



Universidad de las Ciencias Informáticas
Facultad 2

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Sistema para la gestión de incidencias y distribución de la fuerza de trabajo asociada.

Autor:

Sheila de la Caridad Tiá Ubeda.

Tutores: Ing. Arianna Pérez Carmenates.

Ing. Yoanny Torres Rubio.

Co-tutor: Ing. Sailyn María Parra López.

11 de junio del 2018

“Año 60 del Triunfo de la Revolución”.



“Seamos realistas y hagamos lo imposible.”

Ernesto Che Guevara

DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis que tiene por título: Sistema para la gestión de incidencias y distribución de la fuerza de trabajo asociada 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 ____ días del mes de _____ del año _____.

Sheila de la Caridad Tiá Ubeda
Autor

Ing. Yoanny Torres Rubio
Tutor

Ing. Arianna Pérez Carmenates
Tutor

Ing. Saily María Parra López
Co-Tutor

DATOS DE CONTACTO

Tutor

Nombre y Apellidos: Arianna Pérez Carmenates

Sexo: F. **Institución:** UCI.

Dirección de la institución: Carretera a San Antonio de los Baños, km 2 ½, Boyeros, Ciudad de La Habana.

Correo electrónico: acarmenates@uci.cu **Teléfono del trabajo:** 8373768.

Categoría docente: Asistente.

Área: Centro Telemática.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2008

Institución donde se graduó: Universidad de las Ciencias Informáticas.

Tutor

Nombre y Apellidos: Yoanny Torres Rubio

Sexo: M. **Institución:** UCI.

Dirección de la institución: Carretera a San Antonio de los Baños, km 2 ½, Boyeros, Ciudad de La Habana.

Correo electrónico: ytrubio@uci.cu **Teléfono del trabajo:** 8373768.

Categoría docente: Instructor.

Área: Centro Telemática.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2008

Institución donde se graduó: Universidad de las Ciencias Informáticas.

Co-Tutor

Nombre : Sailyn María Parra López

Sexo: F **Institución:** UCI

Dirección de la institución: Carretera a San Antonio de los Baños, km 2 ½, Boyeros, Ciudad de La Habana.

Correo electrónico: smparra@uci.cu **Teléfono del trabajo:** 8372142.

Área: Centro Telemática.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2015

Institución donde se graduó: Universidad de las Ciencias Informáticas.

AGRADECIMIENTOS

A mi familia por apoyarme y en especial a mi mamá, a quien se lo debo todo. A mi novio Sergio por ser mi guardián y compañero en esta etapa tan difícil. A mis tutores y en especial a Yoanny, sin él no hubiera podido hacerlo. A mi oponente por apoyarme y a los que han formado parte del tribunal por guiarme. A todos mis profesores, todos ellos responsables de mi formación como profesional y persona, en 5 años se vuelven más que profes, Nolberto, que no me miraba al exponer los trabajos para que no gagueara, a Yaniseli y Leonid por sacarme de tantos aprietos, los conocí como profes y hoy son mis amigos. A mis tías del comedor por echarme un poquito más, y en especial a Yusi. A mi amiga Katia que no pudo estar aquí, Jessie sabes que te i love you. A mi ex compañero de cuarto Julio, por tenderme la mano cuando más lo necesité,

¡A TODOS MUCHAS GRACIAS!!!!

DEDICATORIA

A mi mamá.

Resumen

Realizar una gestión eficaz de las incidencias reportadas a las empresas encargadas de brindar servicios y la distribución equitativa de la fuerza de trabajo asociada a esta, es hoy un factor determinante en su calidad, en la tenencia y mantenimiento de una fiel clientela. En el contexto empresarial cubano, gestionar las incidencias y destinar el personal encargado de solucionarlas, constituyen operaciones que se vuelven cada vez más complicadas. Esto implica gran cantidad de tiempo en realizar dichas acciones. Escasea su gestión y de existir, la información se registra en formato duro, o digital como Word u hoja de cálculo. Resultando complejo manejar los mismos.

Para dar solución a este inconveniente, después de haber realizado un análisis de tendencia actual de los sistemas de gestión de incidencias, se desarrolló un sistema que es capaz de gestionar las incidencias y distribuir la fuerza de trabajo asociada a esta. Para el desarrollo del mismo se empleó Visual Paradigm como herramienta de modelado, como lenguajes de programación: Python 2.7.6, HTML 5, CSS3, JavaScript 1.9.1, como servidor de base de datos PostgreSQL 9.4.1, como framework de desarrollo: Django 1.7, BackboneJS, Django-Rest-Framework, Xilema base web y como IDE de desarrollo PyCharm 2017.1.3. La metodología de desarrollo de software seleccionada fue Programación Extrema. El sistema creado posee un gran impacto en la sociedad debido a que es genérico, posee asignación automática de incidencias, no es privado y emplea un mecanismo de distribución de carga de trabajo.

Palabras clave: cliente, distribuir fuerza de trabajo, empresa, gestión de incidencias, servicios.

Abstract

Efficient management of the incidents reported to the companies in charge of providing services and the equitable distribution of the workforce associated with it, is today a determining factor in its quality, in the possession and maintenance of a loyal clientele. In the Cuban business context, managing incidents and assigning the personnel in charge of solving them, are becoming increasingly complicated operations. Which take a great deal of time to be performed. There is a lack of management and if this is not the case, the information is recorded in hard format, or in digital format such as Word or spreadsheet, being complex to manage them.

In order to solve this problem, after having carried out a current trend analysis of incident management systems, a system capable of managing incidents and distributing the associated workforce was developed. Visual Paradigm was used as a modeling tool, as programming languages were used: Python 2.7.6, HTML 5, CSS3, JavaScript 1.9.1, PostgreSQL 9.4.1 database server, as development framework: Django 1.7, BackboneJS, Django-Rest-Framework, Xilema web base and as PyCharm 2017.1.3 development IDE. The software development methodology selected was Extreme Programming. The system created has a great impact on society because it is generic, has automatic assignment of incidents, is not private and uses a workload distribution mechanism.

Keywords: client, distribution of workforce, company, incident management, services.

Índice

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DEL SISTEMA PARA LA GESTIÓN DE INCIDENCIAS Y DISTRIBUCIÓN DE LA FUERZA DE TRABAJO ASOCIADA.....	5
1.1 CONCEPTOS RELACIONADOS.....	5
1.2 SISTEMAS PARA LA GESTIÓN DE INCIDENCIAS (SGI) EXISTENTES EN EL MUNDO..	6
1.3 SISTEMAS PARA LA GESTIÓN DE INCIDENCIAS (SGI) EXISTENTES EN CUBA	8
1.4 MECANISMOS DE DISTRIBUCIÓN DE LA CARGA DE TRABAJO	10
1.5 HERRAMIENTAS, TECNOLOGÍAS Y METODOLOGÍA.....	12
1.5.1 Metodología de desarrollo de software	12
1.5.2 Herramienta CASE.....	14
1.5.3 Lenguajes de programación.....	15
1.5.4 Servidor de base de datos	16
1.5.5 Framework de desarrollo.....	16
1.5.6 Entorno Integrado de desarrollo.....	18
1.6 Conclusiones parciales.....	18
CAPÍTULO 2: EXPLORACIÓN, PLANIFICACIÓN Y DISEÑO DEL SISTEMA...19	
2.1 PROPUESTA DEL SISTEMA.....	19
2.2 CARACTERÍSTICAS NO FUNCIONALES DEL SISTEMA.....	20
2.3 FASE DE EXPLORACIÓN	21
2.3.1 Involucrados en el sistema.....	22
2.3.2 Historias de Usuarios (HU).....	22
2.4 FASE DE PLANIFICACIÓN.....	26
2.4.1 Estimación de esfuerzo	26
2.4.2 Plan de iteraciones	27
2.4.3 Plan de entrega	29
2.5 FASE DE DISEÑO.....	29
2.5.1 Arquitectura	30
2.5.2 Patrón arquitectónico.....	30
2.5.3 Patrones de diseño	32
2.5.4 Tarjetas CRC.....	33
2.5.5 Diseño de la base de datos	34

