>Tabla 4. Historia de Usuario “Asignar roles”</i>	30
<i>Tabla 5. Historia de Usuario “Gestionar visitas”</i>	31
<i>Tabla 6. Historia de Usuario “Realizar búsquedas de visitas”</i>	32
<i>Tabla 7. Historia de Usuario “Mostrar datos de las visitas buscadas”</i>	33
<i>Tabla 8. Historia de Usuario “Realizar informes de visitas”</i>	34
<i>Tabla 9. Historia de Usuario “Graficar los resultados”</i>	35
<i>Tabla 10. PA a la HU_3-1 Adicionar Visita</i>	48
<i>Tabla 11. PA a la HU_15-1 Realizar informe de visitas</i>	49
<i>Tabla 12. PA a la HU_14-1 Alertar sobre visita ejecutada sin resumen</i>	49
<i>Tabla 13. PA a la HU_5-1 Notificar visita</i>	49
<i>Tabla 14. PA a la HU_9-1 Modificar una visita</i>	50
<i>Tabla 15. Caso de prueba # 2 utilizando método de caja negra</i>	52
<i>Tabla 16. Caso de prueba # 3 utilizando método de caja negra</i>	52
<i>Tabla 17. Caso de prueba # 4 utilizando método de caja negra</i>	53

Sistema de planificación de visitas para el Centro de Soporte

Introducción

Las Tecnologías de la Información y las Comunicaciones (TICs), desde su surgimiento, han puesto a disposición de la comunidad un conjunto de herramientas para facilitar la ejecución de las actividades de empresas, organizaciones y de cada persona en particular. Definidas por diferentes autores, las TICs se conocen como aquellas “tecnologías que permiten la adquisición, almacenamiento, procesamiento, evaluación, transmisión, distribución y difusión de la información. Dichas TIC son desarrolladas mediante la convergencia de la informática, las telecomunicaciones, la electrónica y la microelectrónica” (UIS Ingenierías, 2012). En el ámbito del procesamiento y tratamiento de la información se encuentran las TI (Tecnologías de la Información) como un subconjunto altamente utilizado en diferentes áreas del conocimiento.

Un ejemplo específico de áreas donde las TI son un factor clave es los servicios y la forma en que estos son gestionados. El concepto de gestión se comprende como *“la capacidad de alcanzar lo propuesto, ejecutando acciones y haciendo uso de los recursos técnicos, financieros y humanos disponibles (Botero Chica, 2009)”*. El término, asociado a los servicios TI, se conoce como GESTI¹ y representa *“una disciplina centrada en el cliente. Su objetivo es contribuir a la calidad de los servicios de TI, e incluye la gestión de calidad y control de procesos para los servicios de TI como una parte de la organización y sus políticas (Lucio Nieto, 2012)”*

La GESTI agrupa un conjunto de estándares y buenas prácticas que son utilizadas para guiar las organizaciones en la implementación y ejecución de los servicios que brindan, entre los que se encuentran COBIT², ITIL³, CMMI-SVC⁴ e ISO/IEC 20 000⁵. La utilización de estos u otras buenas prácticas facilita la calidad de los servicios que se brindan, sin embargo, los procesos y actividades relacionadas con los clientes son complejas y deben ser manejadas con un alto nivel de relevancia.

La calidad en el servicio al cliente no es un tema reciente dentro de las empresas, pues son la clave del éxito en la ejecución de cualquier servicio, incluidos los TI. Es por ello que es necesario ejecutar constantemente acciones que permitan ganar su confianza y mantener su interés en el intercambio y la retroalimentación. Estas prácticas se han acrecentado y diversificado llegando a cada lugar del mundo donde exista una empresa u organización, que comprenda entre sus objetivos el de brindar este tipo de servicios, teniendo una amplia representación incluso en América Latina. (BID (Banco Interamericano de Desarrollo), 2017)

Cuba como parte de su crecimiento y desarrollo tecnológico cuenta con entidades que representan la informática y las comunicaciones con excelentes resultados, pudiéndose mencionar la

¹ Gestión de Servicios TI.

² Del inglés: Control Objectives for Information and related Technology (Objetivos de Control para la Información y Tecnologías Relacionadas).

³ Del inglés: Information Technology Infrastructure Library.

⁴ Del inglés: Capability Maturity Model Integration for Services (Modelo de Madurez de la Capacidad Integrado).

⁵ Del inglés: International Organization for Standardization.

Sistema de planificación de visitas para el Centro de Soporte

Universidad de las Ciencias Informáticas (UCI). Entre sus principales objetivos la UCI destaca la producción de software a partir de su modelo de formación que incluye la vinculación estudio-trabajo. Para esto se apoya en diferentes centros creados por la Universidad, entre los que se encuentra el Centro de Soporte. Este centro se encarga de “brindar el servicio de soporte técnico a las aplicaciones y servicios informáticos desarrollados por la UCI, con calidad y eficiencia, a partir de su correcta gestión y garantizando elevados niveles de satisfacción de sus clientes” (Soporte UCI, 2016).

Con el objetivo de alcanzar la misión planteada, el Centro de Soporte cuenta con un conjunto de procesos, procedimiento y actividades continuas con el fin de mantener la calidad del servicio. Entre ellos se encuentra el proceso de contratación, a través del cual se organiza la relación entidad – cliente con los elementos legales necesarios. Este proceso, además, orienta sobre los servicios específicos que brindan el centro y las facilidades que posee cada uno de ellos. Los tipos de soporte existentes son:



Ilustración 1. Tipos de Servicios que ofrece el Centro de Soporte UCI.

Estos tipos de soporte permiten al cliente tener un servicio ajustado a sus características y son incluidos en los contratos legales que ambas partes firman para iniciar el soporte técnico. La cantidad de contratos varía constantemente debido a que se ejecutan por períodos de tiempo determinados (entre 6 meses y un año), por lo que las fechas de inicio y fin no coinciden.

Otro elemento importante de la contratación del centro radica en que más del 90% de los contratos son de tipo Profesional, debido a que incluye soporte in-situ, actividad considerada por el cliente como imprescindible. Este tipo de soporte tiene como característica particular que genera un alto

Sistema de planificación de visitas para el Centro de Soporte

número de visitas (asociado a los clientes con contratos en ejecución) por partes de los especialistas de soporte, que pueden variar su frecuencia en dependencia de los incidentes que ocurran. Además, es necesario acudir a las entidades para realizar actualizaciones a los sistemas, chequear el funcionamiento de ellos o efectuar el proceso de facturación.

La planificación de visitas en el Centro de Soporte es realizada manualmente por los directivos, asociada generalmente al cronograma de visitas anual que se realiza. Debido a que los especialistas son los que tratan directamente con los clientes, en ocasiones la planificación no está en correspondencia a las necesidades reales, desaprovechándose tiempo y recursos. Esta consecuencia incide, además, en la disponibilidad de estos recursos para visitas de urgencia que pueden surgir durante la semana de trabajo.

Otro elemento importante es la documentación de las visitas planificadas por parte de los involucrados y su correcta gestión. La organización de los encuentros se realiza de manera verbal o mediante correo electrónico, así como la información referente a los resultados de la visita. Este proceder perjudica la veracidad, confiabilidad y seguimiento de cada visita, afectando otros procesos vitales como la Gestión de Incidencias, la Gestión de Problemas y la Evaluación de Desempeño del especialista del centro. De manera general el estado actual de la gestión de visita afecta directamente en la organización del Centro de Soporte y la calidad del servicio que ofrece.

Teniendo en cuenta lo expuesto previamente se plantea como **problema a resolver**: ¿Cómo contribuir con la planificación de visitas a los clientes que contratan los servicios del Centro de Soporte?

Una vez identificado el problema, se plantea como **objeto de estudio**: los sistemas de planificación de visitas.

Teniendo como **campo de acción**: la planificación de visitas a los clientes que adquieren contratos con el Centro de Soporte.

Para dar solución al problema planteado se define como **objetivo general**: desarrollar una herramienta que permita la planificación de vistas para el Centro de Soporte.

Para guiar la investigación se definen las siguientes **preguntas científicas**:

1. ¿Cuáles son las bases teóricas que sustentan el desarrollo de los sistemas de planificación?
2. ¿Qué elementos asociados al desarrollo de software son necesarios para obtener la herramienta propuesta como solución?
3. ¿Cómo verificar el funcionamiento del sistema de planificación de visitas propuesto?

Para dar cumplimiento a la situación planteada se trazan como **tareas de la investigación** las siguientes:

Sistema de planificación de visitas para el Centro de Soporte

1. Elaboración del marco teórico que describen los principales conceptos y definiciones asociados a la investigación.
2. Elaboración del estado del arte asociado a las herramientas de planificación de visitas, así como las herramientas y tecnologías utilizadas en el desarrollo de la herramienta propuesta.
3. Diseño e implementación de la herramienta propuesta, de acuerdo a la metodología de desarrollo seleccionada.
4. Realización de las pruebas de software necesarias para verificar el cumplimiento del objetivo planteado.

Para desarrollar el presente trabajo de diploma se propone utilizar los siguientes **Métodos Científicos**

Métodos Teóricos

- **Analítico-Sintético:** Se utiliza para hacer un análisis bibliográfico del tema relacionado con los sistemas de planificación, con el objetivo de realizar una síntesis de los aspectos necesarios para llevar a cabo el desarrollo de la aplicación web con el objetivo de automatizar los sistemas de planificación de un entorno determinado.
- **Modelación:** Se utiliza para vincular el objeto de estudio a la investigación. Permite seleccionar que parte del objeto de estudio es de vital importancia y en esa dirección basar el análisis teórico de la investigación.

Métodos Empíricos

- **Observación:** Se utiliza para observar los procesos que se realizan en los sistemas ya existentes, para así recopilar los puntos más importantes que sean necesarios en la aplicación para obtener un producto final con la calidad necesaria.
- **Entrevista:** Se utilizó para recoger información referente al proceso de pruebas en el Centro de Soporte y mediante el criterio de los integrantes de los equipos de desarrollo entrevistados diagnosticar y detectar los problemas existentes.

Posibles resultados:

- Una herramienta que permita planificar las visitas a los clientes que han contratado los servicios del Centro de Soporte, para facilitar la gestión y organización de las visitas por parte de los especialistas a los clientes y contribuir a la calidad de los servicios que ofrece dicho centro.
- La documentación que sustenta la investigación, asociada a la metodología de desarrollo de software seleccionada.

El documento estará estructurado en tres capítulos:

Sistema de planificación de visitas para el Centro de Soporte

Capítulo I - Fundamentación Teórica: Se definen elementos teóricos que sustentan la investigación. En él se describen las tendencias actuales de la gestión de requisitos además de las herramientas y tecnologías para el desarrollo de la misma.

Capítulo II - Análisis y Diseño del Sistema: Se ofrece una propuesta de solución para dar respuesta a la situación presentada. En este capítulo es donde se realiza una descripción más detallada del sistema, se realizan las actividades de análisis y de diseño, realizando la construcción de prototipos no funcionales del sistema.

Capítulo III - Implementación y Pruebas: En este capítulo se diseñan y ejecutan casos de prueba y se corrigen las no conformidades encontradas para asegurar el funcionamiento del Sistema.

CAPÍTULO I: Fundamentación Teórica

1.1 Introducción

En el presente capítulo se definen los conceptos más relevantes asociados a la investigación, con el fin de introducir los términos necesarios para comprender la propuesta. Se realice una valoración de los sistemas de planificación como parte del estado del arte, estudiando los aportes teóricos y tecnológicos que tienen en el desarrollo del Sistema de Planificación de Visitas. El análisis de estos sistemas permite comprender la estructura básica de este tipo de software y sus funciones generales. Finalmente se realiza la descripción de las tecnologías y herramientas que intervienen en el proceso de desarrollo, con el propósito de comprender su funcionamiento y utilidad. De manera similar se estudia la metodología de desarrollo que guía cada etapa de construcción del producto final.

1.2 Conceptos fundamentales

1.2.1 Cliente

El término Cliente se asocia generalmente a la persona que accede a un producto o servicio por medio de una transacción financiera (dinero) u otro medio de pago. Aunque este concepto ha sido popularizado en la web, otras definiciones formales refieren al cliente como “la persona que solicita un servicio. Estos pueden ser clasificados en dos tipos: los clientes externos, que son los consumidores finales y los internos que son los trabajadores de una organización. En ambos la satisfacción es fundamental para la empresa” (Aguilar Morales, y otros, 2010).

Muy relacionada con esta definición se encuentra la de “persona que utiliza con asiduidad los servicios de un profesional o empresa, con el objeto de realizar un pedido o compra de un determinado producto o servicio” (ESIC, 2016). Este concepto, dado en el Manual de Técnicas Comerciales, realizado por el Programa de Desarrollo Comercial para Tecnólogos en Asturias, clasifica a los clientes en diferentes tipos, ayudando de esta forma a las organizaciones a crear estrategias que le faciliten el intercambio en cada caso.



Ilustración 2. Tipos de Clientes empresariales.

Sistema de planificación de visitas para el Centro de Soporte

Por su parte, el profesor peruano de marketing, William Estrada Vera describe al cliente como *“una persona impulsada por un interés personal y que tiene la opción de recurrir a una organización en busca de un producto o servicio. Según su definición los clientes no sólo están presentes en el campo comercial, empresarial o institucional, sino también en la política y en la vida diaria, relacionándolos con todo aquel que busca satisfacer una necesidad”* (Estrada Vera, 2007). Esta definición, según el autor, es la más completa y por tanto se toma como referencia para comprender el papel del cliente como parte de la solución que se propone.

1.2.2 Servicio al cliente

El cliente es el activo más valioso que posee una empresa. Son disímiles los conceptos asociados a los servicios, específicamente a los servicios al cliente. Para Ivan Thompson, servicio significa las *“actividades identificables que son el resultado de esfuerzos humanos o mecánicos que producen un hecho, que implican generalmente la participación del cliente y que no es posible poseer físicamente, ni transportarlos o almacenarlos”* (Thompson, Ivan, 2016)

Relacionado con el concepto descrito se puede definir al servicio como un bien intangible que se manifiesta mediante resultados a corto plazo y se sostiene en el trabajo de especialista de la rama a la que se asocia el servicio que se brinda (Lago Clara, y otros, 2016)

De manera general un servicio, según Van Jan Bon, es *“un medio para entregar valor a los clientes, facilitando los resultados que los clientes quieren asumir sin costes o riesgos específicos. Los resultados dependen de la realización de tareas y están sujetos a diversas restricciones. Los servicios mejoran el rendimiento y reducen el efecto de las restricciones, lo que aumenta la probabilidad de conseguir los resultados”* (Lago Clara, y otros, 2016).

Esta definición se puede encontrar referencia en artículos realizados para la metodología ITIL y el Manual de CMMI-SVC. Teniendo en cuenta la relación que existe entre la actividad de soporte y los estándares mencionados, se considera el concepto más aceptado de servicios para la presente investigación.

1.2.3 Gestión de Servicio TI

La gestión, como un concepto se conoce por el conjunto de guías para orientar la acción, previsión, visualización y empleo de los recursos y esfuerzos a los fines que se desean alcanzar, la secuencia de actividades que habrán de realizarse para lograr objetivos y el tiempo requerido para efectuar cada una de sus partes y todos aquellos eventos involucrados en su consecución (Benavides, 2011).

Relacionado con los servicios TI, Gil Gómez lo define como *“una disciplina basada en procesos, enfocada en alinear los servicios de TI proporcionados con las necesidades de las empresas, poniendo énfasis en los beneficios que puede percibir el cliente final”* (3C TIC, 2015). La Gestión de Servicios TI, en general, se encarga de *“organizar el uso y coordinación de personas, flujos de*

Sistema de planificación de visitas para el Centro de Soporte

trabajo y tecnologías de la información. Estos 3 componentes son el corazón de los procesos de negocio” (Roig, y otros, 2015)

Las definiciones presentadas, en su conjunto, representan la Gestión de Servicios TI en el Centro de Soporte por lo que pueden ser tomadas como punto de referencia para trazar los objetivos a alcanzar con el desarrollo de la solución propuesta. De esta manera se espera tributar a los logros del centro como entidad dedicada a los servicios de soporte.

1.2.4 Planificación

En entidades donde el resultado final depende de la ejecución de varias actividades procedentes de diferentes procesos, la organización es fundamental. Es por ello que la utilización del término Planificación ha ganado importancia tras las tendencias multidisciplinarias mostradas por las organizaciones.

Según Jiménez: *“La planificación es un proceso de toma de decisiones para alcanzar un futuro deseado, teniendo en cuenta la situación actual y los factores internos y externos que pueden influir en el logro de los objetivos” (Rendón Artola, y otros, 2016)*. Otros autores han definido el término de acuerdo a la aplicación en áreas específicas.

George Terry, por ejemplo, argumenta que *“la planificación es seleccionar información y hacer suposiciones respecto al futuro para formular las actividades necesarias para realizar los objetivos organizacionales”*. Idalberto Chiavenato, por su parte, identifica la planificación como *“una técnica para minimizar la incertidumbre y dar más consistencia al desempeño de la empresa” (Bernal, 2012)*.

La necesidad de planificación en la presente investigación es fundamental, por lo que los elementos mencionados en los conceptos tratados conforman un criterio general que permitirá encaminar los objetivos de la propuesta de desarrollo, con el fin de apoyar los procesos a los que tributa la gestión de las visitas en el Centro de Soporte.

1.2.5 Proceso

El trabajo orientado a procesos se ha acrecentado, teniendo en cuenta la organización que brinda y las facilidades de interrelación que poseen. La ISO 9001:2015 define al proceso como un conjunto de actividades que están interrelacionadas y que pueden interactuar entre sí. Estas actividades transforman los elementos de entrada en resultados, para ello es esencial la asignación de recursos (ISO 9001, 2015).

De manera similar se define en el Manual de Procesos y Procedimientos de la Universidad Politécnica de Valencia, donde refieren que la ejecución de estas actividades permite crear valor añadido al usuario (CalidadUPV, 2011). El concepto reflejado en este documento permite asumir que un proceso debe estar dirigido a una persona o grupo de ellas, que tengan un fin común, descrito en el propio proceso que utilizan.