2.6 CONCLUSIONES PARCIALES.....	35
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA PARA LA GESTIÓN DE INCIDENCIAS Y DISTRIBUCIÓN DE LA FUERZA DE TRABAJO ASOCIADA.....	36
3.1 IMPLEMENTACIÓN.....	36
3.1.1 <i>Tareas de ingeniería (TI)</i>	36
3.2 PRUEBAS.....	39
3.2.1 <i>Pruebas Unitarias</i>	39
3.2.2 <i>Pruebas de Aceptación</i>	40
3.3 CONCLUSIONES PARCIALES.....	43
CONCLUSIONES GENERALES.....	45
RECOMENDACIONES.....	46
REFERENCIAS BIBLIOGRÁFICAS.....	47
BIBLIOGRAFÍA	50
ANEXOS.....	53
ENTREVISTAS REALIZADAS.....	53

Índice de Tablas

Tabla 1 Comparación de sistemas gestores de incidencias estudiados	9
Tabla 2 Especificaciones de hardware en el servidor	21
Tabla 3 Especificaciones de hardware en el cliente	21
Tabla 4 Especificaciones de software	21
Tabla 5 Involucrados en el sistema	22
Tabla 6 HU 1 Insertar incidencia	23
Tabla 7 HU 21 Insertar Técnico	24
Tabla 8 HU 31 Seleccionar opción de distribución	25
Tabla 9 HU 32 Distribuir carga de trabajo	25
Tabla 10 Estimación de esfuerzo	26
Tabla 11 Plan de iteraciones	28
Tabla 12 Plan de entrega	29
Tabla 13 Tarjeta CRC Técnico	33
Tabla 14 Tarjeta CRC Incidencia	33
Tabla 15 Tareas de Ingeniería	36
Tabla 16 Tarea de Ingeniería HU Insertar incidencia	37
Tabla 17 Tarea de Ingeniería HU Insertar técnico	37
Tabla 18 Tarea de Ingeniería HU Seleccionar opción de distribución	38
Tabla 19 Tarea de ingeniería HU Distribuir carga de trabajo (Manual)	38
Tabla 20 Tarea de Ingeniería 20 HU Distribuir carga de trabajo (Automática)	38
Tabla 21 Prueba de aceptación HU Insertar incidencia	41
Tabla 22 Prueba de aceptación HU Insertar técnico	42
Tabla 23 Prueba de aceptación HU Seleccionar opción de distribución	42
Tabla 24 Prueba de aceptación HU Distribuir carga de trabajo	42

Índice de Figuras

Fig. 1 Algoritmo MEDM	12
Fig. 2 Estrella de Boehm y Turner	14
Fig. 3 Propuesta del sistema	19
Fig. 4 Arquitectura Cliente-Servidor	30
Fig. 5 Patrón arquitectónico MTV	31
Fig. 6 Clases del Modelo	32
Fig. 7 Clases de la Vista	32
Fig. 8 Clases de la Plantilla	32
Fig. 9 Modelo lógico de la base de datos	34

Fig. 10 Modelo físico de la base de datos	35
Fig. 11 Resultado de las pruebas unitarias	40
Fig. 12 Descripción de los resultados de las pruebas unitarias.....	40
Fig. 13 Descripción de los resultados de las pruebas de aceptación.....	43

Introducción

Durante un largo período de la historia del pensamiento económico, los “bienes inmateriales”, como eran llamados los servicios, no fueron tomados en consideración principal de los afanes de los estudiosos de la realidad social. Sin embargo, el impulso seguido en las últimas décadas por las economías y sociedades más desarrolladas ha puesto de relieve que estos se han convertido en la forma dominante de actividad económica. (Valotto Patuzzo, 2011)

El tener un enfoque bien definido hacia el cliente y una orientación hacia el servicio constituye un reto para toda organización. La asistencia que se ofrece a los usuarios es importante siempre y actualmente constituye un elemento básico para el éxito o fracaso de muchas empresas, ya que una relación comprensiva ante cualquier tipo de reclamo conforma una ventaja competitiva. Esto permite evitar disímiles problemas o incidencias que puedan ocurrir. Esta última es una fuente inagotable de información útil para la empresa. A través de ella, los clientes se ponen en contacto para mostrar su insatisfacción e informar que algo no está funcionando como debería, aspecto que frecuentemente es difícil de detectar por la propia compañía en ese mismo momento. (IntegralMS, 2016)

Las incidencias reportadas son una oportunidad para afianzar la relación con el cliente, ya que se sentirá atendido, escuchado y como parte valiosa que aporta información de mejora a la empresa. La comunicación, siendo una condición necesaria para la existencia del hombre, es un factor clave para reducir la tensión y plasmar una buena imagen de la empresa. Cuanto más rápida sea la respuesta y más personal sea el método para comunicarla a las personas perjudicadas, mayor satisfacción se logrará por el trato recibido. Ofrecer calidad en el momento de la entrega del servicio constituye la mejor publicidad que se puede ofrecer a los clientes.

Si la empresa escucha con atención logra conseguir la lealtad de sus usuarios. En caso contrario, este buscará otro proveedor que satisfaga sus requerimientos. Existen organizaciones que manifiestan y afirman con orgullo que no reciben reclamaciones, esto no significa que todos sus clientes estén satisfechos. En realidad, pierden muchas oportunidades importantes para el mejoramiento de su desempeño. (González, 2011)

Las incidencias reportadas, constituyen un valioso tesoro para la empresa, puesto que proporcionan una invaluable retroalimentación sobre la calidad del servicio. Surge con nitidez la importancia fundamental que tiene para una organización responder ante las incidencias reportadas, lo que le permitirá posteriormente establecer acciones de mejora.

Según la Norma ISO 9000, 2005 Sistemas de gestión de la calidad — Fundamentos y vocabulario, define el concepto de satisfacción del cliente como su percepción sobre el grado en que se han cumplido sus requisitos. Aclara además un aspecto muy importante sobre las incidencias: su existencia es un claro indicador de una baja satisfacción, pero su ausencia no implica necesariamente una elevada satisfacción del cliente, ya que también podría estar indicando que son inadecuados los métodos de comunicación

entre este y la empresa, o que las incidencias se reportan, pero no se registran adecuadamente, o que simplemente el cliente insatisfecho, en silencio, cambia de proveedor.

La mejor manera de fidelizar clientela es gestionar correctamente las incidencias reportadas sobre el servicio brindado, puesto que transmite una imagen de empresa interesada en la atención y asistencia al cliente. La organización debe mostrar en el desarrollo del proceso de tratamiento y gestión de incidencias su disposición a la retroalimentación a partir de las experiencias negativas de los usuarios, esto hará que el servicio se convierta en la mejor referencia.

La información registrada de una incidencia debe estar almacenada y clasificada de forma que permita su consulta posterior y así facilitar su seguimiento a lo largo de todo el proceso. Es de vital importancia, que las empresas encargadas de brindar servicios sean capaces de solucionar sus errores o fallos y de hacerlo en el menor tiempo posible. Tener supervisado cuáles son las inconformidades y el personal destinado a solucionarlas, que debe estar disponible en todo momento. Valorar la carga de trabajo de cada uno de ellos, con el objetivo de reducir el tiempo dedicado a la gestión y asignación de las incidencias, con eficiencia, garantizando así un servicio completo y de calidad. La rapidez y destreza que desempeñen a la hora de disipar una incidencia presentada puede ser un hecho determinante para fidelizar clientela. (KOTLER, 2008)

En Cuba, el desarrollo de la informática y las comunicaciones es un proceso al cual se le ha dedicado gran empeño y recursos, buscando un continuo avance, que posibilite una mejora constante en los sectores de la sociedad. Ejemplo de ello es el Departamento de Componentes del Centro de Telemática perteneciente a la Universidad de Ciencias Informáticas, encargado de brindar servicio de consultorías e instalación de centros de datos y servicios a las entidades nacionales. La experiencia de trabajo del centro corrobora que, en el contexto empresarial cubano, gestionar las incidencias y destinar el personal encargado de solucionarlas, constituyen operaciones que se vuelven cada vez más complicadas. Implicando gran cantidad de tiempo realizar dichas acciones.