Sistema de planificación de visitas para el Centro de Soporte

De manera general la mayoría de los artículos científicos y documentos consultados utilizan el concepto anterior como definición para el término proceso. Estos elementos, junto a la puesta en práctica del método científico de la observación, se puede decir que un proceso en el Centro de Soporte es un conjunto de actividades planificadas que implican la participación de un número de personas y de recursos materiales coordinados para conseguir un objetivo previamente identificado. Se estudia la forma en que el Servicio diseña, gestiona y mejora sus procesos (acciones) para apoyar su política y estrategia; y para satisfacer las necesidades de los clientes y otros grupos de interés.

1.3 Estudio del estado del Arte de los Sistemas de Planificación

1.3.1 Diagrama de Gantt

El diagrama de Gantt, nombre que recibió de su creador Herny Gantt, no constituye en sí mismo un sistema de planificación. Aun así, cuando se trata de planificar no puede dejarse de mencionar por su utilización, especialmente en la producción de software y tecnologías TI. Específicamente el diagrama de Gantt representa “una herramienta para planificar y programar tareas a lo largo de un período determinado”. Entre sus principales características se encuentra una fácil y cómoda visualización de las acciones previstas, lo que permite realizar el seguimiento y control del progreso de cada una de las etapas de un proyecto. Además, reproduce gráficamente las tareas, su duración y secuencia, incluyendo el calendario general del proyecto (Business School, 2016).

La construcción del diagrama, además de los elementos de planificación propios del proyecto, implica conocer el diseño básico que propone Gantt (Espinosa, 2003):

- Un eje horizontal: un calendario, o escala de tiempo definido en términos de la unidad más adecuada al trabajo que se va a ejecutar: hora, día, semana, mes, entre otros.
- Un eje vertical: las actividades que constituyen el trabajo a ejecutar. A cada actividad se hace corresponder una línea horizontal cuya longitud es proporcional a su duración en la cual la medición efectúa con relación a la escala definida en el eje horizontal conforme se ilustra.
- Símbolos convencionales: representaciones básicas para los elementos representados en el diagrama.

A partir de la utilidad demostrada de esta técnica se han desarrollado algunos programas computarizados que facilitan la elaboración de los diagramas de Gantt. Entre ellos se pueden mencionar el OpenProj, GanttProject, TaskJuggler, Planner, Microsoft Excel, entre otros.

La utilización de la técnica Gantt para planificar las visitas del Centro de Soporte no podría considerarse como la solución definitiva pues no constituye propiamente un sistema de gestión. En tal caso su presencia solo sería útil durante la elaboración del cronograma de visita que se realiza anualmente, pero no permite el almacenamiento de datos más profundos, como son los resultados

Sistema de planificación de visitas para el Centro de Soporte

específicos que se obtienen de las visitas. No obstante, la filosofía de administración del calendario de actividades y las funcionalidades asociadas a la reestructuración del cronograma, después de las modificaciones, pueden resultar útil para el autor en el desarrollo de la solución propuesta.

1.3.2 Sistemas ERP ⁶

Los sistemas ERP se consideran un conjunto de sistemas de información que permite la integración de ciertas operaciones de una empresa. Se compone de un conjunto de módulos que permite a las empresas automatizar e integrar las diferentes operaciones que se realizan en las diferentes áreas de la empresa (contabilidad, finanzas, ventas, compras, recursos humanos etc.). Un ERP captura la información de estas áreas y lo centraliza en una sola base de datos. Esto permite garantizar la integridad y unicidad de los datos a los que accede cada departamento, evitando que estos tengan que volver a ser introducidos en cada módulo (Huerta, y otros, 2015).

Definidos bajo estas características, son variados los productos ERP que existen y se comercializan en el mundo. Algunos de ellos poseen licencias privativas, pero existen otros de código abierto que pueden resultar útiles para las empresas que abogan por la soberanía tecnológica. La competencia del mercado en este tipo de plataformas es amplia y las productoras líderes en el desarrollo de herramientas ERP fluctúan en dependencia del momento y del alcance de cada una de ellas. Hasta el 2015 se listan nombres como SAP AG, Oracle-PeopleSoft y SSA Global. Específicamente en Cuba, la Universidad de las Ciencias Informáticas, incluye entre sus líneas de desarrollo, los productos de gestión empresarial, entre ellos los ERP, entre los que se puede mencionar CEDRUX.

De manera general, el alcance de estos productos suele ser su mayor ventaja, pues favorece la integración de múltiples procesos en la búsqueda de un mismo objetivo; así como la disponibilidad de información. Sin embargo esto aumenta sus costos (aun siendo código abierto⁷) por el número elevado de funcionalidades que se deben desarrollar.

“Algunos seguidores de este tipo sistemas, han observado que la sensación de las empresas respecto a los ERP, es que tienen un precio elevado y que no se utilizan todas sus funcionalidades. Asimismo, muchas veces se realiza una formación insuficiente por parte de la consultoría para ahorrar costes, por lo que como se ha comentado anteriormente, el personal que utiliza el ERP no lo domina y muchas funcionalidades quedan en desuso. Además, cuando se incorpora nuevo personal, los nuevos empleados no tienen opción a la formación” (Huerta, y otros, 2015).

⁶ Sistemas de planificación de recursos empresariales.

⁷ Se refiere a: software libre ≠ libre de pago.

Sistema de planificación de visitas para el Centro de Soporte

En el caso de aquellos ERP que son distribuidos bajo licencia GPL⁸, aunque es posible que sean adaptados a las necesidades esenciales de la empresa, suele ser complejo destinar un grupo de especialistas en desarrollo de software, con los conocimientos para asumir la reorganización de este tipo de productos. Este cometido suele ser largo y agotador, e implica generar gastos extras en términos de recursos, tiempo y esfuerzo. Desde el punto de vista de la solución propuesta, un ERP no sería la opción más viable, teniendo en cuenta las desventajas antes mencionada.

1.3.3 Sistemas de Planificación de la Producción

Similar a los ERP se han desarrollado sistemas dirigidos especialmente a la planificación de la producción en las empresas. Dentro de este grupo de producto se destacan los MRP (Planificación de Requerimientos de Materiales) y los HPP (Planificación Jerárquica de la Producción). Todas estas arquitecturas de sistemas tienen en común el objetivo de planificar los recursos de las empresas con el fin de aumentar las posibilidades de éxitos de la producción.

El sistema MRP transforma un Plan Maestro de Producción (MPS) en un programa detallado de necesidades de materiales y componentes requeridos para la fabricación de los productos finales utilizando, para ello, las listas de materiales. A pesar de poseer ediciones que perfeccionaban las funcionalidades de MPR I, el sistema sigue siendo dependiente del comportamiento regular de los elementos incluido en la planificación (Gestión Industrial, 2005). En caso de que algún paso establecido previamente fallara o se retrasara, o si las capacidades de la empresa cambiarán, el sistema no permite generar un plan alternativo de producción. Entre otras de sus desventajas se mencionan los elevados requerimientos computacionales que se necesitan para el procesamiento de la información de los planes.

Los HPP, por su parte, organizan la planificación por niveles jerárquicos. En esta línea, los modelos jerárquicos para la Planificación de la Producción descomponen el problema global de toma de decisiones en una serie de sub-problemas que corresponden a diferentes niveles de una jerarquía de planes. Estos sub-problemas pueden resolverse en secuencia, de modo que la solución en cada nivel impone restricciones sobre el problema del nivel inferior (enfoque de arriba-abajo) (Gestión Industrial, 2005). La falta de éxito de este tipo de sistema radica esencialmente en su propio mecanismo de trabajo, pues el exceso de jerarquías hace el sistema complejo, y la base matemática sobre el que está desarrollado suele ser difícil de comprender. Además de ello, posee otras desventajas técnicas que impiden que en todos los casos se obtenga el resultado esperado.

Ambos prototipos de sistemas no son factibles como solución a la problemática existente en el Centro de Soporte por los recursos de hardware que se necesitan para su puesta en marcha. Aun

⁸ es la licencia de derecho de autor más ampliamente usada en el mundo del software libre y código abierto, y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software

Sistema de planificación de visitas para el Centro de Soporte

cuando se pudieran obtener estos recursos, el desarrollo de sus funcionalidades está personalizado para planificar la producción de las industrias, por lo que no resultan útiles para el tipo de planificación que se necesita hacer en el Centro de Soporte.

1.3.4 Sistema de Planificación de Actividades

El SIPAC, como se denomina el Sistema de Planificación de Actividades, es una herramienta desarrollada bajo la Instrucción no.1 del Presidente de los Consejos de Estado y de Ministros para la Planificación de los objetivos y actividades en los órganos, Organismos de la Administración Central de Estado, entidades nacionales y Administraciones locales del Poder Popular. Está destinado a facilitar la gestión de las actividades a todos los niveles organizacionales, permite interrelacionar objetivos de trabajo y actividades en tiempo real; garantizando el seguimiento del desarrollo y cumplimiento de los objetivos y tareas en las entidades. Informatizando los procesos de Elaboración, Ejecución y Control del Plan, Evaluación de los objetivos y Puntualización de las actividades (CEIGE, 2015).

El propio equipo de desarrollo del sistema lo describe como: *“un producto que cuenta con varios módulos encargados de generar las configuraciones necesarias que otorgan al sistema y al cliente una simulación de los procesos de organización del personal, así como los niveles de subordinación necesarios e indispensables para efectuar una planificación de actividades basadas en reglas de la compartimentación de la información, permitiendo que la información planificada sea accedida por la persona autorizada, en el momento indicado. Puede catalogarse como una solución integral para la gestión de elementos de la planeación estratégica y operativa basada en actividades, objetivos, y planes, diseñada sobre las bases de la compartimentación de la información”* (CEIGE, 2015).

Este sistema tiene como ventaja que está desarrollado por la red de centros de la UCI, lo que facilita su acceso y soporte. Es un sistema basado en una línea de desarrollo estable y tiene como característica fundamental que es altamente configurable. A pesar de ello su adopción como solución al problema planteado no se considera oportuno pues la base del SIPAC es la planeación estratégica de los recursos humanos de las entidades. Sus funcionalidades están dirigidas a la creación y seguimiento de las actividades planificadas siguiendo la jerarquía establecida por las propias empresas, acorde a la estructura que posean. De manera general, el objetivo que persigue el SIPAC no tiene puntos de coincidencia con el objetivo planteado en la presente investigación.

1.4 Metodología de Desarrollo

Conceptualmente puede decirse que una metodología define artefactos, roles y actividades, junto con prácticas y técnicas recomendadas. Específicamente en el desarrollo de software se considera un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Comprende los procesos a seguir sistemáticamente para idear, implementar

Sistema de planificación de visitas para el Centro de Soporte

y mantener un software desde que surge la necesidad del producto hasta que se cumple el objetivo por el cual fue creado (Romero, 2014).

Las metodologías de desarrollo de software surgen con el objetivo de guiar los proyectos de desarrollo y organizar cada una de las etapas elementales en estos procesos. Desde su surgimiento hasta la actualidad se han creado varias metodologías debidas, fundamentalmente, a la necesidad de acercarlas a la forma de hacer de cada empresa. No obstante, las clasificaciones existentes no han variado, creando un marco común entre ellas. Estas clasificaciones son (Romero, 2014):

- Metodologías Tradicionales: Se caracterizan por su robustez, por su buena documentación en cada proceso, pero como desventaja tiene los altos costos que se generan al implementar un cambio.
- Metodologías Ágiles: Nace como una propuesta a los problemas o falencias de las metodologías tradicionales. Son más flexibles a los cambios en el desarrollo, pero en los procesos se tiene menos control.

Cuando se decide iniciar un proyecto de desarrollo nuevo, es fundamental realizar la selección de la metodología adecuada. Esta tarea precisa de un estudio esmerado y que permita profundizar en las características de cada metodología para asegurar que brinde los mayores beneficios posibles. En el caso de la presente investigación, la metodología que se utiliza para guiar el proceso de desarrollo del Sistema de Planificación de Visita es AUP (Proceso Unificado Ágil).

La decisión se basa en la existencia de una adaptación de esta metodología al sistema de trabajo UCI y permite seguir una estructura homogénea en el desarrollo de todos los productos de la red de centro de la universidad. Además, se tiene acceso a la documentación relacionada con la metodología, existe experiencia sobre su utilización y se conoce, a través de resultado tangibles, la factibilidad de su utilización. Como punto relevante se puede mencionar que AUP UCI es altamente adaptable, permitiendo combinar las características de los marcos robustos con los ágiles. Esto posibilita que el producto sea entregado en el menor tiempo posible sin que se descuide la calidad final del software.

1.4.1 Proceso Unificado Ágil (AUP)

Creada por Scott Ambler, la metodología AUP es considerada una versión simplificada del Proceso Unificado Racional (RUP). AUP describe, de una manera simple y fácil de entender, la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. Entre las técnicas ágiles que incluye se encuentra el Desarrollo Dirigido por Pruebas, el Modelado ágil, la Gestión de Cambios y la Refactorización de Base de Datos para mejorar la productividad.

Sistema de planificación de visitas para el Centro de Soporte

La estructura de esta metodología se organiza en cuatro fases y define 7 disciplinas, de ellas cuatro a los procesos ingenieriles y tres dedicados a la Gestión de Proyectos. Similar a otras metodologías AUP describe roles generales para el desarrollo entre los que se encuentran (Programa de Mejora, 2014):

1. **Administrador de proyecto:** Encargado de planear, manejar y asignar los recursos.
2. **Ingeniero de procesos:** Desarrolla, adapta y apoya en del proceso de desarrollo del software.
3. **Desarrollador:** Escribe, testea y construye el software.
4. **Administrador de Base de Datos (BD):** Diseña, prueba, desarrolla, y apoya los esquemas de BD.
5. **Modelador ágil:** Crea y desarrolla modelos, bosquejos o los archivos de la herramienta CASE.
6. **Administrador de la configuración:** Encargado de la configuración, es responsable de proporcionar la infraestructura total para el equipo de desarrollo.
7. **Stakeholder⁹:** Representa los intereses de los clientes del producto.
8. **Administrador de pruebas:** Responsables del éxito de la prueba, incluyendo el planeamiento, la gerencia, y la defensa para la prueba y las actividades de la calidad.
9. **Probador:** Encargado de ejecutar las pruebas.

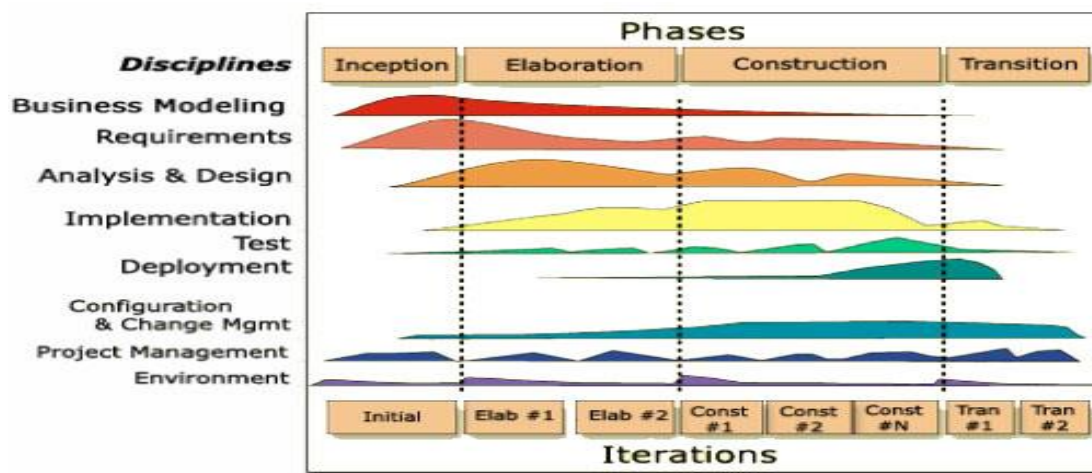


Ilustración 3. Fases e iteraciones de Proceso Unificado Ágil.

La metodología AUP, en la Universidad de las Ciencias Informática, posee una variación que se adecua a las características específicas de esta entidad productora de software. Esta variación se realizó de acuerdo al ciclo de vida productivo de la UCI y se apoyó en las buenas prácticas que propone el modelo CMMI-DEV 1.3¹⁰. De acuerdo a estas especificaciones se decidió “mantener la

⁹ Se refiere a las partes interesadas.

¹⁰ Es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software.

Sistema de planificación de visitas para el Centro de Soporte

fase de Inicio, pero modificando su objetivo, se unifican las restantes tres fases de AUP¹¹ en una sola, denominada Ejecución y se agrega la fase de Cierre.

Las disciplinas, por su parte, fueron ampliadas a 8 y algunas de ella se modificaron con el fin de abarcar la mayor parte de los procesos vitales de la producción de software en la UCI. Además, se adaptaron algunas de ellas para su compatibilidad con áreas de procesos específicas de CMMI-DEV. En tal caso se determinó definir las de la siguiente manera (Programa de Mejora, 2014):

Disciplinas AUP	Disciplinas Variación AUP-UCI	Objetivo
Modelo	Modelado del Negocio	Destinada a comprender los procesos de negocio de una organización.
	Requisitos	Desarrolla un modelo del sistema que se va a construir. Comprende la administración y gestión de los requisitos funcionales y no funcionales del producto.
	Análisis y diseño	Modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales.
Implementación	Implementación	A partir de los resultados del Análisis y Diseño se construye el sistema.
Prueba	Pruebas interna	Verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas.
	Pruebas de liberación	Pruebas diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.
	Pruebas de Aceptación	Prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.
Despliegue	Despliegue	Constituye la instalación, configuración, adecuación, puesta en marcha de soluciones informáticas y entrenamiento al personal del cliente.
Gestión de configuración	Se cubren con las áreas de procesos PP, PMC y CM que propone CMMI-DEV v1.3. Las mismas son áreas de procesos de gestión y soporte respectivamente.	De acuerdo con los objetivos definidos por CMMI-DEV para estas áreas de procesos
Gestión de proyecto		
Entorno		

Tabla 1. Disciplinas de la metodología AUP, según la variación UCI.

¹¹ Elaboración, Construcción y Transición.

Sistema de planificación de visitas para el Centro de Soporte

De manera similar se decidió ampliar los roles definidos en AUP de 9 a 11, agregando nuevos y unificando algunos de los existentes. A pesar de lo cambios realizados se mantuvieron las características de la metodología de ser iterativa e incremental.

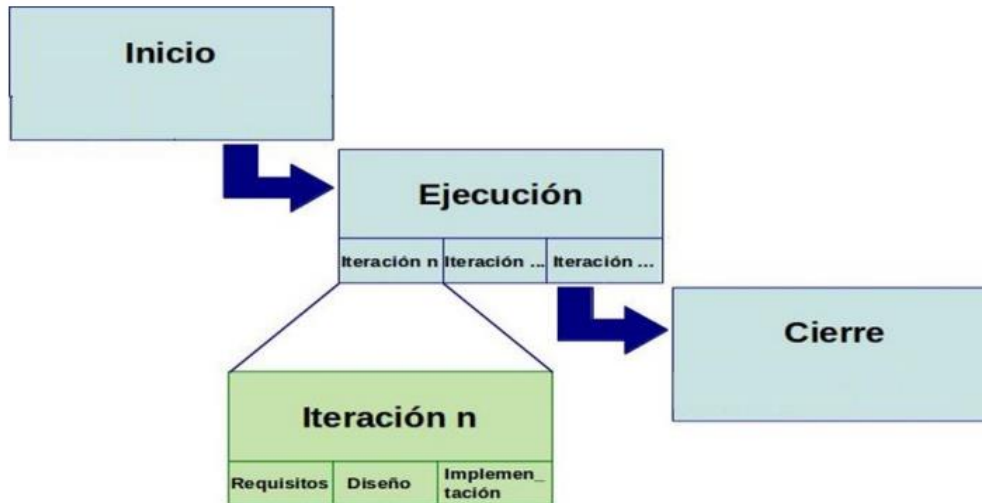


Ilustración 4. Fases e iteraciones de la variación AUP-UCI.

1.5 Herramientas y tecnologías

La selección de las herramientas y tecnologías en las que se apoya el desarrollo del Sistema de Planificación de Visitas del Centro de Soporte está en correspondencia con las solicitudes del cliente. Estas especificaciones se realizaron teniendo en cuenta la aplicación de ellas al contexto de trabajo del sistema, la vigencia en las comunidades de desarrollo y las facilidades funcionales que aportan al desarrollo del producto.

1.5.1 Lenguajes de programación

En el desarrollo de la solución propuesta se utilizan dos tipos de lenguaje de programación: uno del lado del cliente y otro en el servidor. Esta estructura se debe a que el sistema que se desarrolla es totalmente web, por lo que debe tenerse en cuenta los lenguajes para estructurar, organizar y representar de manera amigable los contenidos que se almacenan y consultan. Ambos lenguajes deben ser aceptados entre ellos, poseer parámetros de comunicación mutuos y complementarse para alcanzar el mismo objetivo.

Lenguaje del lado del cliente

El lenguaje de programación JavaScript 1.8.2 se utiliza para el manejo de las interfaces, los diálogos de notificaciones y las validaciones en el lado del cliente. Su función en el desarrollo de la solución está asociada a que es “un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Se considera un lenguaje interpretado, por lo que no es necesario

Sistema de planificación de visitas para el Centro de Soporte

compilar¹² los programas para ejecutarlos. Esto significa, de manera breve, que los productos desarrollados con JavaScript pueden ser probados en cualquier navegador sin necesidad de procesos intermedios” (Eguíluz Pérez, 2009).

Actualmente, todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web. De manera general, JavaScript permite crear aplicaciones basadas en páginas HTML dinámicas, que procesen la entrada del usuario y que sean capaces de gestionar datos persistentes usando objetos especiales, archivos y bases de datos relacionales. Además, se pueden construir aplicaciones que varían desde sencillos productos informativos hasta grandes sistemas de gestión, incluyendo aquellos que manejan transacciones de comercio electrónico. La versión que se utiliza en la investigación está relacionada con la que posee el *framework*¹³ AngularJS, seleccionado para apoyar el desarrollo en este lenguaje.

HTML5 y CSS 3

HTML5 es una tecnología creada para modernizar la web y el desarrollo de aplicaciones web, en línea y sin conexión, que conforma una colección de estándares para el diseño y desarrollo de páginas web. La versión 5 ha prestado especial atención a la definición de criterios para mejorar la interoperabilidad. HTML5 ofrece nuevas herramientas y posibilidades para abordar los problemas comunes a los que se enfrentan los desarrolladores web en la Internet más interactiva de hoy en día. Entre ellas se encuentra el uso de tecnologías web basadas en estándares para crear aplicaciones y sitios interactivos ricos en gráficos sin tener que utilizar herramientas especializadas ni escribir código específico del navegador. Incluye un conjunto de etiquetas nuevas que potencian sus ventajas, fundamentalmente que tributa al desarrollo de animaciones (Reyes Jiménez, y otros, 2014).

CSS 3, por su parte, se encarga de mejorar las interfaces y el comportamiento del contenido en la web, de forma tal que lo escrito dentro de etiquetas HTML pueda ser organizado de manera más atractiva y de acuerdo a las expectativas, tanto del desarrollador como del cliente. Las Hojas de Estilo en Cascada, se considera “un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. Es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Entre sus principales ventajas se encuentran (Musciano, Chuck, 1999):

- Las hojas de estilos gestionan todo lo concerniente a la apariencia, dejando al HTML la función de estructura y de codificar la información bruta.
- Contribuyen a despertar la creatividad de los usuarios aportando nuevas posibilidades de

¹² Traducir un lenguaje de alto nivel a código absoluto o lenguaje binario.

¹³ estructura conceptual y tecnológica de soporte definido, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Sistema de planificación de visitas para el Centro de Soporte

presentación ausentes en el código HTML.

- Simplificación del código: permiten aligerar considerablemente el código fuente volviéndolo más legible y accesible.
- Permiten garantizar un mantenimiento al sitio más fácil y con mayor rapidez.

Lenguaje del lado del servidor

PHP (de Hypertext Preprocessor) es un lenguaje de lado servidor que permite embeber fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz. Desde su creación en 1994 este lenguaje ha alcanzado una gran aceptación en la comunidad de desarrolladores, debido a la potencia y simplicidad que lo caracterizan, así como al soporte generalizado en la mayoría de los servidores de hospedaje tecnológico (Alvarez, y otros, 2013). La versión de PHP que se utiliza en el desarrollo de las funcionalidades del Sistema de planificación de visitas es la 7.1.

Alguna de las ventajas que posee son (Vázquez Meriño, 2008):

- Se ejecuta en casi cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado, por ejemplo, en diferentes versiones de Unix y Windows.
- La sintaxis de PHP es similar a la del C, por lo que puede ser aprendido fácilmente por desarrolladores con experiencia en este lenguaje o en C++
- Posee un amplio conjunto de módulos y una variedad de extensiones de código, por lo que permite desarrollar casi todo tipo de funcionalidades.
- Es soportado por variados servidores web, como Apache, Netscape, Nginx, entre otros.
- Puede interactuar con muchos motores de bases de datos tales como MySQL, Oracle, PostgreSQL, y otros.
- Es distribuido como un lenguaje *Open Source*, lo cual significa que el usuario no depende de una compañía específica para corregir las funcionalidades que presenten algún inconveniente.

1.5.2 Entorno de Desarrollo Integrado

El desarrollo de software ha conformado, en la medida que ha ido evolucionando, un conjunto de herramientas para facilitar la utilización de los lenguajes de programación. Además de los framework, se han creado los Entornos de Desarrollo, conocidos como IDEs, los cuales permiten incluir y compilar el código fuente. Los IDEs son desarrollados de forma tal que puedan ser utilizados para distintos lenguajes y poseen características especiales para organizar el trabajo durante la implementación. El desarrollo de la solución propuesta se realiza en el IDE NetBeans, teniendo en cuenta la compatibilidad tanto con el lenguaje de programación como con el framework.

NetBeans 8.1

Sistema de planificación de visitas para el Centro de Soporte

NetBeans es una herramienta de código abierto, distribuida bajo la licencia GPL, que permite el desarrollo de aplicaciones con características modulares. Entre sus funcionalidades se encuentran las facilidades para escribir, depurar, compilar y ejecutar programas. NetBeans soporta varios lenguajes de programación entre los que se encuentra PHP, siendo el más representativo en el IDE. Se destacan funcionalidades como el auto-completamiento de código, la integración con diferentes frameworks de desarrollo. Es un software multiplataforma y extensible mediante librerías que además posee abundante documentación referente a las clases y subprogramas que lo conforman (García, y otros, 2013).

Además de la amplia documentación disponible sobre el IDE, la selección está asociada las facilidades incluidas a partir de la versión 7.3. De esta versión en adelante el soporte de JavaScript ha sido reescrito desde cero, se introdujeron nuevas características para apoyar y mejorar la experiencia de desarrollo con aplicaciones web del lado del cliente que utilizan la familia de tecnologías de HTML5. El apoyo incluye coloración específica de sintaxis, autocompletado de código y otras características disponibles. También permite controlar las opciones de formato para el lenguaje JavaScript en sí (Reyes Jiménez, y otros, 2014).

1.5.3 Marco de Trabajo (Framework)

La utilización de *frameworks* o marcos de trabajo en el desarrollo de software es una tendencia altamente aceptada por la comunidad mundial, teniendo en cuenta la organización que brinda al código fuente (Riehle, Dirk, 2000). Esta práctica de programación apoya la reutilización, la fácil modificación de las funcionalidades de los programas y la inclusión de patrones de diseño en la implementación de software. En la medida que estos productos han evolucionado, el conjunto de funcionalidades que incluyen son mayores y la compatibilidad entre diferentes marcos de trabajo favorece la dinámica del desarrollo de software, permitiendo ganar en tiempo y aprovechar mejor los recursos.

En la presente investigación se utilizan los *frameworks* Symfony 2.8, para PHP, y *AngularJS* 1.5 para el lenguaje JavaScript.

Symfony

Symfony es un marco de trabajo diseñado principalmente para optimizar el desarrollo de las aplicaciones web, separando la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Brinda varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador centrarse en los aspectos específicos de cada aplicación (Reyes, y otros, 2014).

El *framework* está desarrollado completamente con PHP 5 por lo que se considera 100% compatible con este lenguaje. Su utilización ha estado presente en numerosos proyectos reales, por lo que existen garantías de que su uso es efectivo en el desarrollo de software. Symfony es

Sistema de planificación de visitas para el Centro de Soporte

compatible con la mayoría de los sistemas gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft y se puede ejecutar tanto en plataformas Unix, como en plataformas Windows.

Algunas de las principales características que hacen de Symfony una buena elección para el desarrollo de software son (Musciano, Chuck, 1999):

- Fácil de instalar y configurar en la mayoría de plataformas.
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Sigue la mayoría de las mejores prácticas y patrones de diseño para la web
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer y que permite un mantenimiento muy sencillo.

AngularJS

Es un proyecto de código abierto, realizado en *Javascript* que contiene un conjunto de librerías útiles para el desarrollo de aplicaciones web y propone una serie de patrones de diseño para llevarlas a cabo. En pocas palabras, es lo que se conoce como un *framework* para el desarrollo, en este caso sobre el lenguaje *Javascript* con programación del lado del cliente (Basalo, y otros, 2016).

Algunas de las principales ventajas que propone AngularJS son:

1. Mejoras del HTML: Permite al usuario crear HTML enriquecido, basado en una semántica clara que se fácilmente comprensible.
2. Promueve patrones de diseño adecuados para aplicaciones web: Permite aplicar la estructura por capas, aplicando MVC del lado del cliente de manera que se separen las responsabilidades.
3. Manejo potente, sencillo y extensible: AngularJS se conforma totalmente de librerías y servicios web de terceros que facilitan lograr los objetivos. Además, permite la incorporación de nuevos complementos de ser necesario.

1.5.4 Herramienta para el modelado

Un paso importante antes de iniciar la implementación de cualquier programa informático es diseñar los elementos esenciales que lo conforman. Esto apoya el proceso de desarrollo y fortalece la calidad del producto. El análisis y diseño de las características del software permite que se tengan en cuenta relaciones, tipos de datos y funciones que puedan omitirse durante la construcción y que generan fallos en el funcionamiento final del sistema.

Sistema de planificación de visitas para el Centro de Soporte

Lenguaje de Modelado UML 2.0

En el ámbito del diseño de los diagramas que representan los programas que se desean desarrollar, uno de los lenguajes más conocidos es el UML (Lenguaje Unificado de Modelado). Este es un lenguaje gráfico elaborado para visualizar, especificar, construir y documentar un sistema de software. Permite reducir el ámbito del problema de estudio al enfocar solo un aspecto a la vez. UML ofrece un estándar para describir modelos del sistema, incluyendo aspectos conceptuales tales como procesos de negocios, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables (Carbonell Perú, 2013).

Entre las principales ventajas de este lenguaje se encuentran (Espinosa, 2013):

- Es un lenguaje distribuido y adecuado a las necesidades de conectividad actual y futura.
- Es ampliamente utilizado por la industria.
- Reemplaza a decenas de notaciones empleadas con otros lenguajes de modelado.
- Modela estructuras complejas.
- Soporta el modelado orientado a objetos, tales como: objetos, clases, componentes y nodos.
- Emplea operaciones abstractas como guía para variaciones futuras, añadiendo variables si es necesario.
- Recrear el comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboración, que sirven para evaluar el estado de las computadoras.

Visual Paradigm 8.0

Teniendo en cuenta la necesidad demostrada de realizar un mejor diseño preliminar de las funcionalidades del software, la utilización del modelado UML ha ganado relevancia. Con ello las Herramientas CASE también han ocupado un lugar importante, representando las opciones de utilizar de manera visual los elementos que UML propone. Dentro de estas herramientas se encuentra el Visual Paradigm, popularizada por su estabilidad, la disponibilidad y la facilidad en el trabajo con el lenguaje UML.

Visual Paradigm es una herramienta CASE muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Este software modela el UML para la construcción de aplicaciones, permitiendo dibujar diagramas UML. Fue creado para el ciclo vital completo del desarrollo del software, que lo automatiza y acelera, facilitando la captura de requisitos, análisis, diseño e implementación (Carbonell Perú, 2013).

Entre sus principales características se encuentran (Ortiz Ricardo, y otros, 2014):

- Modelado de Caso de uso.

Sistema de planificación de visitas para el Centro de Soporte

- Generación de reportes.
- Ingeniería inversa de código Java, .NET, PHP5 y Python.
- Ingeniería inversa de código a modelo y de código a diagrama.

1.5.5 Sistema Gestor de Base de Datos

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD¹⁴ y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y sus últimas versiones se encuentran al nivel de otras bases de datos comerciales. PostgreSQL 9.2 utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Entre sus características más relevantes se encuentran (PostgreSQL, 2013):

- Es una base de datos 100% ACID¹⁵.
- Mantiene la integridad referencial.
- Posee Espacio de Tablas.
- Permite la replicación asincrónica/sincrónica.
- Permite que se realicen copias de seguridad en caliente.
- Es Unicode.
- Contiene juegos de caracteres internacionales.
- Permite regionalizar el contenido por columna.
- Es multiversión.
- Permite múltiples métodos de autenticación.
- Acceso encriptado vía SSL¹⁶.

1.6 Conclusiones parciales

El estudio de los conceptos asociados a la Gestión de Servicios y la planificación permitió comprender términos usualmente tratados durante la investigación. Facilitó la búsqueda bibliográfica, aportando términos concretos que acercaran a la temática con el fin de comprender la utilización de estas definiciones en el ámbito del objeto de estudio propuesto. El análisis de sistemas de planificación posibilitó asegurar la necesidad de continuar con el desarrollo de la solución, aunque se pudo tomar de base la estructura de trabajo que proponen los Diagramas de GANTT para organizar el cronograma de visitas y las modificaciones en los calendarios asociados a este cronograma. Como parte del análisis de las tecnologías de desarrollo se decidió utilizar aquellas que el cliente solicitó. Durante el estudio de estas tecnologías se corroboró su utilidad, la

¹⁴ Licencia que permite el uso del código fuente en software no libre.

¹⁵ Acrónimo de Atomicidad, Consistencia, Aislamiento, Durabilidad. Se le denomina ACID a los sistemas de base de datos que cumplen con estas características.

¹⁶ Seguridad de la Capa de Transporte son los protocolos criptográficos que proporcionan comunicaciones seguras por la red.

Sistema de planificación de visitas para el Centro de Soporte

facilidad que aportan al desarrollo y las múltiples funcionalidades que permiten obtener un resultado con calidad. Finalmente se selecciona la metodología AUP en su variación UCI por su adecuación a los procesos productivos de la universidad, su aporte reconocido y la relación que mantiene con el modelo CMMI-DEV. Todo esto permite fortalecer el proceso de desarrollo y obtener un Sistema de Planificación de Visitas que esté en correspondencia con las necesidades del Centro de Soporte.

CAPÍTULO II: Análisis y Diseño del sistema

2.1 Introducción

Cuando se inicia el desarrollo de un sistema informático es necesario realizar el análisis de las particularidades del producto y diseñar previamente las funcionalidades más importantes. De acuerdo a la metodología AUP, en su variación UCI, el desarrollo del software puede realizarse mediante prácticas ágiles o robustas, dependiendo de las características del sistema y el equipo de desarrollo. En el caso de la solución propuesta se utiliza la variante ágil, a partir de la cual se realiza la descripción de los requisitos funcionales y no funcionales mediante la Lista de Reserva del Producto (LRP). Los requisitos funcionales son ampliados en las Historias de Usuario como funcionalidades del sistema de planificación de visitas, siendo organizados en la línea de tiempo del desarrollo del producto. Además, se diseñan las clases y relaciones más importantes dentro del sistema, y se identifican los principales patrones de diseño mediante los cuales se garantiza la presencia de buenas prácticas de programación. De manera general, durante el capítulo, se expondrán los elementos más relevantes obtenidos sobre el dominio del negocio.