Se ha comprobado que aproximadamente en más de un 50% de las instituciones no existe un proceso de gestión de incidencias, en caso de existir, se realiza comúnmente en formato duro o en documentos digitales como Word u hoja de cálculo, dificultando el manejo y control de dicha información. Presenta gran complejidad la recogida de las incidencias, así como su almacenamiento y búsqueda, ya que al tener un gran cúmulo de datos que tratar en estos formatos resulta muy engorroso realizar cualquier proceso de organización, consulta o control.

Un sistema de gestión manual se basa en gran medida en las acciones de las personas, lo que aumenta la posibilidad de errores humanos, proporcionando duplicidad, pérdida por mala manipulación y alteración de la información. Otro de los factores que entorpecen el proceso es el tiempo, ya que la demora es realmente un factor determinante para la calidad de una asistencia. Todo lo anteriormente expuesto se traduce en falta de eficiencia, afectando la calidad del servicio brindado, siendo caracterizado este como insuficiente.

La gran heterogeneidad de ambientes laborales dificulta el uso de sistemas gestores de incidencias comunes, necesitando cierta personalización según el flujo de trabajo de cada entidad. Muchas soluciones factibles existentes en el mercado son basadas en sistemas privativos, deviniendo esto en gastos por el pago de licencias. La mala atención al cliente es uno de los aspectos más agravantes apreciados, debido a que no se le da la suficiente importancia al trato emitido, siendo este la razón de ser de cualquier negocio que brinde servicios.

Otro aspecto contundente es que las empresas no cuentan frecuentemente con mecanismos para distribuir la carga de trabajo de forma equitativa al personal destinado a solucionar las incidencias, por lo que este proceso se realiza manualmente también.

Al no valorarse este aspecto, se sobrecargan algunos y quedan ociosos otros, causando descontento e inconformidades. La incorrecta distribución de la carga de trabajo, poco a poco se va manifestando en el comportamiento de las personas, mostrando falta de motivación, propiciando conflictos y agravios, ausentismo, agresividad, incapacidades, problemas estratégicos con personas claves sobrecargadas, se empeora el descanso y aumenta el desgaste físico y mental del trabajador. Afectando directamente el clima organizacional y por ende la productividad de la organización. (ScienceDirect, 2017)

A partir de la anterior **situación problemática**, se deriva el siguiente **problema de investigación**:
¿Cómo gestionar las incidencias reportadas por los clientes de empresas que brindan servicios y la distribución de la fuerza de trabajo asociada?

Se enmarca el **objeto de estudio** en el proceso de la gestión de incidencias y la distribución de la fuerza de trabajo asociada de empresas que brindan servicios.

Se propone el siguiente **objetivo general**: desarrollar una aplicación informática que permita realizar la gestión de las incidencias reportadas por los clientes de empresas que brindan servicios y realizar la distribución de la fuerza de trabajo asociada.

Se enmarca la investigación en el **campo de acción**: herramientas informáticas para la gestión de incidencias y la distribución de la fuerza de trabajo asociada en empresas que brindan servicios.

Trazando como **tareas de investigación** para guiar la misma:

- Análisis del proceso de gestión de incidencias en empresas que brindan servicios para entender su funcionamiento e identificar prácticas recomendadas.
- Análisis de los sistemas similares para obtener características comunes.
- Análisis de mecanismos de distribución de la fuerza de trabajo para identificar el más idóneo para la propuesta de solución.
- Fundamentación de la metodología, herramientas y lenguaje de programación seleccionadas para guiar el desarrollo del software.
- Análisis de la metodología seleccionada para el posterior diseño e implementación del sistema.
- Análisis de las pruebas propuestas por la metodología seleccionada para la validación del software desarrollado.

Se emplearon los siguientes métodos de investigación:

Métodos Teóricos:

Análítico-sintético: Se analiza la documentación referente al proceso de gestión de la fuerza de trabajo móvil en los sistemas gestores de incidencias, brindando información y datos claves en el proceso.

Métodos Empíricos:

Entrevista semiestructurada: se realizan entrevistas a administradores de sitios gestores de incidencias para estudiar su funcionamiento, tomando así lo positivo y negativo de éstos. (Ver en [Anexos](#))

Observación: Empleado para tomar experiencias mediante la interacción con sistemas gestores de incidencias para analizar los flujos de gestión de las mismas.

La estructura del presente trabajo consta de 3 capítulos, definidos a continuación:

Capítulo 1: Fundamentación teórica del sistema Sistema para la gestión de incidencias y distribución de la fuerza de trabajo asociada.

En este capítulo se realiza un estudio de los sistemas existentes en el ámbito nacional e internacional relacionados con la gestión de incidencias. Se analizan sus procesos de gestión, así como los mecanismos de distribución de la carga de trabajo existentes. Se caracteriza la metodología seleccionada para guiar el proceso de desarrollo del sistema. Se caracterizan la herramienta CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) empleada, los lenguajes de programación, el sistema gestor de base de datos, el marco de trabajo y el entorno de desarrollo integrado para el desarrollo del sistema.

Capítulo 2: Exploración, planificación y diseño del sistema.

Constituido por la descripción del sistema propuesto y el diseño de la aplicación, partiendo de la fundamentación teórica realizada en el capítulo anterior y teniendo en cuenta las fases de exploración, planificación y diseño definidas por la metodología seleccionada. Se definen también los elementos importantes dentro del proceso de desarrollo tales como: las historias de usuario (HU), el plan de iteraciones, el plan de entregas, la arquitectura, los patrones de diseño y las tarjetas CRC (Clase, Responsabilidad y Colaboración).

Capítulo 3: Implementación y pruebas.

En este capítulo se muestra el desarrollo de las fases de implementación y pruebas definidas por la metodología seleccionada. Se realiza una descripción de los principales artefactos a generar, como son: las tareas de ingeniería por cada HU identificada y las pruebas empleadas para validar la solución propuesta.

Capítulo 1: Fundamentación teórica del Sistema para la gestión de incidencias y distribución de la fuerza de trabajo asociada.

Durante las últimas décadas el mundo se ha enfrentado a continuas revoluciones en la esfera científico técnica, la era de la información y el conocimiento está transformando los sistemas económicos, políticos y sociales. Los sistemas de gestión representan un factor determinante en la actualidad. Este capítulo engloba los conceptos y definiciones fundamentales respecto a los sistemas de gestión de incidencias y la distribución de la carga de trabajo asociada. Se estudian los sistemas existentes en Cuba y el mundo orientados a la gestión de incidencias, así como los mecanismos de distribución de la fuerza de trabajo para arribar a conclusiones respecto a buenas prácticas y tendencias actuales. Además, se realiza un análisis que fundamenta el uso de las distintas herramientas, tecnologías y metodología para el desarrollo del sistema propuesto.

1.1 Conceptos relacionados

Para un mayor entendimiento de la problemática planteada anteriormente se hace necesario explicar algunos términos relacionados con la gestión de incidencias. Estos términos se muestran a continuación.

Incidencia

Una incidencia es toda interrupción o reducción de la calidad no planificada del servicio. Pueden ser fallos o consultas reportadas por los usuarios, el equipo del servicio o por alguna herramienta de monitorización de eventos. (SERVICETONIC, 2017)

Ticket

Un ticket del incidente se utiliza para comunicar y gestionar incidencias como una interrupción, falta de disponibilidad, reducción de la calidad de un sistema o del servicio. Los tickets del incidente se gestionan mediante procesos del flujo de trabajo de gestión de incidentes. (Desk-Spanish-Spain, 2017)

Servicio

El análisis conceptual de los servicios consiste en entender que el servicio es fundamentalmente diferente de un bien o de un producto. Servicio es trabajo en proceso y no el resultado de la acción del trabajo; por esa razón fundamental, no se produce un servicio, sino que se presta un servicio. (Valotto Patuzzo, 2011)

Transferencia de Estado Representacional (REST)

Cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. (BBVA Open4U [sin fecha])

Clúster

En un sistema informático, un clúster es un grupo de servidores y otros recursos que actúan como un único sistema y permiten una alta disponibilidad y, en algunos casos, equilibrio de carga y procesamiento paralelo. (IBM DeveloperWorks®, 2002)

Middleware

Software que brinda servicios a aplicaciones de software más allá de las disponibles en el sistema operativo. El middleware facilita que los desarrolladores implementen comunicación y entrada/salida, para que puedan enfocarse en el propósito específico de su aplicación. (InternetArchiveWayBackMachine, 2007)

1.2 Sistemas para la gestión de incidencias (SGI) existentes en el mundo

La investigación sobre los SGI mostró que existe una tendencia creciente de su utilización en la actualidad. Al realizar un análisis sobre estos sistemas a nivel internacional y nacional, se encontraron diversas soluciones, a continuación, se ofrece una breve descripción de algunas de ellas.

IntegrialMS (Integria Incident Manager Sistem)

IntegrialMS es un software de asistencia de escritorio sencillo, potente y fácil de usar. Dispone de inventario automatizado y controlado desde una sola interfaz. Agiliza los procesos reduciendo los tiempos y costes de gestión en sus proyectos. Posee informes personalizados para ver fácilmente la gestión y calidad del servicio dado en tiempo real. (Integria, 2018)

Brinda facilidades como:

- Automatización y notificaciones de tareas rutinarias mediante potentes flujos de trabajo.
- Creación, seguimiento y visualización intuitiva de los tickets.
- Creación, actualización y seguimiento de las incidencias por email.
- Personalización: adaptable a diversas necesidades mediante múltiples campos personalizados.
- Satisfacción del servicio mediante el sistema de evaluación de calidad.
- Seguridad y permisos: mediante roles, perfiles y grupos.
- Informes: accede fácilmente a la información, visualiza tendencias y encuentra rápidamente los puntos débiles del servicio.
- Sistema de artículos, ficheros y soluciones que ayuden al cliente.
- Visualización por tipos y categorías configurables.
- Los tickets pueden enlazarse fácilmente para que alimenten la base de conocimientos.
- Multi-idioma.

Integria es privativo y puede limitar el número de tickets abiertos en total por cada grupo, pero no por cada operador. No tiene forma de distribuir equitativamente la carga de trabajo ni asignación automática de tickets.

Open Technology Real Services (OTRS)

Software de código abierto gratuito y soluciones de software de negocios profesionales. OTRS es uno de los sistemas de tickets web más flexibles del mercado y se emplea en la asistencia a clientes, la asistencia de escritorio y la gestión de servicios informáticos. De implementación fácil y con una adaptación sencilla a diversidad de necesidades. (OTRS, 2017)

OTRS está separado en varios componentes. El principal es el marco que contiene todas las funciones centrales para la aplicación y el sistema de tickets. A continuación, se muestran los servicios que ofrece el Grupo OTRS:

- OTRS usa Tickets para reunir todas las comunicaciones internas externas que pertenecen juntas. Estas entradas están organizadas en Colas.
- Hay muchas maneras diferentes de mirar los tickets en un sistema (basado en colas, estado, escalada, etc.) en diferentes niveles de detalle (pequeño / mediano / vista previa).
- El historial del ticket registra todos los cambios en un ticket.
- Los tickets se pueden cambiar de muchas maneras, como responder, reenviar, rebotar, moverse a otra cola, actualización de atributos (estado, prioridad, etc.), bloqueo y contabilidad.
- Es posible modificar muchos tickets a la vez (acción masiva).
- Permite la programación basada en el tiempo y restricciones en tickets.
- Los tickets se pueden vincular a otros tickets u otros objetos, como entradas de preguntas frecuentes.
- Las acciones automáticas y programadas en los tickets son posibles con el "GenericAgent".
- OTRS viene con un poderoso motor de búsqueda que permite búsquedas complejas y de texto completo en tickets.

Señalar que OTRS hace uso de mecanismos de distribución de carga de trabajo para la asignación de incidencias. En su versión libre no provee de asignación automática de tickets, se apoya en colas genéricas, pero en su versión privativa OTRS Business Solution™, sí. Hace uso de servicios REST.

ManageEngine ServiceDesk Plus

Sistema privativo. Permite consolidar, dentro de una única plataforma de software, todos los recursos de soporte para múltiples clientes. Se puede personalizar el servicio para cada cliente y gestionar de forma individual los aspectos más importantes del servicio. Entre sus prestaciones se encuentra facturación automática, catálogo de servicios, inventario de hardware y software, asignación automática, gestión de incidencias, entre otros. (Zoho Corp, 2017)

La funcionalidad de gestión de incidencias brinda facilidades como:

- Gestión de solicitudes como incidencias y peticiones de servicio.
- Reglas de negocio para clasificación y asignación automática de solicitudes.
- Asignación automática de tickets por balanceo de carga.
- Integración con correo electrónico POP/IMAP/SMTP.
- Base de datos de conocimientos.
- Notificaciones y alertas personalizadas.
- Comandos de email para clasificar automáticamente las solicitudes.
- Integración con Directorio Activo/LDAP.
- Acciones de respuesta automatizadas.

- Portal de autoservicio para clientes.
- Gestión de tareas, incluyendo tareas programadas de mantenimiento preventivo.
- App para dispositivos móviles (iOS y Android)
- Encuestas de satisfacción del cliente.

Como inconveniente cabe mencionar que el sistema es privado, por lo que no se tiene acceso al código fuente, no permite desagregar los módulos que no se necesitan y se pagan las licencias.

1.3 Sistemas para la gestión de incidencias (SGI) existentes en Cuba

Las empresas cubanas se han percatado de la necesidad e importancia del uso de estos sistemas. Ejemplo de ello es la Universidad de las Ciencias informáticas, la cual cuenta con SGI encargados de gestionar las incidencias de diferentes áreas de la misma.

La Dirección de Gestión Tecnológica (DGT), utiliza una personalización de **OTRS**, la gestión de las incidencias mediante este sistema se basa en la forma de trabajo que se implementa en las áreas que tienen el flujo de trabajo montado en él. Según el tipo de incidencia que sea, que este dato se registra cuando se crea el ticket, ya sea mediante el sitio o una llamada telefónica, es el área que le corresponde solucionarla, esta se le asigna a su respectivo jefe y este a su vez a sus subordinados. La distribución de la carga de la fuerza de trabajo queda a consideración de los jefes del área que son los que distribuyen sus tickets según su personal disponible, manualmente (L. Pérez Mejías, comunicación personal, octubre 23, 2017).

Este mismo sistema es el usado por las áreas de Mantenimiento e Informatización de la propia universidad, en el sistema se controlan las incidencias de estos departamentos ya que tiene implementado los flujos de trabajo de cada uno corriendo sobre la misma plataforma, trayendo como consecuencia que los tickets estén organizados por números de serie secuenciales en general entre ambos departamentos y no separado por cada uno. De igual forma es usado por el departamento de Dirección de Redes y Seguridad Informática (DSI), este último solo tiene implementado su flujo de trabajo y es el mismo funcionamiento (M. González Cruz, comunicación personal, octubre 25, 2017).

El Centro de Soporte de la universidad gestionan sus tickets por la plataforma para la gestión de incidencias **ManageEngine ServiceDesk Plus 9.0**, mediante este el registro de la incidencia se realiza de tres formas, mediante un correo que automáticamente crea el ticket, directamente el usuario en el portal web o mediante una llamada telefónica y un personal designado a esta tarea lo registra en el portal. La asignación de los tickets se realiza manualmente, mediante una lista con los especialistas que cuenta el centro y de esta manera se le asocia la incidencia, de igual forma la distribución de la fuerza de trabajo no se tiene informatizada. (N. Lago Clara, comunicación personal, octubre 29, 2017).

Gestión de la fuerza de trabajo móvil para la red de acceso de ETECSA.

Aplicación web que permite organizar la fuerza de trabajo móvil para la red de acceso de ETECSA con el objetivo de lograr una correcta planificación y programación de las tareas a realizar por los técnicos de dicha entidad. La aplicación cuenta con tres subsistemas principales que se explican a continuación.