2.2 Modelo de Dominio

Es un modelo conceptual que se realiza cuando los procesos a ejecutar no son del todo visibles, ya que no se pueden identificar con claridad los actores y trabajadores del negocio en un problema específico. En él se describen las distintas entidades del problema, en forma de conceptos modelados por objetos y sus relaciones, además de las restricciones que rigen el dominio del problema.

En el Sistema de Planificación de Visitas el Centro de Soporte brinda soporte técnico a las a las aplicaciones desarrolladas por la UCI que necesitan de este servicio, las cuales son vendidas a varios clientes. A su vez el centro planifica las visitas a los clientes, las cuales son ejecutadas por los especialistas del centro. Una visita tiene varios elementos que la conforman, como el estado, los involucrados y el resumen, entre otros. Es importante destacar que para que exista la visita debe estar asociada obligatoriamente a un cliente que haya contratado los servicios del Centro de Soporte. Las visitas a su vez generan uno o varios reportes para su correcta gestión por parte de los directivos del Centro de Soporte.

Sistema de planificación de visitas para el Centro de Soporte

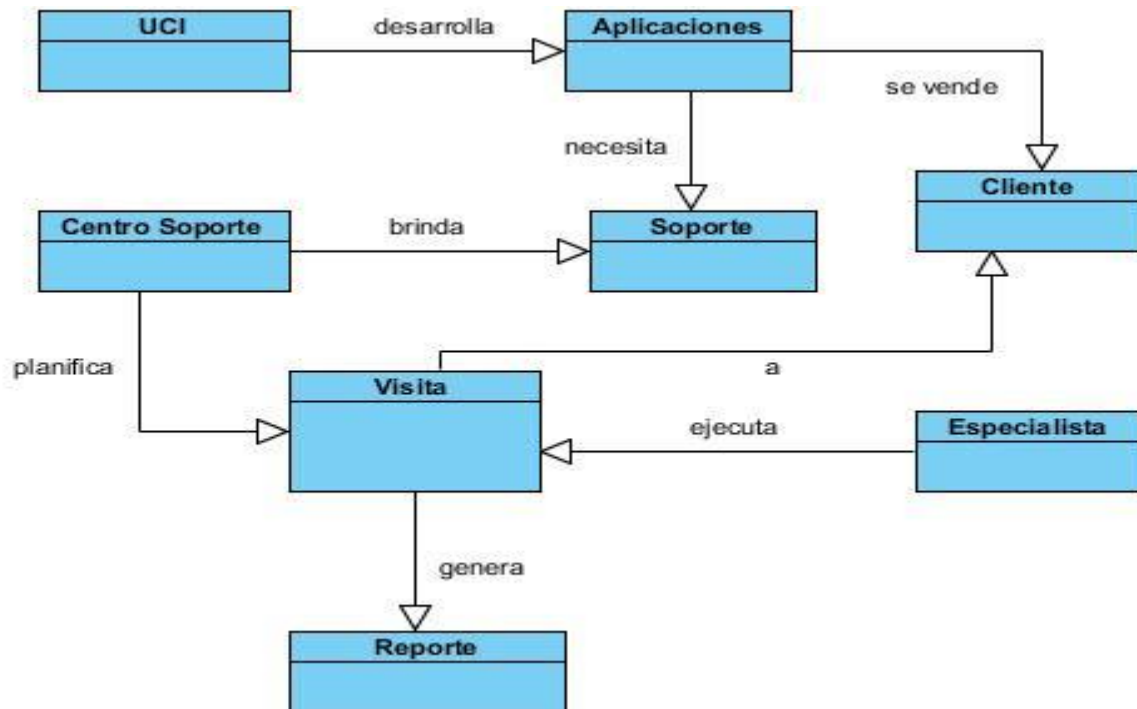


Ilustración 5. Diagrama de Dominio del Sistema.

2.3 Solución Propuesta

El Centro de Soporte de la Universidad de la Ciencias Informáticas tiene como misión: “Brindar el servicio de soporte técnico a las aplicaciones y servicios informáticos desarrollados por la Universidad de las Ciencias Informáticas con calidad y eficiencia a partir de una correcta gestión de los mismos y garantizando elevados niveles de satisfacción de sus clientes” (Soporte UCI, 2016). Es por ello que la atención a clientes juega un papel primordial en su funcionamiento diario, lo cual incluye las visitas a las entidades que contratan el servicio de soporte.

Al observar la forma en la que se planifican las visitas a clientes se detectó que no existe un sistema informático que controle este proceso. Lo que trae como consecuencia la ausencia de información fidedigna y controlada sobre las visitas efectuadas. Para dar solución a estas deficiencias se propone crear una herramienta informática que, en primer orden, este acorde a la tendencia de la universidad de desarrollar sobre tecnologías libres y ser un sistema multiplataforma. Además, se informatizará el proceso de visitas a clientes, evitando conflictos entre visitas a entidades en un mismo día y se podrá tener mayor control de la información resultante del proceso. Por medio de este sistema, el centro de soporte incrementará el valor agregado de sus servicios al ser más organizado y brindar el soporte in-situ con mayor calidad.

2.4 Concepción del Sistema

La estructura, el diseño y las funcionalidades que debe ejecutar el sistema para alcanzar su objetivo, son vitales para entender el curso de la investigación. Es asociado a esta idea que se

Sistema de planificación de visitas para el Centro de Soporte

hace preciso la gestión documental de la concepción del sistema, mostrando los principales detalles del producto de software que se propone como solución. Esto permitirá un mayor entendimiento del proyecto, no solo para su desarrollo, sino también para el cliente y la creación de futuras versiones.

2.4.1 Visión y alcance del sistema

El sistema de planificación de visitas contendrá como objetivos fundamentales:

- La planificación de visitas a clientes.
- El control del estado de las visitas planificadas en cada momento.
- La formulación de reportes asociados a las visitas.

Para el cumplimiento de estos objetivos se implementan funcionalidades óptimas desde el punto de vista del código fuente, que permitan una fácil interacción y una respuesta rápida a las peticiones de los usuarios. En caso de ser necesario las funciones del producto pueden ampliarse o modificarse, para responder a la planificación de visitas de otro centro o entidad.

2.5 Especificación de requisitos

Cuando se elabora un producto de software, es preciso satisfacer un grupo de necesidades que sirven como objeto de su creación. Estas dificultades encarecen o dificultan el trabajo de un cliente, por lo que es preciso ser muy cuidadoso cuando se elabora un producto, ya que sus funcionalidades son en concreto la respuesta a esas necesidades, las cuales son plasmadas por las metodologías bajo el nombre de requisitos o requerimientos (Turner, Raymond, 2005).

Para el ciclo de vida de la metodología AUP-UCI los requisitos pueden ser encapsulados dependiendo de las características del proyecto, por lo que se define como artefacto a utilizar para registrar los requisitos del sistema, la Lista de Reserva del Producto (LRP). Esta facilitará que una vez definidos los requisitos, se realice una estimación previa del costo en tiempo que implicará cada requisito para el desarrollador. Debido a que la LRP se genera al inicio del proyecto, como consciencia de la información que recoge, puede sufrir cambios durante el desarrollo del producto o respecto al equipo de proyecto. No obstante, los datos almacenados en este artefacto permiten tener una visión amplia y estructurada, de la concepción general del sistema.

2.5.1 Lista de reserva del producto

Prioridad	Ítem *	Descripción	Estimación	Estimado por
Muy Alta				
	RF1	Autenticar usuario	2	Analista
	RF2	Asignar roles.	2	Analista
	RF3	Adicionar visita.	4	Analista
	RF4	Asignar responsable.	2	Analista
	RF5	Notificar visita.	2	Analista

Sistema de planificación de visitas para el Centro de Soporte

RF6	Notificar nueva visita	2	Analista
RF7	Notificar Visita modificada	2	Analista
RF8	Notificar visita eliminada	2	Analista
RF9	Modificar una visita.	4	Analista
RF10	Eliminar una visita.	2	Analista
RF11	Brindar Apis Res de visitas.	2	Analista
RF12	Crear un resumen de vista.	2	Analista
RF13	Cambiar el estado de la visita,	4	Analista
RF14	Alertar sobre visita efectuada sin resumen.	2	Analista
RF 15	Consultar visitas planificadas.	2	Analista
RF 16	Realizar búsquedas de visitas.	5	Analista
RF 17	Mostrar datos de las visitas buscadas.	5	Analista
RF 18	Paginar los resultados de las búsquedas.	3	Analista
RF 19	Búsqueda avanzada de visitas	4	Analista
RF20	Paginar resultados de búsqueda avanzada	3	
RF21	Realizar informes de visitas	2	Analista
RF 22	Crear informes.	3	Analista
RF 23	Exportar informes a PDF.	2	Analista
RF 24	Exportar resumen de las visitas ejecutas a PDF	2	Analista
RF 25	Graficar los resultados.	3	Analista
RF 26	Actualizar categorías.	4	Analista
RF 27	Adicionar cliente	4	Analista
RF 28	Modificar cliente	4	Analista
RF 29	Eliminar cliente	4	Analista
RF 30	Adicionar transporte	4	Analista
RF 31	Modificar transporte	4	Analista
RF 32	Eliminar transporte	4	Analista
RF 33	Adicionar tipo de visita	4	Analista
RF 34	Modificar tipo de visita	4	Analista
RF 35	Eliminar tipo de visita	4	Analista
RF 36	Adicionar estado	4	Analista
RF 37	Modificar estado	4	Analista
RF 38	Eliminar estado	4	Analista
RF 39	Adicionar sistema	4	Analista
RF 40	Modificar sistema	4	Analista
RF 41	Eliminar sistema	4	Analista

Media

Baja

Requisitos No Funcionales

Usabilidad	<ul style="list-style-type: none"> La terminología del menú debe ser constante en todo el sitio. El sistema será utilizado por todo el personal del centro que tenga acceso a la red del centro y necesite hacer uso del mismo. 		
Rendimiento	<ul style="list-style-type: none"> El sistema debe responder en un tiempo menor de los 10 segundos. El sistema deberá soportar una conexión simultánea de al menos 100 usuarios. 		
Disponibilidad	<ul style="list-style-type: none"> Los usuarios deben tener acceso a la información 		

Sistema de planificación de visitas para el Centro de Soporte

	desde cualquier dispositivo sin que los mecanismos utilizados para la seguridad de los datos retrasen la obtención de los mismos.		
Restricciones de diseño	<ul style="list-style-type: none"> • Lenguajes de desarrollo: Simfony 2.8 con PHP 7.1, CSS 3, HTML5, AngularJs 1.5 como framework de JavaScript. • Como IDE se empleará NetBeans 8.1. • Como servidor web Apache 2.4 • El sistema gestor de base datos PostgreSQL 9.3. • El sistema operativo a usar en el entorno de desarrollo Nova 5. • Los artefactos del análisis se realizarán con Visual Paradigma 8.0. 		
Software	<ul style="list-style-type: none"> • En los ordenadores donde se encuentre instalado se recomienda como servidor de aplicación Debian 6 o superior. • A los clientes que acceden al sistema se les recomienda para el acceso Mozilla Firefox 40 en adelante. • El servidor de base de datos datos puede estar alojado en la misma servidor de aplicación, contando con espacio disponible para almacenar los datos.(50 GB) 		
Hardware	<ul style="list-style-type: none"> • En virtualización al menos 2 núcleos con 2 GB.de RAM. • En una PC física Al menos dual core a 2.8 GHZ con 2 GB de RAM. • Monitor VGA o superior. • Espacio en disco duro de 10 GB. 		
Portabilidad	<ul style="list-style-type: none"> • El sistema debe ser multiplataforma. 		

Tabla 2. Requisitos funcionales y no funcionales.

2.5.2 Historias de usuario

Una vez que los requisitos son establecidos y estimados es necesario que sean descritos minuciosamente para facilitar su desarrollo. Las metodologías ágiles, generalmente, proponen las Historias de Usuarios (HU) como el mecanismo para gestionar los requisitos, aplicándose de manera similar en AUP UCI en su versión ágil. Las HU “se escriben desde la perspectiva del cliente, aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas. El contenido de estas debe ser concreto y sencillo. El tratamiento de las HU es muy dinámico y flexible, en cualquier momento pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada HU es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas pocas semanas” (Pozo Guevara, 2014).

A continuación, se muestran siete Historias de Usuario. El resto se encuentran en la sección de los Anexos (ver [Anexo #1](#)).

Sistema de planificación de visitas para el Centro de Soporte

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Autenticar usuario
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoel Antonio Casado Gonzáles	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 0.4 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 0.4 semanas
Descripción: Permite el acceso de los usuarios del centro de soporte al sistema de planificación de visitas.	
Observaciones:	
<ul style="list-style-type: none"> - El sistema debe estar iniciado y el cliente desde el que se accede estar conectado a red. - El usuario autenticado debe ser un usuario del sistema. 	
Prototipo de interfaz:	
	

Tabla 3. Historia de Usuario "Autenticar usuario"

Historia de Usuario	
Número: HU_2	Nombre Historia de Usuario: Asignar roles
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoel Antonio Casado Gonzáles	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 0.4 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 0.4 semanas

Sistema de planificación de visitas para el Centro de Soporte

Descripción: Permite asignar permisos específicos a los usuarios, para delimitar las funcionalidades a las que accede y brindarle al sistema mayor seguridad a nivel de usuario.

Observaciones:

- Los roles deben estar bien descritos, para evitar conflicto o solapamientos.
- Solamente podrán acceder a las funcionalidades del sistema asignadas al rol que posee.

Prototipo de interfaz:

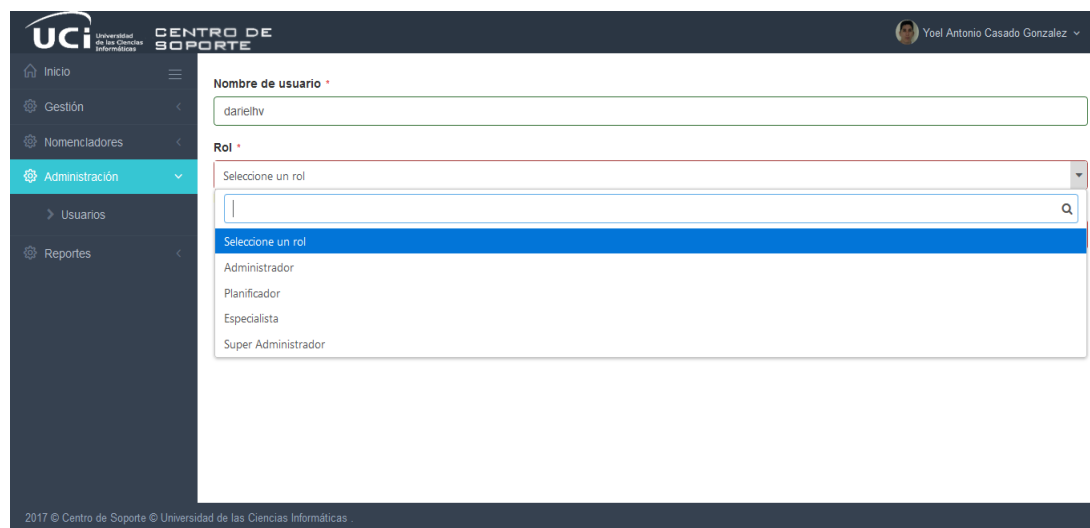


Tabla 4. Historia de Usuario "Asignar roles"

Historia de Usuario	
Número: HU_3	Nombre Historia de Usuario: Gestionar visitas.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoel Antonio Casado Gonzáles	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 2 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 2 semanas
Descripción: Permite adicionar, modificar y eliminar una visita en el sistema.	
Observaciones:	
<ul style="list-style-type: none"> - El usuario debe asignado los permisos que le permitan gestionar las visitas y estar autenticado en el sistema. - Para adicionar la visita se deben proporcionar los datos establecidos. - Para realizar las acciones de Modificar o Eliminar debe existir al menos una visita en el 	

Sistema de planificación de visitas para el Centro de Soporte

sistema.

- En una visita planificada no se completa el resumen.

Prototipo de interfaz:

UCI Universidad de las Ciencias Informáticas **CENTRO DE SOPORTE** Yoel Antonio Casado Gonzalez

Listado de visitas

+ Adicionar Actualizar Eliminar Filtro global

Responsable	Categoría	SubCategoría	Estado	Fecha ejecución
Todos los responsable	Todas las categorías	Todas las subcategorías	Todos los estados	Buscar
<input type="radio"/> Nelson Sanchez Alvarez	FGR	SIGE	Planificada	2017/06/16
<input type="radio"/> Ricardo Valle Priel	UCI	GESPRO	Ejecutada	2017/06/18
<input type="radio"/> Richard Abel Ravelo Tabet	FGR	SIGE	Ejecutada	2017/06/21

2017 © Centro de Soporte © Universidad de las Ciencias Informáticas

Tabla 5. Historia de Usuario "Gestionar visitas"

Historia de Usuario

Número: HU_11

Nombre Historia de Usuario: Realizar búsquedas de visitas.

Modificación de Historia de Usuario Número: ninguna

Usuario: Yoel Antonio Casado Gonzáles

Iteración Asignada: 2

Prioridad en Negocio: Alta

Puntos Estimados: 1.0 semanas

Riesgo en Desarrollo: Alto

Puntos Reales: 1.0 semanas

Descripción: Permite realizar búsquedas sobre visitas ya planificadas y efectuadas para conocer información detallada de las mismas.

Observaciones: Se debe tener acceso al módulo de búsquedas y debe haber visitas generadas.

Prototipo de interfaz:

Sistema de planificación de visitas para el Centro de Soporte

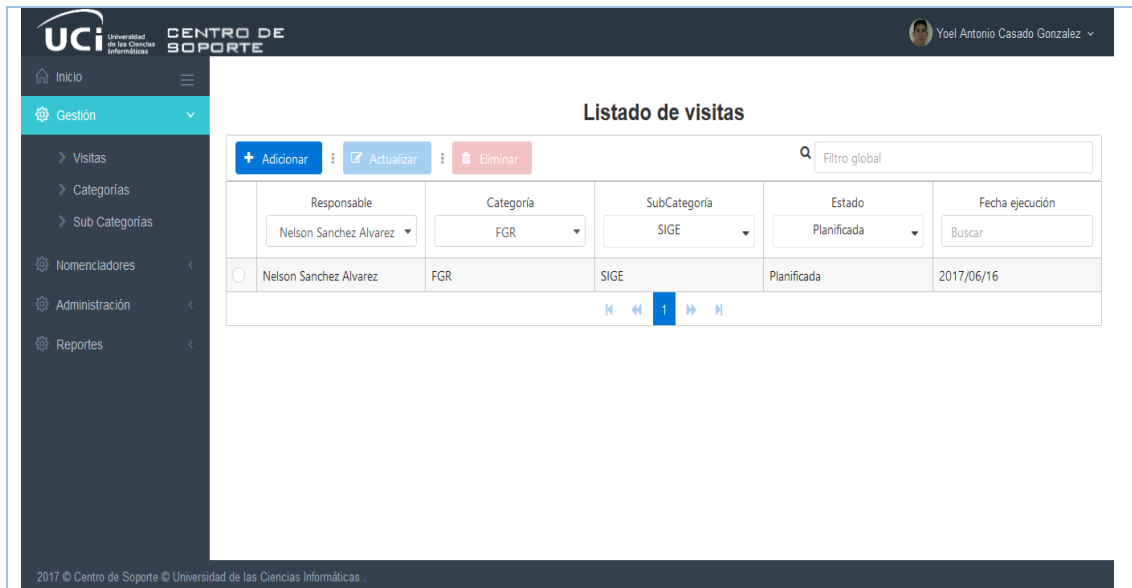


Tabla 6. Historia de Usuario “Realizar búsquedas de visitas”

Historia de Usuario	
Número: HU_12	Nombre Historia de Usuario: Mostrar datos de la visitas buscadas.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoel Antonio Casado Gonzáles	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1.0 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 1.0 semanas
Descripción: Permite visualizar la información de las visitas después de haber efectuado una búsqueda.	
Observaciones:	
<ul style="list-style-type: none"> - Se debe tener acceso al módulo de búsquedas y debe haber visitas generadas. - Se debe haber realizado una búsqueda. 	
Prototipo de interfaz:	

Sistema de planificación de visitas para el Centro de Soporte

The screenshot shows a web interface for the 'UC Centro de Soporte' system. On the left is a dark sidebar menu with options: Inicio, Gestión (with sub-items: Visitas, Categorías, Sub Categorías), Nomencladores, Administración, and Reportes. The main content area is a form titled 'Historia de Usuario' with the following fields:

- Responsable ***: Nelson Sanchez Alvarez
- Categoría ***: FGR
- Subcategoría ***: SIGE
- Fecha ejecución ***: 2017/06/16
- Dirección ***: Mayabeque
- Tipo de visita ***: Funcionamiento
- Tipo de transporte ***: Centro

Tabla 7. Historia de Usuario “Mostrar datos de las visitas buscadas”

Historia de Usuario	
Número: HU_15	Nombre Historia de Usuario: Realizar informes de visitas
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoel Antonio Casado Gonzáles	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 0.4 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 1 semanas
Descripción: Permite almacenar los datos de la visita seleccionada en forma de informe.	
Observaciones: Debe seleccionarse la visita que se desea y el estado debe estar en Ejecutada.	
Prototipo de interfaz:	

Sistema de planificación de visitas para el Centro de Soporte

Categoría	SubCategoría	Estado	Fecha ejecución
Todas las categorías	Todas las subcategorías	Todos los estados	Buscar
FGR	SIGE	Ejecutada	2017/06/21

Tabla 8. Historia de Usuario “Realizar informes de visitas”

Historia de Usuario	
Número: HU_19	Nombre Historia de Usuario: Graficar los resultados.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoel Antonio Casado Gonzáles	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 0.6 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 1 semanas
Descripción: Permite mostrar los datos de las visitas mediante gráficas.	
Observaciones: Se puede obtener por entidad, por fecha, ambas o todas dependiendo de la necesidad de usuario.	
Prototipo de interfaz:	

Sistema de planificación de visitas para el Centro de Soporte



Tabla 9. Historia de Usuario "Graficar los resultados"

2.6 Arquitectura del sistema y Diagrama de clases

Cuando se utiliza el *framework* Symfony, se utiliza necesariamente la arquitectura Modelo-Vista-Control (MVC), por ser sobre la cual se encuentra desarrollado el marco de trabajo, aunque no sea obligatoria durante su utilización. En esta arquitectura la interacción entre las capas lógicas del sistema puede realizarse en cualquier dirección, permitiendo un acceso más rápido y eficiente. Utilizando la propia estructura organizativa que proporciona Symfony se modelaron las clases y funcionalidades del sistema, mediante el diseño del Diagrama de Clases. De esta forma es posible comprender los elementos de implementación esenciales en el producto y la relación que se establece entre ellos. Además, se muestra la comunicación entre el controlador del *framework* y los archivos.twig¹⁷, donde desde el controlador es posible de renderizar una vista y darle los valores necesarios que contienen los modelos para que sean mostrados. Esto es posible a través de una función del *framework*, denominada *Action*, que introducida en el código fuente en forma de método.

¹⁷ Es la extensión de las vistas en Symfony.

Sistema de planificación de visitas para el Centro de Soporte

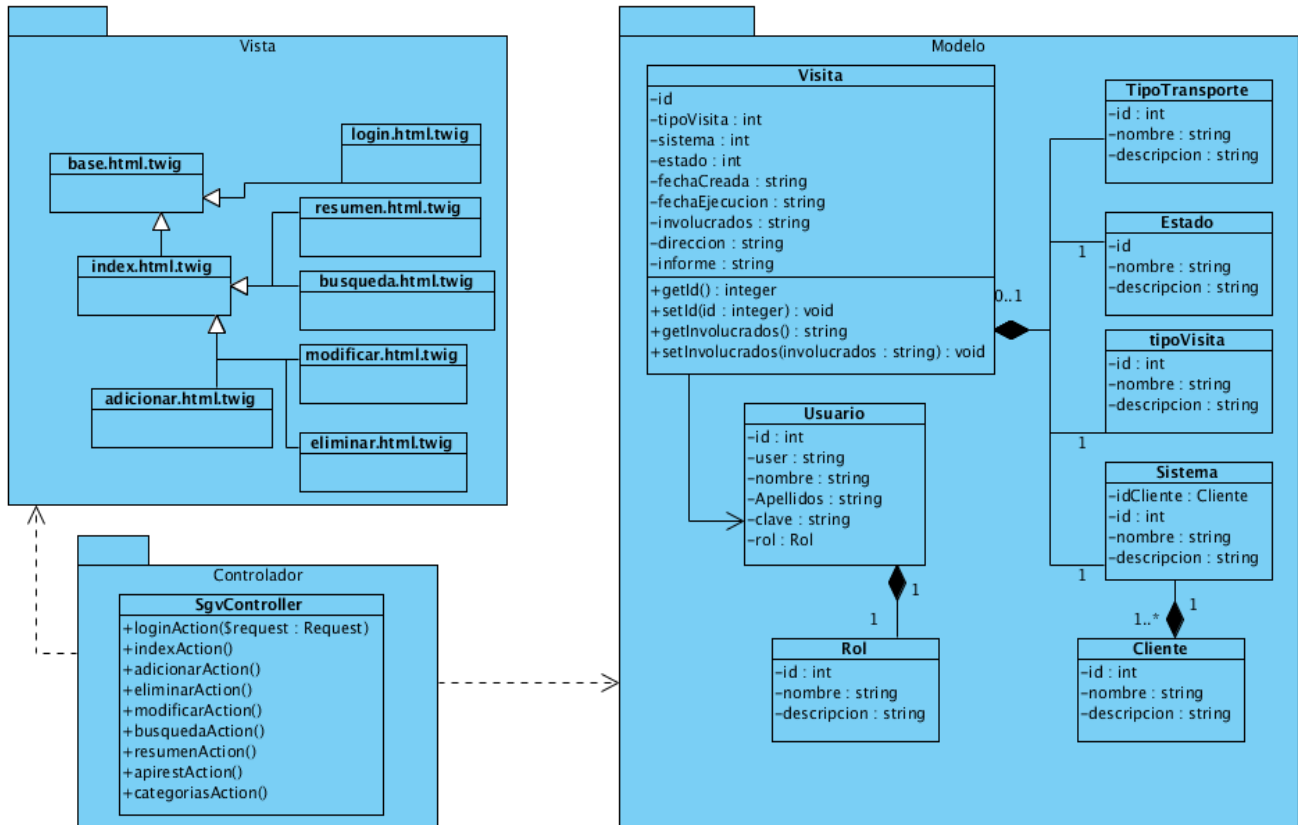


Ilustración 6. Diagrama de Clases y arquitectura.

2.7 Patrones del Diseño

Los patrones del diseño son un conjunto de buenas prácticas que se evidencia en la implementación de los productos software de manera reiterada, e incluso inherentemente. Su utilización permite obtener un código fuente organizado, una comunicación entre clases bien delimitada y la asignación de responsabilidades a los objetos que pueden cumplir el objetivo necesario. En desarrollo del Sistema de Planificación de Visitas se tuvieron presentes varios de ellos, los cuales se enuncian a continuación.

2.7.1 Patrones GRASP

Controlador: Asigna la responsabilidad de administrar un mensaje de eventos del sistema a una clase que represente una de las siguientes opciones (Lerman, Craig. 1999):

- El negocio o la organización global (un controlador de fachada).
- El "sistema" global (un controlador de fachada).
- Un ser animado del dominio que realice el trabajo (un controlador de papeles).
- Una clase artificial (Fabricación Pura) que represente el caso de uso (un controlador de casos de uso).

Este patrón se evidencia, principalmente, en la clase `SgvController` que controla la información desde y hacia las vistas, siendo un controlador único en la gestión del sistema.

Sistema de planificación de visitas para el Centro de Soporte

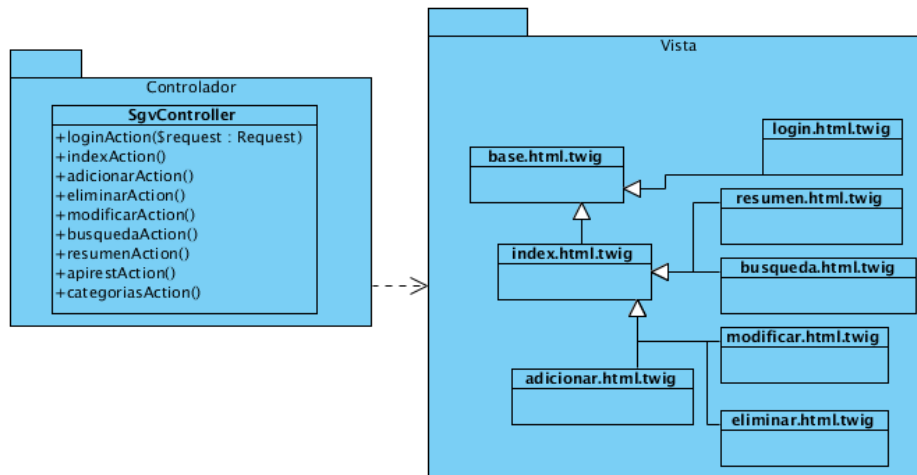


Ilustración 7. Patrón de Diseño Controlador.

Experto: El patrón experto en información es el principio básico de asignación de responsabilidades. Consiste en asignar la responsabilidad de creación de un objeto o la implementación de un método al contiene la información específica. La utilización de este patrón evita que deban solicitarse datos entre las clases de forma innecesaria (Fuente, y otros, 2014).

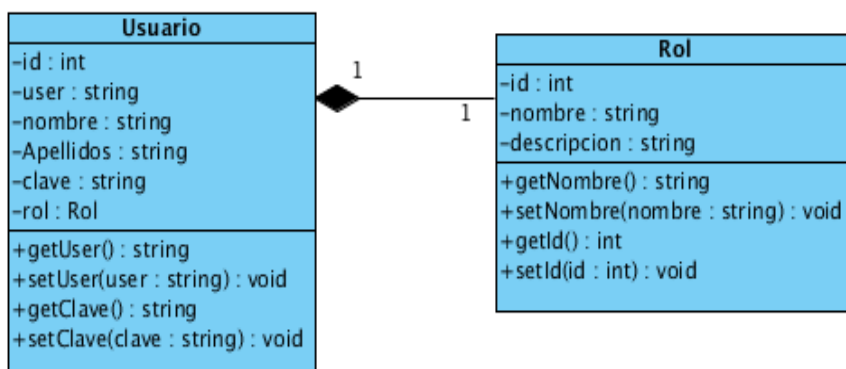


Ilustración 8. Patrón de Diseño Experto

Creador: La creación de objetos es una actividad muy frecuente en los sistemas orientados a objetos, por lo tanto, es conveniente asignar esta responsabilidad de manera que potencie un bajo acoplamiento, una mayor claridad y una alta reutilización. La nueva instancia debe ser creada por la clase que tenga la información mínima necesaria para hacerlo (Larman, Craig, 2007).

Este patrón se evidencia en la clase *SgvController*, ya es quien crea las instancias necesarias para hacer funcional el sistema de la forma más óptima posible.

Sistema de planificación de visitas para el Centro de Soporte

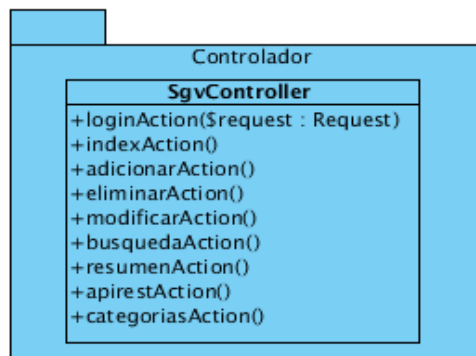


Ilustración 9. Patrón de Diseño Creador

Bajo Acoplamiento: El patrón bajo acoplamiento sigue la idea de tener clases con la menor relación posible entre ellas, pues en caso de producirse un cambio sustancial en alguna de ellas, no se afecten las demás. Este patrón le brinda al sistema mayor reutilización y evita la dependencia entre clases. En el SGV es posible evidenciar el Bajo Acoplamiento en la clase Visita, ya que una modificación en la clase Estado no afectaría su estabilidad y usabilidad.

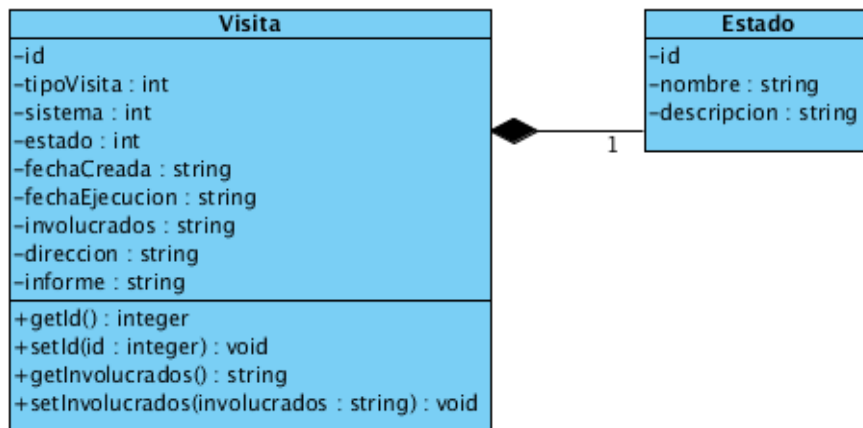


Ilustración 10. Patrón de Diseño Bajo Acoplamiento

Alta cohesión: El *framework* Symfony facilita la alta cohesión de la información, pues su concepción está familiarizada con la idea de que: “En la perspectiva del diseño orientado a objetos, la cohesión (o, más exactamente, la cohesión funcional) es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Una clase con baja cohesión hace muchas cosas no afines o un trabajo excesivo” (Lerman, 1999).

La clase controladora contiene las diferentes funcionalidades que se encuentran estrechamente relacionadas, posibilitando que el software sea flexible frente a grandes cambios.

Sistema de planificación de visitas para el Centro de Soporte

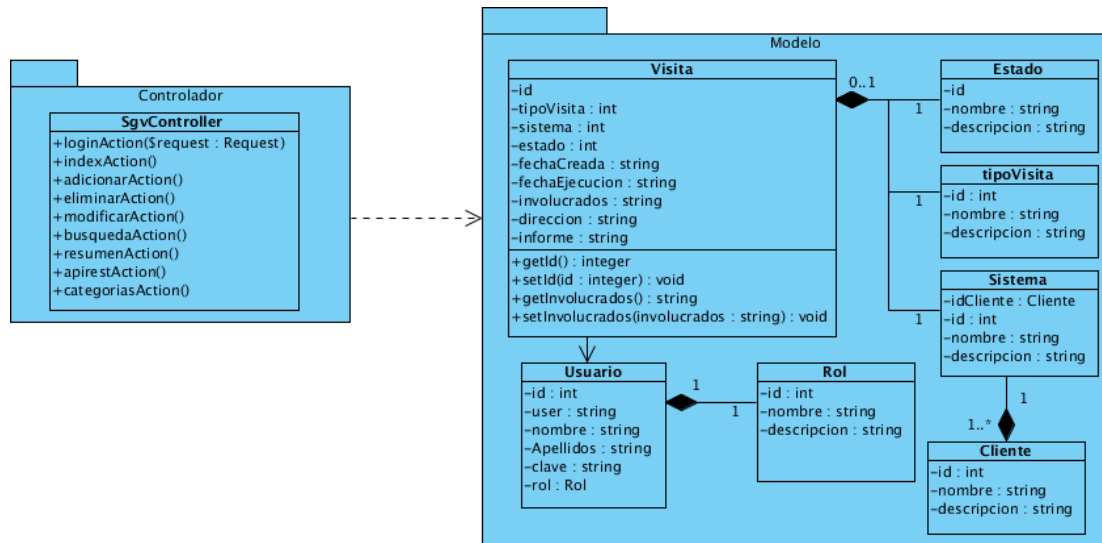


Ilustración 11. Patrón de Diseño Alta Cohesión

2.7.2 Patrones GOF

Los patrones GOF (Banda de los Cuatros) no son más que problemas recurrentes en el desarrollo de software que fueron clasificados y agrupados a partir de dos criterios, su propósito y alcance. Según este grupo los patrones “describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos” (Guerrero, y otros, 2013).