Subsistema de estadísticas descriptivas:

El sistema realiza una estadística descriptiva mostrando el comportamiento de los mantenimientos realizados por la empresa. La estadística muestra por días de la semana, indicadores para la interpretación de los resultados tales como el total, el promedio, el coeficiente de variación, la varianza y la desviación típica o desviación estándar.

Subsistema de planificación de tareas:

Realiza una planificación de la fuerza de trabajo necesaria para darle solución a las actividades pendientes, teniendo en cuenta parámetros que comúnmente no se consideraban como, la cantidad de técnicos disponibles, el tiempo de viaje según la zona, el tiempo efectivo de trabajo, la cantidad de interrupciones pendientes y su normativa de tiempo de ejecución. Muestra un resumen con importantes datos necesarios para la correcta provisión de servicio.

Subsistema de programación de tareas:

Este subsistema cuenta con un algoritmo que tiene en cuenta para la programación de tareas, los técnicos que están de descanso o no, el tiempo extra de cada tarea, el tiempo de viaje de la zona, los tipos de actividades y la prioridad que tiene cada una. Teniendo en cuenta todos esos parámetros, el algoritmo asigna las tareas a los técnicos logrando un balanceo de cargas, de forma tal que nunca queden técnicos muy sobrecargados y otros muy libres de tareas. El subsistema muestra un listado de las tareas asignadas a cada técnico especificando hora inicio y hora fin de cada una. (Vazquez Garcia y García Fuentes, 2012)

Tabla 1 Comparación de sistemas gestores de incidencias estudiados

Softwares	Privado	Empleo de mecanismos de distribución de carga de trabajo	Genérico	Asignación de tickets automática
OTRS	Posee versión libre y privada	Sí	Sí	En su versión libre no y en su versión privada sí.
Gestión de la fuerza de trabajo móvil para la red de acceso de ETECSA	NO	Sí	No	Sí
IntegralIMS	Sí	No	Sí	No
ManageEngine ServiceDesk	Sí	Sí	Sí	Sí

Plus				
------	--	--	--	--

Los SGI estudiados cumplen con las características necesarias para los cuales fueron creados, pero no se ajustan a las cualidades que se requieren para darle solución a la problemática de este trabajo ya que el nuevo sistema debe ser genérico, para que se pueda adaptar a los diferentes flujos de gestión de incidencias de las empresas. Emplear un mecanismo de distribución de la carga de trabajo asociada para equilibrar la carga de trabajo teniendo en cuenta diversos parámetros que facilitan y optimizan el proceso de asignación. Ser de código abierto (open source) para así brindar la posibilidad de reutilizar el código fuente en versiones posteriores para que sea homogéneo e interoperable. Tener asignación automática de tickets, para minimizar el tiempo de asignación de estas. Demostrando así la necesidad de un nuevo sistema que cumpla con estos requerimientos.

1.4 Mecanismos de distribución de la carga de trabajo

La propuesta de solución tiene como forma de gestión de la fuerza de trabajo asociada la lógica de funcionamiento de los mecanismos de balanceo de carga para la distribución equitativa de la misma. El estudio de las cargas de trabajo es un tema que toma fuerza en el nuevo milenio conforme el trabajo se vuelve cada vez más administrativo y cuya medición es muy difícil desarrollar si no se cuenta con una técnica que permita obtener dicha medición. Se vuelve un arma esencial para la ventaja competitiva de las organizaciones, cuyo objetivo es optimizar los recursos haciendo cada vez más productivos sus procesos. Existen varios conceptos del tema, a continuación, se citan algunos.

El balance o balanceo de carga es un concepto usado en informática que se refiere a la técnica usada para compartir el trabajo a realizar entre varios procesos, ordenadores, discos u otros recursos. Está íntimamente ligado a los sistemas de multiprocesamiento, o que hacen uso de más de una unidad de procesamiento para realizar labores útiles. (Gil Dmínguez, 2015)

El balance de carga se mantiene gracias a un algoritmo que divide de la manera más equitativa posible el trabajo, para evitar los así denominados cuellos de botella. De forma sencilla, es la manera en que las peticiones de internet son distribuidas sobre una fila de servidores. (Tanenbaum, 2009)

La carga de trabajo se define operacionalmente en términos de factores tales como los requisitos de la tarea o el esfuerzo que debe ser invertido para realizarla. Sin embargo, no es aconsejable tener en cuenta sólo un aspecto de la carga de trabajo, ya que estos factores se relacionan entre sí de manera compleja. (Farmer, Browson y QinetiQ, 2003)

Existen varios algoritmos para realizar el balanceo de carga, a continuación, se citan algunos de los frecuentemente usados.

- Round Robin(RR): Las solicitudes entrantes son distribuidas una a una entre los servidores en línea, de manera que se reparten de forma equitativa. Todos los procesos se tratan con la misma importancia, sin importar la urgencia de la solicitud o de la carga que ocasione en el servidor.

- **Weighted Round-Robin:** Las debilidades del clásico algoritmo Round Robin en clúster heterogéneos de servidores se pueden equilibrar por medio de una distribución Round Robin ponderada, donde las peticiones se entregan dependiendo del peso estático de cada servidor, establecido por el administrador.
- **Least Connection:** Las peticiones se distribuyen en función de las conexiones ya existentes en el correspondiente servidor: el que tiene menos conexiones activas recibe la siguiente petición por parte del balanceador de carga.
- **Weighted Least Connection:** Aquí se tiene en cuenta tanto la cantidad de conexiones activas de un servidor como su capacidad. El balanceador de carga asegura el reparto equitativo de la carga en el clúster. El balanceador asigna la siguiente petición al servidor con una previsible carga menor.
- **Retraso Mínimo Esperado (MED),** algoritmo que posee como principal característica la valoración del parámetro tiempo de espera en el servidor, en el que el valor del retraso esperado se define como el tiempo medio de espera calculado a partir de los esquemas de conectividad de los nodos móviles, es decir, calcula en tiempo real y sobre un histórico parcial, el tiempo que debe esperar un cliente en la cola para ser atendido y selecciona el servidor que tenga el menor valor. (Park et al. ,2012)
- **Retraso Mínimo Esperado Modificado (MEDM),** es un híbrido entre los dos algoritmos MED y RR. Además de calcular el retardo esperado en cada uno de los servidores y determinar cuál es el menor, utiliza la técnica de turno rotativo cuando el menor retardo esperado es el mismo para dos o más servidores, teniendo en cuenta a cuál de estos se le envió la última petición. (Rizo Sandoval, 2015)

Luego de analizado los algoritmos anteriormente vistos se selecciona el MEDM como base para la implementación de la distribución de la carga de trabajo asociada en la aplicación debido a sus características adaptables a las necesidades del negocio. Al ser un híbrido de dos algoritmos, aprovecha las ventajas de ambos. El RR facilita la asignación cuando los parámetros son de igual importancia y asegura la aleatoriedad y no repetición de la selección. Al emplear el MED, se tiene en cuenta el tiempo de espera de las incidencias, garantizando el orden en la asignación. Siempre dando espacio para tener en cuenta otros parámetros de importancia como la disponibilidad de los técnicos, su área de trabajo y prioridad de la incidencia. De esta manera el algoritmo toma la decisión de distribución de la carga de una forma más equilibrada y completa. A continuación, se describe el funcionamiento del algoritmo propuesto.

Inicialmente se capturan los datos necesarios de la incidencia y de los técnicos registrados. Si hay técnicos disponibles, más de una incidencia pendiente con iguales prioridades y tiempos de espera, se aplica el RR, si el área del técnico seleccionado es igual al tipo de incidencia en cuestión, se asigna la incidencia. De no haber más de una incidencia pendiente, se aplica el RR y si el área del técnico

seleccionado es igual al tipo de incidencia en cuestión, se asigna la incidencia. Si hay más de una incidencia pendiente y no tienen igual prioridad, se asigna por el mayor valor de esta, comprobando luego que el área del técnico seleccionado es igual al tipo de incidencia en cuestión. Si hay más de una incidencia pendiente, con iguales prioridades y diferentes tiempos de espera en el servidor se asigna según el mayor valor de este parámetro, comprobando luego que el área del técnico seleccionado es igual al tipo de incidencia en cuestión. Si no hay técnicos disponibles se espera a que se inserte una nueva incidencia o un nuevo técnico para volver a ejecutar el algoritmo. Para su mejor comprensión, seguidamente se muestra el algoritmo en una gráfica.

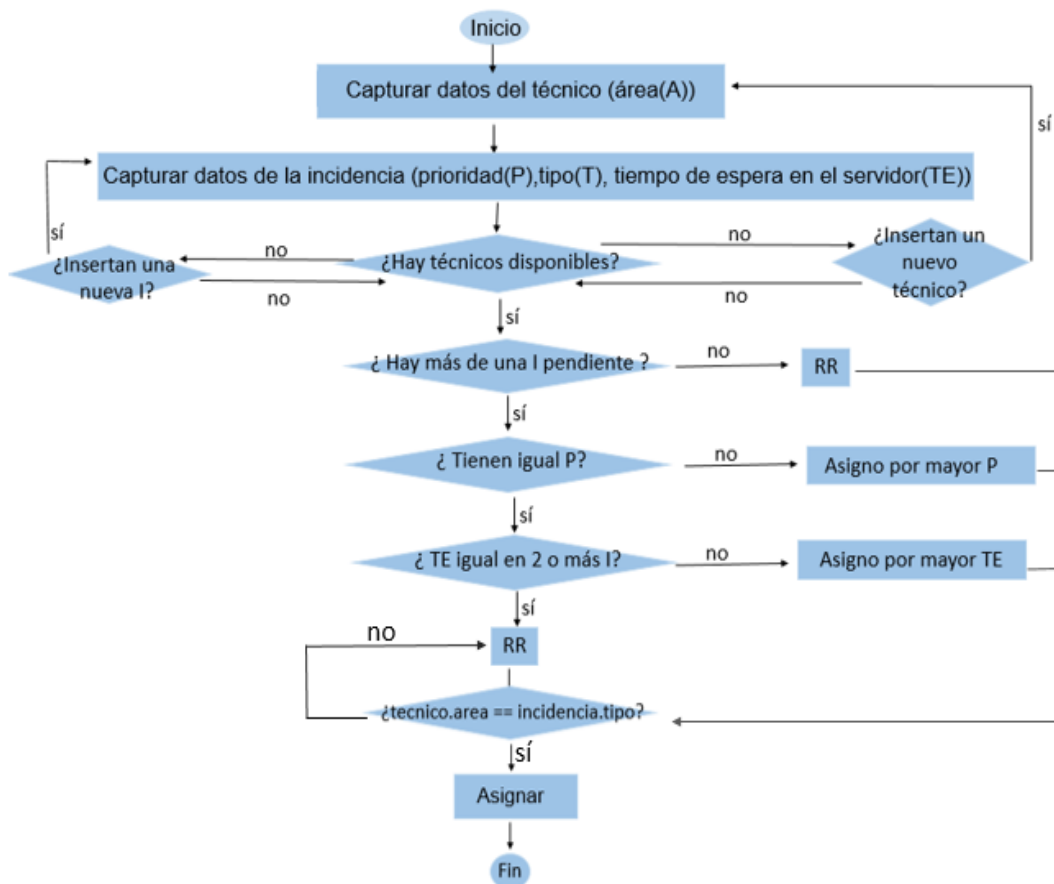


Fig. 1 Algoritmo MEDM

1.5 Herramientas, tecnologías y metodología.

A continuación, se presentan las descripciones de las herramientas, tecnologías y metodología a utilizar para dar solución al problema. Para definir la selección se tomaron en cuenta las necesidades existentes y tendencias actuales.

1.5.1 Metodología de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas, herramientas y soporte documental para el desarrollo de un producto (software). Se basan en una combinación de los modelos de proceso genéricos (cascada, evolutivo e incremental). Son para estructurar, planear y

controlar el proceso de desarrollo del software. Constituyen una guía que define las tareas y actividades que se deben realizar para obtener un software de buena calidad. (Letelier y Penadés, 2006)

Al enfrentar el desarrollo de un software existen varias metodologías por las cuales regirse, elegir la más adecuada es un aspecto de gran importancia. La Estrella de Boehm y Turner es una técnica empleada para esta selección. La misma propone un espacio de 5 dimensiones para determinar qué grado de agilidad-formalismo debe tener un proceso de desarrollo de software; criticidad, personal, tamaño, cultura y dinamismo, se arriba a una mejor opción. (Vázquez, Pedro y Rodríguez [sin fecha]) A continuación se ofrece una breve descripción de estas dimensiones.

- Personal: El por ciento del personal con capacidad, experiencia y posibilidades para enfrentar tareas.
- Criticidad: La criticidad del sistema resultada determinante cuando se habla de proyectos cuya falla puede traer pérdidas de vidas humanas, daño al medio ambiente y grandes pérdidas económicas.
- Cultura: El por ciento del personal con capacidad para enfrentarse a entornos caóticos resulta importante a la hora de adoptar métodos ágiles los cuales se basan fundamentalmente en el talento de las personas.
- Tamaño: La cantidad de personas de un proyecto resulta importante en dependencia de qué método ágil o robusto se vaya a poner en práctica.
- Dinamismo: El por ciento de cambios de requerimientos que pueden ocurrir en un mes tiene gran peso en los métodos ágiles preparados para enfrentar este tipo de situación.

En la realización de la estrella se asignaron los valores de la siguiente forma: en el aspecto de Personal, se decide tomar el valor de 10 en la escala Junior, ya que el personal de desarrollo del software cuenta con poca experiencia de trabajo. En el aspecto de Criticidad se determina el valor, teniendo en cuenta que, ante un posible fallo del sistema, al no ser un software crítico solo ocasionaría pérdidas de utilidades. Al aspecto de Cultura se determinó el valor de 90, debido a que el personal de desarrollo es pequeño, por lo que existe buena comunicación y trabajo en equipo. En el aspecto de Tamaño se escogió el valor 1 ya que es el número de personas con que se cuenta para desarrollar el sistema. Al aspecto Dinamismo se le otorga un alto valor puesto que los requisitos son cambiantes por poca experiencia de trabajo en el tipo de negocio en cuestión. Como el área del interior de la figura creada es cercana al centro se llega a la conclusión que según las características del proyecto se ajusta más una metodología ágil. A continuación, se muestra una imagen con el análisis realizado:

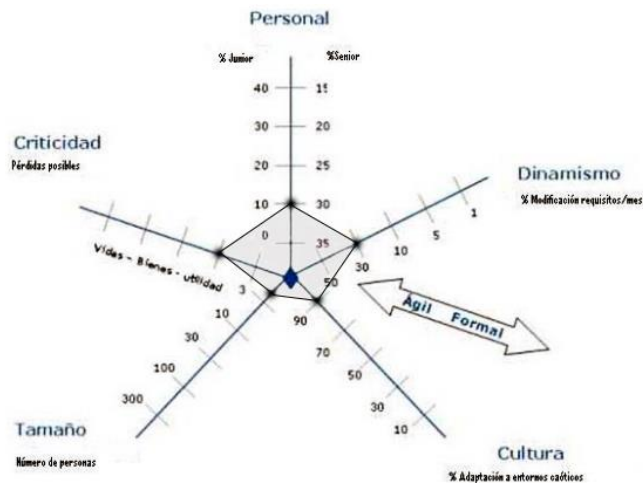


Fig. 2 Estrella de Boehm y Turner

1.5.1.2 Metodologías ágiles

Son orientadas a la interacción con el cliente y el desarrollo incremental del software. Mostrando versiones parcialmente funcionales de la aplicación al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando. Se enmarca más en la capacidad de respuesta ante un cambio realizado que en el seguimiento estricto de un plan.

La metodología seleccionada fue Extreme Programming (XP), es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes y donde existe un alto riesgo técnico. (Letelier y Penadés, 2006)

Se adopta esta metodología debido a:

- El equipo de desarrollo y el usuario final deben trabajar juntos en el transcurso del proyecto.
- Es un proyecto pequeño donde todo el trabajo es desarrollado por un programador.
- El cliente juega un papel imprescindible en el equipo de desarrollo.
- Las funcionalidades tienden a variar frecuentemente durante el proceso de desarrollo.