Singleton

Este patrón garantiza que solamente se cree una instancia de la clase y provee un punto de acceso global a él. Todos los objetos que utilizan una instancia de esa clase usan la misma instancia (Larman, Craig, 2007). Es evidenciado en la clase *SgvController* que utiliza las funciones de un único *Controller* por medio del método *\$this*.

Adapter

La evidencia de este patrón se hace posible gracias a la utilización del *framework* Symfony, teniendo en cuenta que este permite el uso de las instancias de los objetos tipo *controller* para acceder a la funcionalidad *getManager()*. De esta forma se puede acceder a las entidades de la base de dato, manejándolas como objetos a través del ORM¹⁸ *Doctrine*.

Template Method

Este patrón opera sobre todo el sistema permitiendo que las subclasses redefinan ciertos pasos sin cambiar su estructura. El uso de este patrón está vinculado en la estructura propia de la definición del *framework* Symfony y la forma en la que este permite gestionar las vistas. Los archivos. twig que son renderizados por el controlador para visualizar los datos del sistema, se crean sobre un archivo base que se usa como contenedor para mostrar sus contenidos y edita cada bloque dependiendo de las necesidades de cada página.

¹⁸ Mapeo de Objeto-Relacional.

Sistema de planificación de visitas para el Centro de Soporte

2.8 Diagrama Entidad - Relación

Teniendo en cuenta que la solución propuesta se basa en el desarrollo sobre tecnologías web es preciso contar con una Base de Datos que permita almacenar la información referente a las visitas gestionadas y demás funcionalidades del sistema. El diseño de la base de datos es el proceso mediante el cual se toman los elementos conceptuales del sistema y materializan en la base de datos. Estos elementos son representados mediante un diagrama de entidad-relación, definido como “una herramienta para el modelado de datos de un sistema de información. Estos modelos están basados en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre esos Objetos (Quintana Rondón, y otros, 2011).

A continuación, se muestra el modelo de datos del Sistema de planificación de Visitas que cuenta con siete entidades, siendo la entidad Visita la de mayor importancia. El usuario, una vez revisado en la base de datos podrá acceder, dependiendo del rol que posea, a sus vistas y ejecutar acciones sobre ellas.

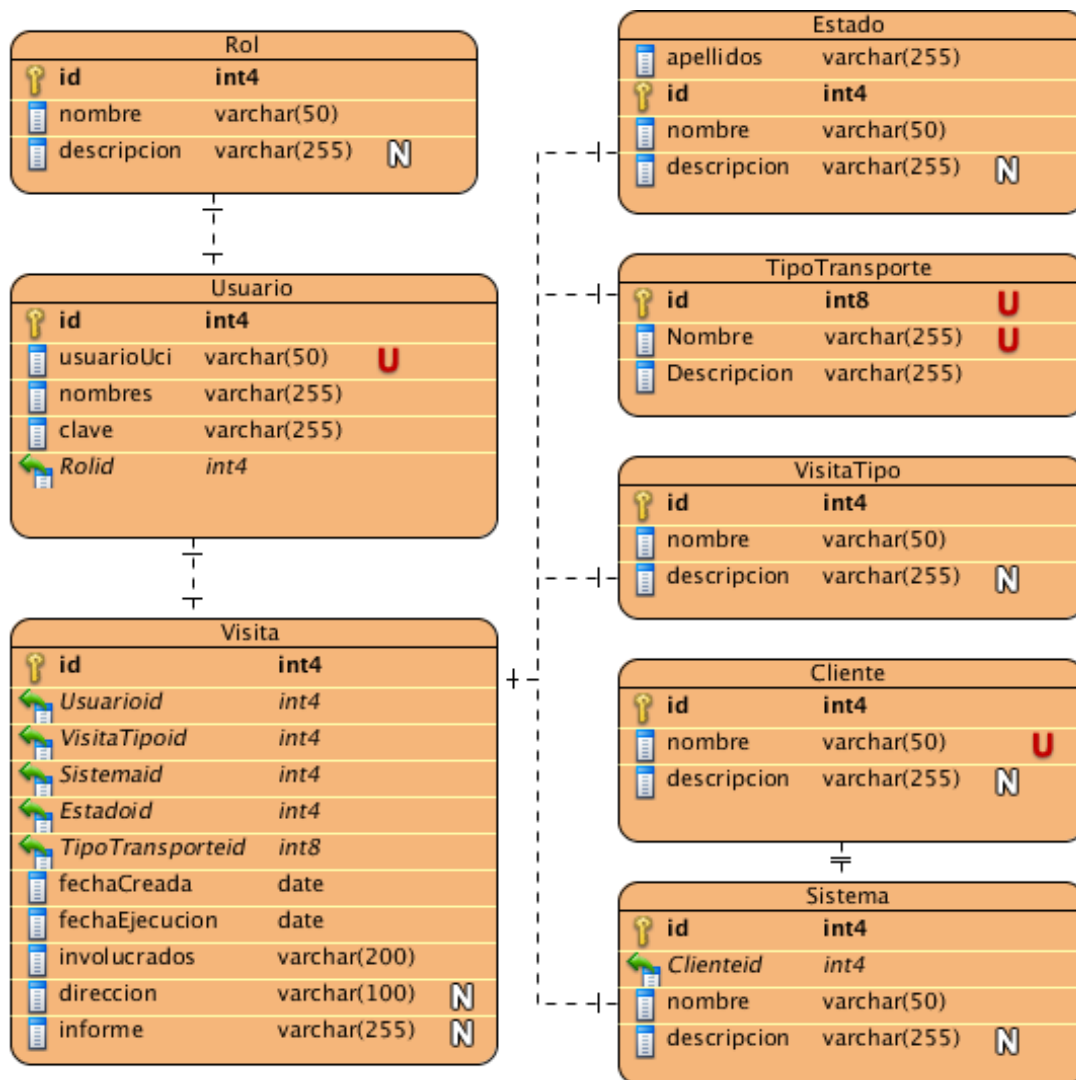


Ilustración 12. Diagrama Entidad- Relación

2.9 Conclusiones parciales

El modelado del Diagrama de Dominio, como parte del negocio para el desarrollo del Sistema de Planificación de Visitas, permitió conocer y comprender los principales conceptos vinculados al producto. Las relaciones y entidades modeladas parten del comportamiento de las visitas en el Centro de Soporte, acercando el desarrollo del software a la perspectiva del cliente. A partir de estos resultados se conformó una solución propuesta y concepción del sistema, para agrupar los objetivos y el funcionamiento genérico del sistema, con el fin de obtener elementos de consulta para los involucrados. Una vez definidas las directivas del desarrollo se realizó el levantamiento de los requisitos, funcionales y no funcionales, determinando el tiempo para concluir la implementación de la solución. La descripción de los requisitos fue apoyada por las Historias de Usuario, definiendo las funcionalidades del sistema y los elementos relevantes de cada una de ellas. Una vez concluido el análisis del Sistema de Planificación de Visita fue posible diseñar el diagrama de clases, definir la arquitectura de desarrollo y representar los principales patrones de diseño, evidenciados en el sistema. Además de la concepción del diagrama Entidad-Relación para representar las entidades de Base de Datos. Todo esto permitió obtener una idea clara del Sistema de Planificación de Visitas, los objetivos que persigue y los elementos claves a tener en cuenta en el desarrollo posterior de la solución.

CAPÍTULO III - Implementación y Pruebas

3.1 Introducción

En la realización de este capítulo se describe el proceso de implementación de la aplicación en términos de componentes; donde se detalla mediante el diagrama de despliegue como quedará distribuida la aplicación. Se muestran los resultados obtenidos de la aplicación de las métricas y técnicas empleadas para validar los requisitos y el diseño del sistema; además de los resultados alcanzados luego de la realización las pruebas. De ello, se generan los principales productos de trabajo definidos por la metodología, correspondientes a la fase de ejecución, en sus disciplinas (Implementación y Pruebas internas, de liberación y de aceptación).

3.2 Implementación

La implementación constituye una de las fases más importantes del desarrollo de software. En ella se toman como punto de partida los resultados obtenidos en el diseño, implementándose el sistema en términos de componentes como ficheros de código binario, código fuente, scripts y ejecutables. Su importancia reside en que se obtiene como consecuencia un sistema ejecutable, siendo esto uno de los principales objetivos en el desarrollo de software (Arizaca, 2009).

3.2.1 Diagrama de Componentes

Un diagrama de componentes representa la separación de un sistema de software en componentes físicos (por ejemplo archivos, módulos, paquetes, base de datos, código fuente, binario o ejecutable etc.) y muestra las dependencias y organización existente entre estos componentes (Arizaca, 2009).

En el diagrama de componentes de la Figura 13, se evidencian las interacciones entre el los componentes de Symfony2 y los del sistema de visitas. Symfony2 cuenta con los componentes Security (para la seguridad), Routing (para las rutas de acceso), Config (para las configuraciones), Mensajería (para la gestión de mensajes informativos y de errores), Excepciones (para el manejo de las excepciones generadas) y Ctr_Frontal es el controlador frontal que recibe cada petición realizada al sistema, entre otros. Estos componentes permiten un adecuado manejo de la seguridad, las configuraciones, las excepciones y errores del subsistema.

Sistema de planificación de visitas para el Centro de Soporte

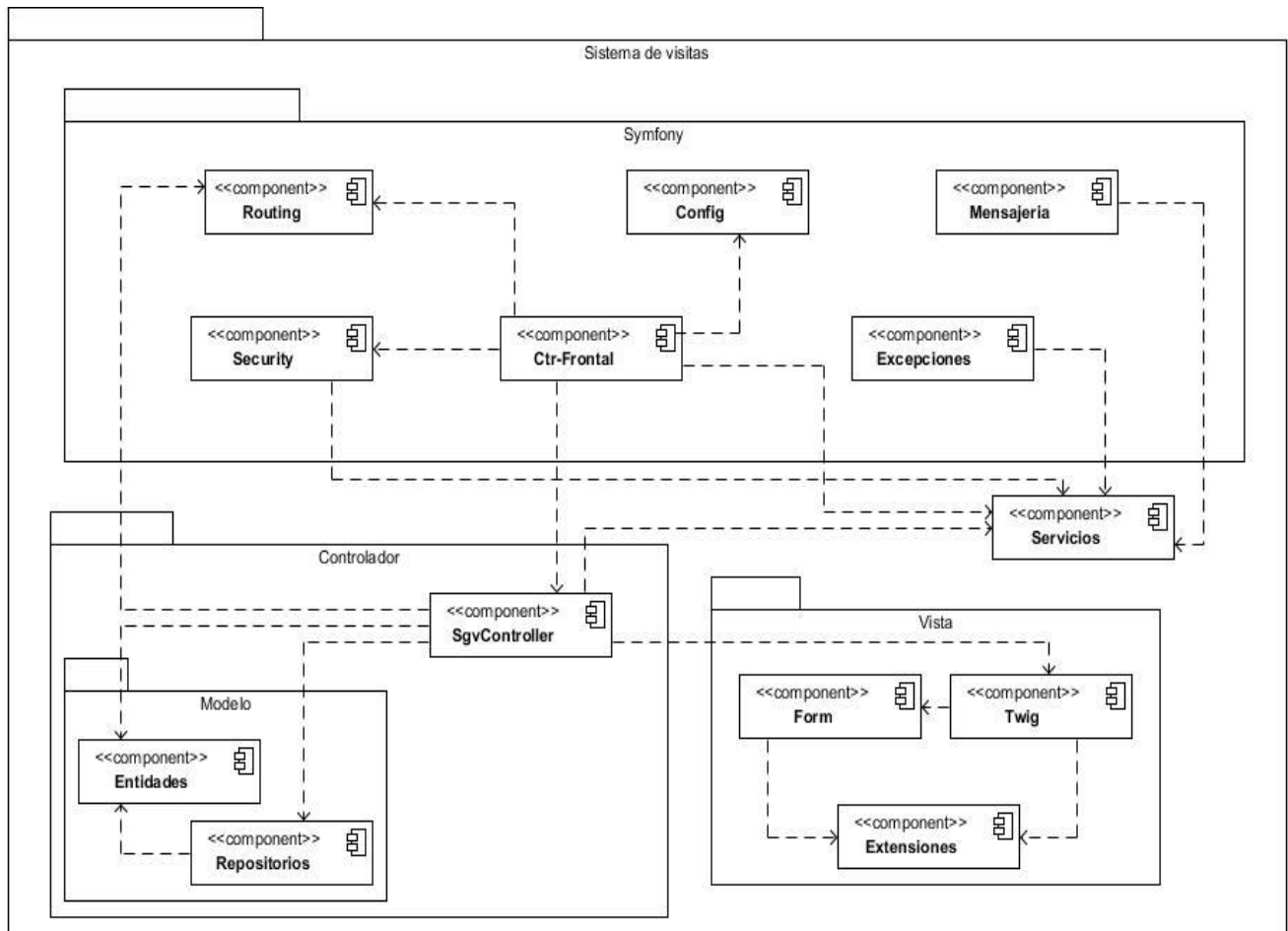


Ilustración 13. Diagrama de Componentes

El paquete Controlador contiene el componente SgvController, responsable de dar respuesta a las peticiones realizadas por los usuarios, posee una relación con el componente Routing para el manejo de las rutas definidas para el módulo y el componente Servicios que ofrece los servicios públicos.

El componente SgvController se relaciona, además, con el componente Entidades, que a su vez es usado por el componente Repositorio el cual contiene las clases repositorios encargadas de realizar las consultas a la base de datos. El paquete Vista incluye los componentes Form (encargado del trabajo con los formularios), Twig (para las interfaces de usuarios y plantillas del módulo) y Extensiones (posee los JavaScript y CSS).

De igual forma, estos componentes pueden ser agrupados y organizados atendiendo a la arquitectura del sistema para lograr un mejor entendimiento de sus funciones, distribución e interacciones entre sí.

3.2.2 Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema. El modelo de despliegue define la arquitectura física del sistema. Se usa

Sistema de planificación de visitas para el Centro de Soporte

para modelar de manera detallada los nodos físicos y las asociaciones de comunicación que existen entre ellos. Del mismo modo queda especificado qué hardware, sistemas operativos, software de interfaces y soporte. A continuación se muestra dicho diagrama, el mismo está compuesto por una computadora que utilizará el usuario para acceder al sistema y deberá tener instalado un navegador web. Este ordenador se comunica con el servidor de aplicación a través del protocolo HTTP¹⁹; el cual a su vez utilizará una conexión TCP-IP²⁰ para la comunicación con el servidor de base de datos. Además el servidor de aplicaciones consume los servicios web que brindan la plataforma de Soporte. Desde la PC cliente se puede además conectar una impresora mediante la interfaz USB.

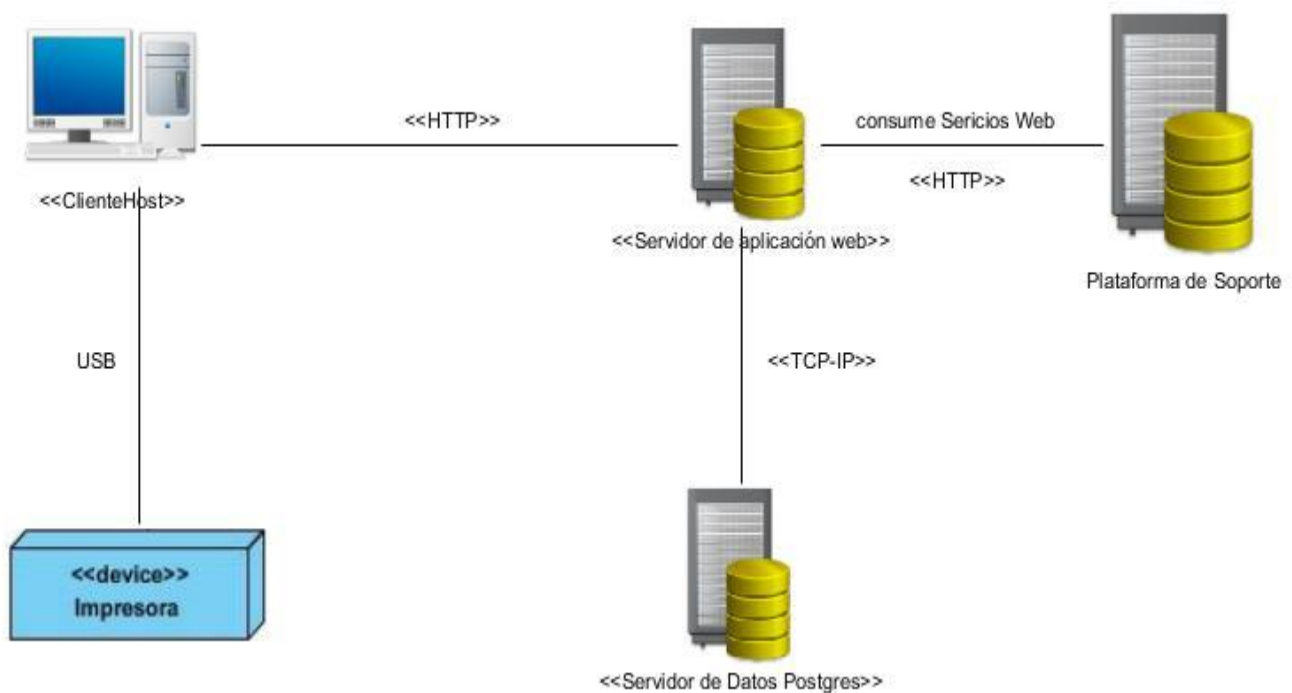


Ilustración 14. Diagrama de Despliegue

3.2.3 Prototipo de Interfaz de Usuario

Prototipos de interfaz de administrador asociados al componente Autenticación y Planificación.

¹⁹ HTTP: Protocolo de transferencia de hipertexto.

²⁰ TCP-IP: Protocolo de Control de Transmisión.