1.5.2 Herramienta CASE

Visual Paradigm 8.0

Utilizada para el diseño de la base de datos y el diagrama de paquetes para representar las clases del sistema según la arquitectura seleccionada. Visual Paradigm forma parte de una herramienta CASE. (Visual Paradigm, 2018)

A continuación, se muestran las razones de elección:

- Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de las clases.

- Disponibilidad en múltiples plataformas: Microsoft Windows (98, 2000, XP, o Vista) y Linux.
- Permite modelar base de datos y transformación de diagramas de Entidad-Relación en tablas de base de datos.

1.5.3 Lenguajes de programación

Python 2.7.6

Es un lenguaje que permite trabajar más rápido e integrar sistemas de manera más efectiva, es dinámico, orientado a objetos que se puede usar para muchos tipos de desarrollo de software. (Python Software Foundation, 2018) Lenguaje utilizado en el Centro de Telemática perteneciente a la Universidad de Ciencias Informáticas, forma parte de las líneas de desarrollo del mismo.

Entre las características que propiciaron su elección están:

- Un tipo de diccionario ordenado.
- Nuevas funciones de pruebas unitarias que incluyen omisión de prueba, nuevos métodos de afirmación y descubrimiento de prueba.
- Numeración automática de campos en el método `str.format()`.
- Establece literales.
- Comprensión y vistas de diccionarios.
- Nueva sintaxis para anidado con declaraciones.
- El módulo `sysconfig`.

HTML 5

Posee elementos, atributos y comportamientos que facilitan su empleo. Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance. («HTML5» ,2018)

Criterios por los cuales se eligió:

- Semántica: Permite describir con mayor precisión cuál es su contenido.
- Sin conexión y almacenamiento: Permite a las páginas web almacenar datos localmente en el lado del cliente y operar sin conexión de manera más eficiente.
- Rendimiento e Integración: Proporciona una mayor optimización de la velocidad y un mejor uso del hardware.
- Acceso al dispositivo: Proporciona APIs para el uso de varios componentes internos de entrada y salida de dispositivos.

CSS3

Utilizado para definir la presentación (aspecto visual) de un documento HTML o página web. La idea es separar el contenido o información de su presentación visual. (W3.CSS, 2018)

Sus módulos brindan un sinfín de opciones, por las cuales se optó por emplearse, entre ellos:

- Selectores
- Modelo de caja

- Fondos y bordes
- Valores de imagen y contenido reemplazado
- Efectos de texto
- Animaciones
- Diseño de columnas múltiples
- Interfaz de usuario

JavaScript 1.9.1

Lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa. Las capacidades dinámicas de JavaScript por las cuales se eligió incluyen construcción de objetos en tiempo de ejecución, listas variables de parámetros, variables que pueden contener funciones, creación de scripts dinámicos, introspección de objetos y recuperación de código fuente (los programas de JavaScript pueden descompilar el cuerpo de funciones a su código fuente original). («Acerca de JavaScript» 2018)

1.5.4 Servidor de base de datos

PostgreSQL 9.4.1

Sistema de gestión de base de datos relacional orientada a objetos, libre y publicado bajo la licencia BSD. Es un gestor de código abierto avanzado, ofreciendo control de concurrencia multi-versión, soportando casi todas las sintaxis SQL como subconsultas, transacciones y funciones definidas por el usuario, cuenta además con un amplio conjunto de enlaces con lenguajes de programación como C, C++, Java y Python. (The PostgreSQL Global Development Group, 2018)

Entre sus características por las cuales se optó por este gestor se encuentran:

- Poseer una gran escalabilidad. Es capaz de ajustarse al número de Unidades Centrales de Procesamiento (Central Processing Units, CPU por sus siglas en inglés) y a la cantidad de memoria que posee el sistema de forma óptima.
- Tener la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos.
- La simplificación del proceso de administración de licencias de software, que no es necesario cuando se usa software libre.

1.5.5 Framework de desarrollo

Django 1.7

Django es un framework web Python de alto nivel que fomenta un desarrollo rápido y un diseño limpio y pragmático. Se encarga de gran parte de las complicaciones del desarrollo web. Es gratis y de código abierto. (Contributors, 2018) («Django REST framework» 2018)

Ventajas por las cuales se decide su utilización:

- Posee un sistema "middleware" para desarrollar características adicionales; por ejemplo, la distribución principal de Django incluye componentes middleware que proporcionan cacheo, compresión de la salida, normalización de URLs, protección CSRF y soporte de sesiones.
- Soporte de internacionalización, incluyendo traducciones incorporadas de la interfaz de administración.
- Documentación incorporada accesible a través de la aplicación administrativa (incluyendo documentación generada automáticamente de los modelos y las bibliotecas de plantillas añadidas por las aplicaciones).

BackboneJS 2.1.2

Utilizado como framework del lado del cliente. BackboneJS es una librería para el desarrollo de sitios web, principalmente usada para aplicaciones web con bastante interacción con el cliente, donde se hace uso intensivo de JavaScript, Ajax, etc. Permite desarrollar en JavaScript atendiendo a patrones, con una variante del paradigma Modelo-Vista-Controlador (MVC), el mismo que se ha introducido con fuerza en la programación del lado del servidor y que, desde hace menos tiempo, viene empezando a ser costumbre en el desarrollo del lado del cliente. («Qué es BackboneJS» 2018)

Razones de su elección:

- Ayuda a crear estructuras bien definidas para los datos de tu aplicación y facilita la creación de eventos cuando los datos cambian. Esto permite olvidar la revisión de las variables o la propagación de los cambios en la aplicación cuando los datos de dichas estructuras cambian.
- Simplifica el uso de vistas, que te ayudan a pintar o "renderizar" interfaces de usuario en la página.
- Permite el uso de colecciones de modelos, en los que se tiene acceso a diferentes operaciones sobre los mismos, como filtrar, buscar, recibir notificaciones cuando cambien, etc.

Django-Rest-Framework 2.2.1

Utilizado como framework del lado del servidor. Django Rest Framework es una aplicación Django que permite construir proyectos software bajo la arquitectura REST, incluye gran cantidad de código para reutilizar (Views, Resources, etc.) y una interfaz administrativa desde la cual es posible realizar pruebas sobre las operaciones HTTP como lo son: POST y GET. («Django REST framework» 2018)

Razones de su elección:

- El marco Django REST es un conjunto de herramientas potente y flexible para crear API web.
- Políticas de autenticación.
- Amplia documentación.

Xilema base web

Marco de trabajo desarrollado en el Centro de Telemática de la Universidad de Ciencias Informáticas que está constituido por Django como framework base, librerías de JavaScript como son JQuery y

Backbone, además cuenta con las pautas de diseño de la Universidad. Actualmente xilema-base-web es un proyecto Django que sirve de base para los subsistemas: gserver y gadmin.

1.5.6 Entorno Integrado de desarrollo

Para la ejecución del sistema se hizo necesario escoger un entorno integrado de desarrollo (IDE) que proporcionara el uso y la integración de todas las tecnologías y lenguajes antes mencionados, por lo que se decidió utilizar el **PyCharm 2017.1.3**

PyCharm es un IDE utilizado en la programación de computadoras, específicamente para el lenguaje Python. (JetBrains, 2018)

Los criterios por los cuales se seleccionó es que:

- Proporciona análisis de código y un depurador gráfico, un probador de unidades integrado.
- Integra con los sistemas de control de versiones (VCS).
- Admite el desarrollo web con Django.
- Es multiplataforma, con versiones de Windows, macOS y Linux.

1.6 Conclusiones parciales

El análisis de las soluciones nacionales e internacionales dedicadas a la gestión de incidencias permitió obtener características útiles y comunes como son el poseer asignación automática de incidencias, la generación dinámica de estadísticas, la visualización de incidencias por diferentes variables, la importancia de la generalización del sistema, entre otros aspectos de valor. Permitted concluir la necesidad de la creación de un nuevo sistema ya que ninguno de los sistemas analizados se adaptaba a los requisitos especificados. Se determinó utilizar el método de balanceo de carga MEDM ya que para el sistema de distribución como el que se pretende desarrollar este resulta idóneo por sus características de ser híbrido, tomando las ventajas del RR y MED para fusionarlas y adaptarlas a las necesidades del negocio. Se seleccionaron las herramientas y la metodología de desarrollo que cumplen con las características necesarias para dar solución a dicha situación problemática.

Capítulo 2: Exploración, planificación y diseño del sistema.

En este capítulo se abordan los temas relacionados con la fase de exploración, planificación y diseño de la metodología de desarrollo XP, donde se describe las características del sistema a desarrollar y su arquitectura. Se confecciona las HU que proporcionan un mayor entendimiento y comprensión del sistema. Se tratan los principales artefactos generados y la planificación del tiempo y el esfuerzo de las fases posteriores.

2.1 Propuesta del sistema

Se propone el desarrollo de una aplicación web que permita gestionar las incidencias reportadas por los clientes de empresas que brindan servicios y la distribución de la fuerza de trabajo asociada a este. Con el objetivo de lograr una gestión más completa de incidencias. El sistema brinda la posibilidad de gestionar los usuarios y los permisos de estos, según el rol asignado es el acceso a los datos que posea. La persona con rol de administrador, debe registrar en el sistema los datos de configuración, necesarios para insertar una incidencia, como la información referente a los técnicos destinados a solucionarla y su área de trabajo. De igual forma los referentes a la propia incidencia, como los estados, prioridades, tipos y seleccionar la opción de distribución de carga de trabajo que se aplica. Tiene acceso a toda la información registrada en el sistema.

El operador, persona encargada de atender las llamadas telefónicas, registra la incidencia reportada por un cliente en el sistema, para luego asignarla al técnico según el mecanismo de distribución seleccionado previamente por el administrador. Tiene acceso solo a gestionar las incidencias y asignarlas a los técnicos, esta última acción, si la opción de distribución es manual.

Internamente el sistema está en constante relación con los subsistemas que la conforman y que completan la gestión. A continuación, se describe el proceso en una imagen.



Fig. 3 Propuesta del sistema

El sistema está conformado por tres subsistemas principales los cuales se describen a continuación.

- Subsistema de gestión de incidencias:

El usuario que interactúa con él debe de estar autenticado y tener los permisos necesarios. Luego, puede realizar diversas funciones relacionadas con la gestión de incidencias. Adicionar, modificar, eliminar, listar, detallar y exportar a hoja de cálculo las incidencias registradas. De igual forma con los datos referentes a la propia incidencia, como los estados, prioridades y tipos. La información referida a los técnicos destinados a solucionarla y su área de trabajo.

- Subsistema de distribución de carga de trabajo:

Este subsistema cuenta con el algoritmo MEDM explicado anteriormente que tiene en cuenta para la asignación de tareas, los técnicos que están disponibles, área de trabajo, los tipos de incidencias, la prioridad que tiene asignada cada una y el tiempo de espera en el servidor. Teniendo en cuenta estos parámetros, el algoritmo asigna las tareas a los técnicos logrando un balanceo de cargas, de forma tal que no queden técnicos muy sobrecargados y otros muy libres de tareas.

Brinda la posibilidad de elegir de qué manera va a ser balanceada la carga: automática o manual. La asignación automática se encarga de realizar todo el proceso invisible a los ojos del operador aplicando el algoritmo antes explicado. La manual muestra la lista de técnicos registrados en el sistema, para que el operador manualmente seleccione el técnico deseado. Haciendo más flexible el proceso de asignación de incidencias.

- Subsistema de Servicios REST:

La herramienta es capaz de publicar como recursos (entidad que representa un concepto de negocio que puede ser accedido públicamente) en internet los datos del sistema para que estos puedan ser consultados mediante aplicaciones web externas como Sistema de Voz Interactiva (IVR). Cada recurso posee un identificador único y un estado interno que no puede ser accedido externamente. Los estados son representaciones de los datos, para que puedan ser consultados desde el exterior del sistema usando el JSON (JavaScript Object Notation). Transfiere los datos por peticiones HTTP hacia los clientes.

2.2 Características no funcionales del sistema

Las características no funcionales de una aplicación son las cualidades o propiedades que el sistema debe poseer, estas hacen al software atractivo, usable, rápido o confiable. Para la realización de la aplicación propuesta se han identificado las siguientes.

Usabilidad

El sistema debe ser sencillo de usar, para que el operador entienda e interactúe con este con facilidad. Todas sus interfaces deben permitir la traducción a los idiomas: español, inglés, francés y portugués.

Interfaz

El diseño gráfico debe tener colores suaves, que se correspondan con los estandarizados por la Universidad de Ciencias Informáticas para esta línea de productos.

Seguridad

- **Confiabilidad:** La información manejada por el sistema debe estar protegida de acceso no autorizado por lo cual se establece un nivel de acceso a la aplicación mediante usuario, contraseña y asignación de permisos.
- **Integridad:** La información manejada por el sistema debe ser objeto de una cuidadosa protección contra la corrupción y estados de inconsistencia.
- **Disponibilidad:** significa tener la seguridad de acceder en forma confiable y oportuna a los servicios de datos para usuarios autorizados.

Hardware

Tabla 2 Especificaciones de hardware en el servidor

Servidor	Especificaciones
Web	Procesador: 3 núcleos RAM: 2 GB Disco duro: 10 GB UPS: 1 Tarjeta de Red: 1
Base de datos	Procesador: 3.00 GHZ RAM: 1GB Disco duro: 160 GB UPS: 1 Tarjeta de Red: 1

Tabla 3 Especificaciones de hardware en el cliente

Cliente	Especificaciones
PC	Procesador: 1.00 GHZ o mayor RAM: 512 MB o superior Tarjeta de Red: 1

Software

Tabla 4 Especificaciones de software

Especificaciones
Sistema Operativo: Linux (Debian, Ubuntu) o Windows Navegador web: Mozilla Firefox v 47.0 o superior, Chrome v 56.0 o superior Instalado Python 2.7 y Django v1.7 como marco de trabajo Gestor de base de datos: PostgreSQL 9.4

2.3 Fase de Exploración

La metodología de desarrollo XP comienza con la fase de exploración. Durante esta se crean las Historias de Usuarios (HU).

2.3.1 Involucrados en el sistema

El personal que interactúa con el sistema directa o indirectamente son los involucrados. A continuación, se describen los mismos.

Tabla 5 Involucrados en el sistema

Involucrados	Descripción
Operador	Persona encargada de insertar los datos de la incidencia en el sistema y distribuir la carga de trabajo si la opción de distribución es manual.
Administrador	Personal encargado de gestionar los datos de configuración (tipo, estado y prioridad de la incidencia; técnico, área de trabajo de este y la opción de distribución).

En esta fase se identificaron las siguientes funcionalidades:

- Gestionar incidencias.
- Gestionar tipo de incidencias.
- Gestionar estado de incidencia.
- Gestionar técnico.
- Gestionar prioridad.
- Gestionar área.
- Gestionar distribución de la carga de trabajo.
- Exportar datos de incidencia.
- Detalles de incidencia.
- Detalles de técnico.
- Cambiar ping de técnico.

2.3.2 Historias de Usuarios (HU).

En XP la gestión de requisitos es simple, el cliente escribe y prioriza las historias de usuario que expresan las necesidades del sistema. Los programadores estiman el esfuerzo asociado y las dependencias entre ellas. Para planificar el trabajo desde el punto de vista técnico las historias de usuario son divididas en tareas para las cuales también se realiza una estimación. Teniendo en cuenta el esfuerzo asociado a las historias de usuario y las prioridades del cliente se define una versión que sea de valor para el cliente y que tenga una duración de unos pocos meses. (Somerville, 2006)

Las HU se representan mediante tablas, las cuales se dividen en varias secciones denominadas:

- Número: representa el número, incremental en el tiempo, de la HU que se describe.
- Nombre: identifica la HU que se describe entre los desarrolladores y el cliente.

- Usuario: rol de la persona que ejecuta la funcionalidad.
- Prioridad en negocio: se otorga una