Sistema de planificación de visitas para el Centro de Soporte

Listado de visitas

+ Adicionar Actualizar Eliminar Filtro global

Responsable	Categoría	SubCategoría	Estado	Fecha ejecución
Todos los responsable	Todas las categorías	Todas las subcategorías	Todos los estados	Buscar
<input type="radio"/> Nelson Sanchez Alvarez	FGR	SIGE	Planificada	2017/06/16
<input type="radio"/> Ricardo Valle Priel	UCI	GESPRO	Ejecutada	2017/06/18
<input type="radio"/> Richard Abel Ravelo Tabet	FGR	SIGE	Ejecutada	2017/06/21

Ilustración 15. Prototipo Gestión de visitas.

Inicio

Visitas por estados

Ejecutada Planificada

Tipos de Visitas

Actualización Capacitación Facturización

Ilustración 16. Prototipo Interfaz Planificación

3.2.4 Estándares de Codificación

Los estándares de codificación son un elemento fundamental en la implementación de proyectos, permitiendo que el código generado sea fácil de leer y modificar independientemente de quien ha sido su autor. Son una guía para el equipo de desarrollo, permiten asegurar que el código presente alta calidad y no contenga errores.

Nomenclatura de las clases:

Los nombres de las clases comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing, esta especifica que los identificadores y nombres de variables, métodos y funciones que están compuestos por múltiples palabras juntas, deben iniciar cada palabra con letra mayúscula, lo que posibilita que con sólo leerlo se reconozca el propósito de la misma.

Ejemplo: TipoTransporte, en este caso el nombre de clase está compuesto por 2 palabras iniciadas cada una con letra mayúscula.

Nomenclatura según el tipo de clases:

Clases controladoras: Las clases controladoras después del nombre llevan la palabra: "Controller".

Ejemplo: SgvController

Nomenclatura de las funcionalidades y atributos:

El nombre a emplear para las funciones y los atributos se escribe con la inicial del identificador en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCasing que es similar a la antes mencionada: PascalCasing con la excepción de la primera letra.

Ejemplo de método: getInvolucrados. El nombre de método está compuesto por 2 palabras, la primera en minúsculas y la siguiente iniciando con letra mayúscula.

Las principales funcionalidades de las clases controladoras se les asigna el nombre y seguidamente la palabra: "Action"

Ejemplo: categoriasAction()

Nomenclatura de los comentarios:

Los comentarios deben ser lo suficientemente claros y concisos para que se entienda el propósito de lo que se está desarrollando. En caso de ser una función complicada se debe comentar su interior para lograr una mejor comprensión del código.

3.3 Validación de la investigación

Una vez concluida la investigación es necesario evaluar la innovación y aporte práctico de la misma, así como analizar si el objetivo general definido para guiarla fue cumplida

Sistema de planificación de visitas para el Centro de Soporte

satisfactoriamente, el cual plantea: “Desarrollar una herramienta que permita la planificación de visitas para el Centro de Soporte.” Entre los elementos que se pueden destacar, que permiten concluir que el objetivo general se cumple completamente, se encuentran los siguientes:

- Constituye un sistema que facilita el trabajo de los especialistas, ya que tiene toda la información sobre la planificación de las visitas de una forma más organizada.
- El sistema recopila todo lo referente a la planificación y asignación de visitas, de una manera estructurada, y al ritmo que se desee, dándole la posibilidad de realizar las modificaciones en caso que lo desee, lo que apoya al control de la información.
- La información generada durante el proceso de planificación cuenta con una seguridad adecuada, garantizada por los niveles de acceso que brinda el sistema, ejemplo de esto lo evidencia los roles que se le asigna a cada usuario con el objetivo que interactúe con el mismo, accediendo a cada parte según lo necesite.

3.4 Pruebas aplicadas al Sistema

El desarrollo del software ha de ir acompañado de alguna actividad que garantice la calidad del software, la prueba es un elemento crítico para ello. Es por ello que se deben incorporar acciones que evalúen la calidad del producto que se está desarrollando. Las pruebas aunque pertenecen al penúltimo flujo de trabajo ingenieril de AUP, no quiere decir que sea lo último que se realiza. Se puede ir realizando pruebas desde la fase de inicio del software hasta la fase de construcción, siendo esta última fase donde tiene mayor volumen el flujo de trabajo de prueba.

Las pruebas validan que las suposiciones hechas en el diseño y los requerimientos se estén cumpliendo satisfactoriamente, por lo que se encargan de verificar que el producto funcione como se diseñó y que los requerimientos se cumplan adecuadamente. Este flujo de trabajo brinda soporte para encontrar, documentar y solucionar defectos en el sistema.

Las pruebas se pueden realizar basándose en dos esquemas diferentes: demostrar a través de pruebas de caja blanca que las operaciones internas se ajustan a lo especificado y que los componentes internos marchan bien, o mediante las pruebas de caja negra, conociendo la función del programa e intentar demostrar que las funciones están correctas.

Pruebas de aceptación

El uso de cualquier producto de software tiene que estar justificado por las ventajas que ofrece. Sin embargo, antes de empezar a usarlo es muy difícil determinar si sus ventajas realmente justifican su uso. Para lograr esta determinación es utilizada la llamada “prueba de aceptación”. En esta prueba se evalúa el grado de calidad del software con relación a todos los aspectos relevantes para que el uso del producto se justifique.

Sistema de planificación de visitas para el Centro de Soporte

Para eliminar la influencia de conflictos de intereses y para que sea lo más objetiva posible, la prueba de aceptación nunca debería ser responsabilidad de los ingenieros de software que han desarrollado el producto. Para la preparación, la ejecución y la evaluación de la prueba de aceptación ni siquiera hacen falta conocimientos informáticos. El cliente es el mayor responsable de verificar cada una de las pruebas y de priorizar la corrección de las pruebas que fallan (IEEE, 2012).

Código caso de prueba : HU_3-1	Nombre Historia de Usuario: Adicionar Visita
Nombre de la persona que realiza la prueba: Yoel Antonio Casado González.	
Descripción de la prueba: El usuario accede al formulario de adición de una nueva visita en el cual llena los datos requeridos para dicho proceso, estos datos son incorrectos o existen campos obligatorios vacíos.	
Condiciones de ejecución: El servidor de base de datos de la aplicación en ejecución se encuentre en funcionamiento.	
Entradas: <ul style="list-style-type: none">-Responsable de la visita.-Involucrados.-Dirección.-Fecha de ejecución.	
Resultado esperado: El sistema alerta al usuario de los errores cometidos en el momento de completar el formulario de creación de una nueva conexión, se muestra un mensaje de error.	
Evaluación: Satisfactoria.	

Tabla 10. PA a la HU_3-1 Adicionar Visita

Código caso de prueba : HU_21-1	Nombre Historia de Usuario: Realizar informes de visitas
Nombre de la persona que realiza la prueba: Yoel Antonio Casado González.	
Descripción de la prueba: Consiste en evaluar el requisito realizar informe de visitas por parte del especialista al cual le fue asignado una visita.	
Condiciones de ejecución:	
Entradas: <ul style="list-style-type: none">-Observación.-Descripción.-Seleccionar la opción generar informe.	
Resultado esperado: El sistema alerta al usuario de los errores cometidos en el momento	

Sistema de planificación de visitas para el Centro de Soporte

de completar el formulario de los datos a llenar para generar el informe.

Evaluación: Satisfactoria.

Tabla 11. PA a la HU_15-1 Realizar informe de visitas

Código caso de prueba : HU_14-1

Nombre Historia de Usuario: Alertar sobre visita ejecutada sin resumen.

Nombre de la persona que realiza la prueba: Yoel Antonio Casado González.

Descripción de la prueba: Consiste en evaluar el requisito Alertar sobre visita efectuada sin resumen.

Condiciones de ejecución:

Entradas:

-Responsable.

-Resumen.

Resultado esperado: El sistema muestra una alerta al usuario que la visita fue ejecutada y no tiene el resumen.

Evaluación: Satisfactoria.

Tabla 12. PA a la HU_14-1 Alertar sobre visita ejecutada sin resumen

Código caso de prueba : HU_5-1

Nombre Historia de Usuario: Notificar visita.

Nombre de la persona que realiza la prueba: Yoel Antonio Casado González.

Descripción de la prueba: Consiste en evaluar el requisito Notificar visita a los especialistas correspondientes.

Condiciones de ejecución:

Entradas:

-Responsable.

-Involucrados.

Resultado esperado: El sistema muestra que la visita ha sido planificada correctamente

Evaluación: Satisfactoria.

Tabla 13. PA a la HU_5-1 Notificar visita

Código caso de prueba : HU_9-1

Nombre Historia de Usuario: Modificar una visita

Nombre de la persona que realiza la prueba: Yoel Antonio Casado González.

Descripción de la prueba: Consiste en evaluar el requisito Modificar una visita y notificar al especialista responsable.

Condiciones de ejecución:

Entradas:

Sistema de planificación de visitas para el Centro de Soporte

-Responsable.

-Involucrados.

-Fecha de ejecución.

Resultado esperado: El sistema muestra que la visita ha sido modificada correctamente.

Evaluación: Satisfactoria.

Tabla 14. PA a la HU_9-1 Modificar una visita

Resultados de las pruebas de aceptación

Se realizaron 11 Pruebas de aceptación, de acuerdo a la cantidad de Historias de usuarios definidas. Para ello el cliente ejecutó los pasos de acuerdo a lo establecido en la fase de negocio para cada una de las funcionalidades, evaluando los resultados.

En la primera iteración resultaron satisfactorias 6 de las pruebas realizadas representando el 54,55 % y 5 no satisfactorias representando el 45,45 %. Las no conformidades encontradas fueron:

- ✓ La funcionalidad Adicionar Visita: No permite cambiar el estado de la visita.
- ✓ La funcionalidad Pagar los resultados de las búsquedas: No permite mostrar las visitas cuando el resultado de la búsqueda excede el total permitido para una página.
- ✓ La funcionalidad Notificar visita: No permitía enviar la notificación por correo a los involucrados de las visitas, solo al responsable.
- ✓ La funcionalidad Asignar responsable: No permite seleccionar a cualquiera de los especialistas como responsable de la visita por dependencia de su rol.
- ✓ La funcionalidad Graficar resultados: No permite graficar los resultados de las vistas por tipo, grafica todos los resultados en general.

En la segunda iteración resultaron 9 pruebas evaluadas de bien lo que representó el 81,82 % y de mal 2 pruebas representando el 18,18% del total. Las no conformidades encontradas fueron:

- ✓ La funcionalidad Modificar visita: Solamente enviaba correo de notificación al responsable y no a los involucrados.
- ✓ La funcionalidad Adicionar Sub-Categoría: Se adicionaba la sub-categoría sin depender directamente de su categoría.

Las deficiencias encontradas se continuaron corrigiéndose, resultando en una tercera iteración el 100 % de las pruebas satisfactorias.

Pruebas de caja negra:

Pruebas que se llevan a cabo sobre la interfaz del software. El objetivo es demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene (no se ve el código). Se centran principalmente en los requisitos funcionales del software.

Sistema de planificación de visitas para el Centro de Soporte

Permiten encontrar:

- Funciones incorrectas.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Las pruebas de caja negra que se le realizó al sistema propuesto, fueron las siguientes mostrando la parte de la interfaz “Gestionar visitas” al cual se le aplicó dichas pruebas.

The screenshot displays the 'Gestionar visitas' form in the UCI Centro de Soporte system. The form includes the following fields and values:

- Responsable:** Mauricio Moreno Molina
- Categoría:** UCI
- Subcategoría:** GESPRO
- Fecha ejecución:** 2017/06/24
- Dirección:** 225 # 218 y 222. Apto 3. A red border and error message are visible below this field: "Solo se permiten los siguientes caracteres 0-9A-Za-z-&'();<>? ".
- Tipo de visita:** Capacitación
- Tipo de transporte:** Piquera
- Estado:** Planificada
- Involucrados:** Richard
- Resumen:** Empty text area.

The interface also features a sidebar with navigation options: Inicio, Gestión, Visitas, Categorías, Sub Categorías, Nomencladores, and Reportes. The user 'Yoel Antonio Casado Gonzalez' is logged in. At the bottom right, there are 'Aceptar' and 'Cancelar' buttons.

Ilustración 17. Interfaz de Usuario HU Gestionar visita

A continuación se muestra un caso de prueba realizado a la Historia de Usuario Gestionar visita. Se entiende por Historia de Usuario, aquellas que son utilizadas en las metodologías de desarrollo ágiles para la especificación de requisitos (acompañadas de las discusiones con los usuarios y las pruebas de validación). Cada historia de usuario debe ser limitada, ésta debería poderse escribir sobre una nota adhesiva pequeña.

Sistema de planificación de visitas para el Centro de Soporte

Historia de usuario	Gestionar visita
Caso de Prueba	2
Entrada	Responsable: Claudia de León Martínez Categoría: UCI Subcategoría: GESPRO Fecha ejecución: 10/6/2017 Dirección: 225/ 218 y 222. Apto 3 Tipo de visita: Capacitación Tipo de transporte: Piquera Estado: Planificada Involucrados: Yoel Antonio Casado Resumen: -
Salida	Todos los datos de los atributos son válidos y la entidad no se encuentra en la base de datos, el sistema inserta la nueva entidad en la base de datos y lo notifica al usuario.

Tabla 15. Caso de prueba # 2 utilizando método de caja negra

Historia de usuario	Gestionar visita
Caso de Prueba	3
Entrada	Responsable: Mauricio Moreno Molina Categoría: UCI Subcategoría: GESPRO Fecha ejecución: 24/6/2017 Dirección: 225/ 218 y 222. Apto 3 Tipo de visita: Capacitación Tipo de transporte: Estado: Involucrados:
Salida	Como existen atributos cuyos campos están vacíos, no se permite la adición de la visita planificada. Se deben completar todos los campos y luego adicionar.

Tabla 16. Caso de prueba # 3 utilizando método de caja negra

Sistema de planificación de visitas para el Centro de Soporte

Historia de usuario	Gestionar visita
Caso de Prueba	4
Entrada	Responsable: Mauricio Moreno Molina Categoría: UCI Subcategoría: GESPRO Fecha ejecución: 24/6/2017 Dirección: 225 # 218 y 222. Apto 3 Tipo de visita: Capacitación Tipo de transporte: Piquera Estado: Planificada Involucrados: Richard Resumen:
Salida	Existen datos de la entidad que poseen caracteres no válidos. El sistema muestra un mensaje de error al usuario donde expone que existen caracteres no válidos. Si estos datos no son corregidos, el sistema no permite la adición de los mismos.

Tabla 17. Caso de prueba # 4 utilizando método de caja negra

Pruebas de caja blanca:

Este tipo de pruebas de software se realiza sobre las funciones internas de un módulo. Las pruebas de caja blanca están dirigidas a las funciones internas. Se basan en un minucioso examen del código y es necesario conocer la lógica del programa y comprobando el estado del software en varios puntos, para verificar que los resultados de dicho estado coincidan con los esperados (Laurie Williams, 2006).

Las pruebas del tipo caja blanca que se realizaron en el sistema fueron los test unitarios o pruebas unitarias como se conoce mayormente. Estos son imprescindibles para controlar la calidad del código de la aplicación. Los test unitarios prueban pequeñas partes del código, como por ejemplo una función o un método.

Symfony 1 disponía de una herramienta propia para crear y ejecutar los test. Symfony2 ha optado por utilizar la librería PHPUnit, que prácticamente se ha convertido en un estándar en el mundo PHP. De esta forma, los test unitarios y funcionales de Symfony2 combinan la potencia de PHPUnit con las utilidades y facilidades proporcionadas por Symfony2. Los test unitarios en Symfony2 prueban que un pequeño fragmento de código de la aplicación funciona tal y como debería hacerlo.

Sistema de planificación de visitas para el Centro de Soporte

Idealmente, los fragmentos de código son la parte más pequeña posible que se pueda probar. En la práctica suelen probarse clases enteras, a menos que sean muy complejas y haya que probar sus métodos por separado. . Fueron realizadas 30 pruebas de las cuales 23 fueron satisfactorias y 7 no satisfactorias, los resultados de algunas de las pruebas realizadas se muestran a continuación.

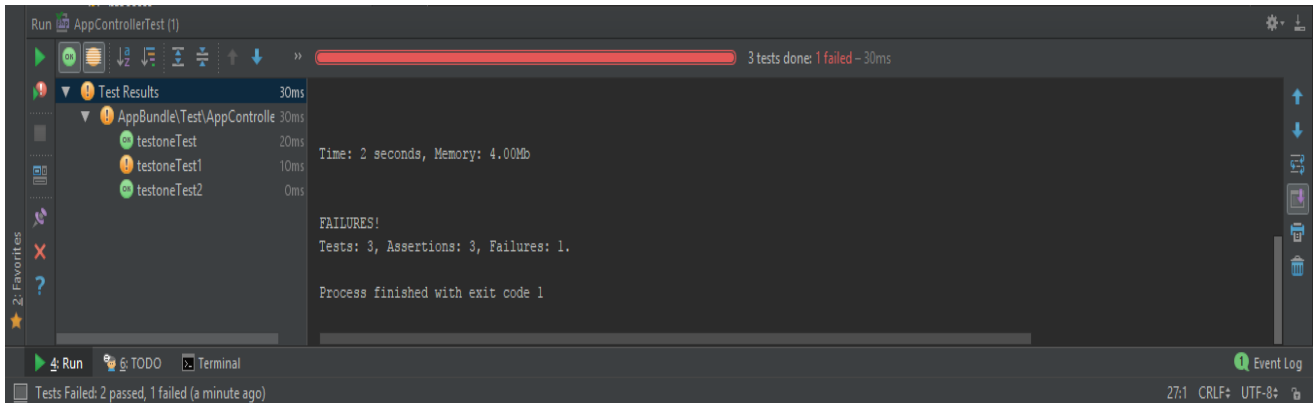


Ilustración 18. Pruebas unitarias AppControllerTest



Ilustración 19. Pruebas unitarias TipoVisitaControllerTest

3.5 Conclusiones parciales

En este capítulo la confección del diagrama de componentes ofreció una vista arquitectónica de alto nivel para ayudar al equipo de desarrollo en la implementación. Mientras que el diagrama de despliegue permitió evidenciar la relación entre el software y el hardware.

Al mismo tiempo la definición de los estándares de codificación a tener en cuenta para la implementación del sistema de visitas, permitió garantizar que el código posea alta calidad, menos errores y que pueda ser mantenido fácilmente, así como reutilizado por desarrolladores de otros proyectos que lo necesiten.

Por otra parte la realización de las pruebas de software al sistema facilitó encontrar los errores existentes para poder solucionarlos, posibilitando determinar y asegurar la calidad y robustez del Sistema de visitas para el Centro de Soporte.

Conclusiones

Con la culminación de la investigación se arriba a las siguientes conclusiones:

El análisis de los sistemas homólogos evidenció la necesidad de desarrollar un nuevo sistema y permitió incluir funcionalidades comunes en este tipo de herramientas.

Se estudiaron las técnicas, metodologías, lenguajes de programación y herramientas necesarias para el desarrollo del sistema. Lo cual permitió definir las herramientas que se utilizarían para el desarrollo de la aplicación.

El uso de la metodología AUP para guiar el proceso de desarrollo de la herramienta y los artefactos generados en cada una de sus fases propició una mejor comunicación con el cliente, logrando obtener un mejor funcionamiento del sistema.

Se desarrolló una aplicación web que contribuyó a tener una mejor organización en el Centro de Soporte, facilitando una mejor comunicación con los clientes e incrementando el control al resolver los incidentes de los sistemas.

Se comprobó la funcionalidad y confiabilidad de la aplicación mediante la realización de pruebas tales como pruebas unitarias automatizadas y de aceptación.

Referencias bibliográficas

- Quintana Rondón, Yoandri, Camejo Domínguez, Lianet y Díaz Berenguer, Abel . 2011.** *Diseño de la Base de Datos para Sistemas de Digitalización y Gestión de Medias*. Argentina : Revista de Informática Educativa y Medios Audiovisuales, 2011. Vol. 8(15) , págs. 17-25, ISSN: 1667-8338.
- 3C TIC. Gutiérrez, Alfonso y Oltra, Raúl. 2015.** No. 12, España : Desarrollo del Proceso de la Capacidad ITIL en una compañía outsourcing de TI, 2015, Vol. Vol. 4. ISSN: 2254-6529.
- Arizaca Ramírez, Eliza, 2009.** *Análisis y diseño de sistemas II*. 9 de junio de 2009.
- Aguilar Morales, Jorge y Vargas Mendoza, Jaime Ernesto. 2010.** *Servicio al Clientes*. México : Asociación Oaxaqueña de Psicología , 2010.
- Alvarez, Ruben, Alvarez, Miguel Angel y Cuenca, Carlos Luis. 2013.** *Programación en PHP*. s.l. : Sitio Oficial Desarrollo Wwb, 2013.
- Basalo, Alberto, y otros. 2016.** *Manual de AngularJS*. s.l. : Desarrollo Web, 2016.
- Benavides, Luis Hernan. 2011.** *Gestión, Liderazgo y Valores en la administración*. s.l. : Centro Universitario de Guayaquil, 2011.
- Bernal, Dra. María. 2012.** *La Planificación: Conceptos básicos, principios, componentes, características y desarrollo del proceso*. s.l. : Universidad Santa María, 2012.
- Botero Chica, Carlos Alberto. 2009.** *Cinco tendencias de la gestión educativa*. s.l. : Revista Iberoamericano de Educación, 2009. ISSN: 1681-5653.
- Business School. 2016.** OBS. OBS . [En línea] Universidad de Barcelona, 2016. <http://www.obs-edu.com/int/blog-project-management/diagramas-de-gantt/que-es-un-diagrama-de-gantt-y-para-que-sirve>.
- CalidadUPV. 2011.** *Manual de Procesos y Procesos*. España : Universidad Politécnica de Valencia, 2011.
- Visual Paradigm for UML.** Visual Paradigm for UML 8.0 Released. [En línea] 16 de Agosto de 2010. [Citado el: 1 de diciembre de 2012.] <http://www.visual-paradigm.com/>.
- CEIGE. 2015.** *Manual de Usuario del Sistema SIPAC*. La Habana : UCI, 2015.
- Eguíluz Pérez, Javier. 2009.** *Introducción a JavaScript*. 2009.
- ESIC. 2016.** *Manual de Técnicas Comerciales*. Asturias : Programa de Desarrollo Comercial para tecnólogos, 2016.
- Espinosa, Eduar Luis. 2013.** *Generador de Datos para Bases de Datos NoSQL*. La Habana : UCI, 2013.
- Espinosa, María Alejandra. 2003.** *Diagrama de Gantt*. 2003.

Sistema de planificación de visitas para el Centro de Soporte

Estrada Vera, William. 2007. *Servicio y Atención al Cliente*. Perú : Proyecto de Mejoramiento de los Servicios de Justicia, 2007.

Fuente, Mario Luis y Hernández, Pedro Enrique. 2014. *Sistema integrado para la monitorización de MongoDB y CouchDB*. La Habana : Universidad de las Ciencias Informáticas, 2014.

García, Sheyla y González, Félix. 2013. *Extensión del NetBeans para el diseño de interfaces gráficas de usuario con ExtJS*. La Habana : UCI, 2013.

Gestión Industrial. **Mula, Josefa, Poler, Raúl y García, José. 2005.** No. 1, Venecia : Evaluación de Sistemas para la Planificación y Control de la Producción, 2005, Vol. Vol.17. ISSN: 0718-0764.

Thompson, Ivan. Definiciones de Servicios. [En línea] Agosto de 2006. [Citado el: 10 de Enero de 2013.] <http://www.promonegocios.net/mercadotecnia-servicios/definicion-servicios.html>.

Guerrero, Carlos A., Suárez, Johanna M. y Gutierrez, Luz E. 2013. Colombia : Revista La Serena, 2013. Vol. 24, ISSN: 0718-0764.

Gantt, Henry. Laurence. «Work, Wages and Profit.»The Engineering Magazine, Nueva York

Musciano, Chuck.HTML, la Guía Completa. México: s.n., 1999.

Huerta, Amaia y Zuzuarregui, Anne. 2015. *Análisis de las características de los ERPs para pymes: Una guía preliminar de cara a la elección de las soluciones más eficientes*. España : Universidad del País Vasco, 2015.

ISO 9001. 2015. Nueva ISO 9001:2015. *Nueva ISO 9001:2015*. [En línea] Norma ISO, 2015. <http://www.nueva-iso-9001-2015.com/2014/11/iso-9001-entendiendo-enfoque-basado-procesos/>.

Lago Clara, Neybis, Quesada, Yunior y Nuez, Carlos Alberto. 2016. *Módulo de Gestión de Incidencias en el Centro de Soporte de la UCI*. Cuba : Informática Habana, 2016. ISSN: 978-959-289-122-7.

Lerman, Craig. 1999. *UML y Patrones*. México : s.n., 1999. 970-17-0261 -1.

Larman, Craig. 2007. *UML y Patrones*.s.l: Printice Hall, 2007.

Lucio Nieto, Teresa. 2010. Customer Care Associates. *Customer Care Associates*. [En línea] Mexico, 2010. [Citado el: 02 de 03 de 2016.] <http://www.customercareassoc.com/boletines/noticias/tenb20601.htm>.

—. **2012.** *Hacia una Oficina de Gestión de Servicios en el ámbito de ITIL*. Madrid : Instituto Tecnológico y de Estudios Superiores de Monterrey, 2012. ISSN 1698-2029.

Lucio Nieto, Teresa, Colomo Palacios, Ricardo y Mora Soto, Arturo. 2012. *Hacia una Oficina de Gestión de Servicios en el ámbito de ITIL*. Mexico : Instituto Tecnológico y de Estudios Superiores de Monterrey, 2012. ISSN 1698-2029.

Sistema de planificación de visitas para el Centro de Soporte

Ortiz Ricardo, Iraini y Pantoja Guerrero, Yunior. 2014. *Red social de investigadores para la gestión del conocimiento en el contexto de la Universidad de las Ciencias Informáticas.* La Habana : UCI, 2014.

PostgreSQL. 2013. PostgreSQL.org. *PostgreSQL.org.* [En línea] 2013.

http://www.postgresql.org.es/sobre_postgresql.

Pozo Guevara, Macdemis Liliana. 2014. *Interfaces para el módulo de Administración del Sistema de Gestión de Servicios de Soporte del Centro de Soporte UCI.* La Habana : UCI, 2014.

Programa de Mejora. 2014. *Metodología de Desarrollo para la actividad productiva de la UCI.* La Habana : UCI, 2014.

Rendón Artola, Ariadna y Castellano Perez, Manuel A. 2016. *Modelación y construcción del Componente Gestor de Actividades y Calendario para el subsistema Planificación por objetivos del SIGE.* La Habana : UCI, 2016.

Reyes Jiménez, Wendy y Lago Vasallo, Orelbys. 2014. *Aplicación para la creación de animaciones web con HTML 5.* La Habana : UCI, 2014.

Reyes, Alain y Dueñas, Yoannys Gustavo. 2014. *Diseño e implementación de una solución para los procesos de Prueba Pericial y de Presunción del Sistema de Informatización para la Gestión de Tribunales Populares Cubanos.* La Habana : UCI, 2014.

Roig, José Manuel y Oltra, Raúl. 2015. *Propuesta de Modelo de evaluación de herramientas para la gestión del proceso de Gestión de Problemas de ITIL.* s.l. : 3C TIC, 2015. ISSN: 2254-6529.

Romero, Jhon Smith. 2014. *Desarrollo de software para barzo robot.* s.l. : Universidad Católica de Pereira, 2014.

Riehle, Dirk, 2000. *Framework Design: A Role Modeling Approach.*

Mora, Ing. Jhon Freddy Montes. 2008. *Puebas del software "Caja Blanca y Caja Negra".* Colombia : Universidad Nacional Abierta y a Distancia UNAD, 2008

Soporte UCI. 2016. Centro de Soporte. [En línea] Universidad de las Ciencias Informáticas, febrero de 2016. <https://soporte.uci.cu/misión>.

Sammet, Jean E, 1969. *Programming Languages: History and Fundamentals,* Prentice-Hall, 1969

Turner, Raymond, 2005. "The Foundations of Specification", Vol 15, No 5 (Octubre 2005), pp. 623-663.

UIS Ingenierías. Sánchez Torres, Jenny Maricela, González Zabala, Mayda Patricia y Sánchez Muñoz, María Paloma. 2012. No.1, Madrid : La Sociedad de la Información: Génesis, Iniciativas, Concepto y su relación con las TIC, 2012, Vol. Vol. 11.

Sistema de planificación de visitas para el Centro de Soporte

Vázquez Meriño, Carlos. 2008. *Programación en PHP 5. Nivel Básico.* 2008.

Laurie Williams, 2006. *White-Box Testing.* 2006.

Sistema de planificación de visitas para el Centro de Soporte

Anexos

Anexo #1: Historias de Usuarios del Sistema de Planificación de Visitas.

Historia de Usuario	
Número: HU_4	Nombre Historia de Usuario: Asignar responsable.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoel Antonio Casado Gonzáles	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 0.4 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 0.4 semanas
Descripción: Permite asignar una visita a un responsable.	
Observaciones: - Debe tener permisos de administración. - El responsable debe coincidir con un especialista o trabajador del Centro de Soporte.	
Prototipo de interfaz:	

Historia de Usuario	
Número: HU_5	Nombre Historia de Usuario: Notificar visita.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoel Antonio Casado Gonzáles	Iteración Asignada: 3
Prioridad en Negocio: Muy alta	Puntos Estimados: 0.4 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 0.4 semanas
Descripción: Permite avisar a los involucrados en la visita de los cambios realizados.	
Observaciones: Los involucrados deben poseer un correo válido.	
Prototipo de interfaz:	

Historia de Usuario	
Número: HU_6	Nombre Historia de Usuario: Brindar Apis Res de visitas.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoel Antonio Casado Gonzáles	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 0,4 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 0,4 semanas
Descripción: Con estos valores estaremos brindando la posibilidad de vincular al sitio del centro con el sistema, para que los clientes puedan ver este cronograma de visitas desde el portal del centro.	

Sistema de planificación de visitas para el Centro de Soporte

Observaciones: Para poder crear este servicio es preciso tener listo el modelado de los datos en la base de datos.

Prototipo de interfaz:

Historia de Usuario

Número: HU_7 **Nombre Historia de Usuario:** Crear un resumen de vista.

Modificación de Historia de Usuario Número: ninguna

Usuario: Yoel Antonio Casado Gonzáles **Iteración Asignada:** 1

Prioridad en Negocio: Alta **Puntos Estimados:** 0.4 semanas

Riesgo en Desarrollo: Alto **Puntos Reales:** 0.4 semanas

Descripción: Una vez efectuada la visita esta no cambiara de estado a ejecutada si no se elabora un resumen por parte del responsable de la visita sobre las actividades de la misma, dejando constancia de la actividad realizada y cambiando el estado de la misma.

Observaciones: Debe haber sido creada con anterioridad la visita.

Prototipo de interfaz:

Historia de Usuario

Número: HU_8 **Nombre Historia de Usuario:** Cambiar el estado de la visita

Modificación de Historia de Usuario Número: ninguna

Usuario: Yoel Antonio Casado Gonzáles **Iteración Asignada:** 1

Prioridad en Negocio: Alta **Puntos Estimados:** 0.8 semanas

Riesgo en Desarrollo: Alto **Puntos Reales:** 0.8 semanas

Descripción: Cuando el operador del sistema agregó una visita esta debe quedar en un estado de planificada, para luego de efectuada la visita cambiar su estado a ejecutada, permitiendo tener un control de las vistas realizadas y las visitas pendientes.

Observaciones:

Prototipo de interfaz:

Historia de Usuario

Número: HU_9 **Nombre Historia de Usuario:** Alertar sobre visita efectuada sin resumen.

Modificación de Historia de Usuario Número: ninguna

Sistema de planificación de visitas para el Centro de Soporte

Usuario: Yoel Antonio Casado Gonzáles	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 0.4 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 0.4 semanas
Descripción: Permite tener al tanto al responsable de la visita y al planificador que no se ha cumplido con el cierre de una visita efectuada en un periodo superior a las 24 horas pasadas la fecha de la vista.	
Observaciones: Debe haber una visita en la que se haya incumplido el tiempo para añadir el resumen de la misma.	
Prototipo de interfaz:	

Historia de Usuario	
Número: HU_10	Nombre Historia de Usuario: Consultar visitas planificadas.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoel Antonio Casado Gonzáles	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 0.4 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 0.4 semanas
Descripción: permite visualizar las fechas y destinos de las visitas en las que un especialista está involucrado.	
Observaciones: El especialista debe tener alguna visita asociada.	
Prototipo de interfaz:	

Historia de Usuario	
Número: HU_13	Nombre Historia de Usuario: Pagar los resultados de las búsquedas
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoel Antonio Casado Gonzáles	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 0.6 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 0.6 semanas
Descripción: Permite organizar las visitas encontradas con el fin de mejorar la visualización de los resultados.	
Observaciones:	
<ul style="list-style-type: none"> - Se debe tener acceso al módulo de búsquedas y debe haber visitas generadas. - Se debe haber realizado al menos una búsqueda. 	
Prototipo de interfaz:	

Sistema de planificación de visitas para el Centro de Soporte

Historia de Usuario	
Número: HU_14	Nombre Historia de Usuario: Búsqueda avanzada de visitas
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoel Antonio Casado Gonzáles	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 0.8 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 0.8 semanas
Descripción: Permite obtener resultados a través de la búsqueda por valores específicos, de acuerdo a los parámetros permitidos por el sistema.	
Observaciones: Se debe tener acceso al módulo de búsquedas y debe haber visitas generadas.	
Prototipo de interfaz:	

Historia de Usuario	
Número: HU_16	Nombre Historia de Usuario: Crear informes personalizados
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoel Antonio Casado Gonzáles	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 0.6 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 1 semanas
Descripción: Permite diseñar un informe de visitas que contenga un encabezado, los logos UCI y del centro de soporte. El informe mostrará el contenido acorde a la personalización realizada por el usuario.	
Observaciones: El informe de realizará de acuerdo a parámetros disponibles en el sistema y podrá realizarse a diferentes vistas del mismo.	
Prototipo de interfaz:	

Historia de Usuario	
Número: HU_17	Nombre Historia de Usuario: Exportar informes a PDF.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yoel Antonio Casado Gonzáles	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 0.4 semanas
Riesgo en Desarrollo: Alto	Puntos Reales: 0.4 semanas
Descripción: Permite almacenar el informe en un dispositivo externo al sistema.	
Observaciones: Se debe haber creado previamente el informe.	

Sistema de planificación de visitas para el Centro de Soporte

Prototipo de interfaz:

Historia de Usuario

Número: HU_18 **Nombre Historia de Usuario:** Exportar resumen de las visitas ejecutas a PDF

Modificación de Historia de Usuario Número: ninguna

Usuario: Yoel Antonio Casado Gonzáles **Iteración Asignada:** 3

Prioridad en Negocio: Alta **Puntos Estimados:** 0.4 semanas

Riesgo en Desarrollo: Alto **Puntos Reales:** 0.4 semanas

Descripción: Obtiene el resumen de las visitas completadas en un informe.

Observaciones: Se puede obtener por entidad, por fecha o ambas, dependiendo de la necesidad de usuario.

Prototipo de interfaz:

Historia de Usuario

Número: HU_20 **Nombre Historia de Usuario:** Actualizar categorías.

Modificación de Historia de Usuario Número: ninguna

Usuario: Yoel Antonio Casado Gonzáles **Iteración Asignada:** 1

Prioridad en Negocio: Alta **Puntos Estimados:** 0.8 semanas

Riesgo en Desarrollo: Alto **Puntos Reales:** 1 semanas

Descripción: Evita que ocurran errores entre la plataforma de soporte y el SGV, por lo que se obtendrán las categorías y subcategorías de los servicios web que ofrece la plataforma.

Observaciones: Es preciso ejecutar esta tarea de forma automática para que esté acorde con la plataforma de soporte.

Prototipo de interfaz: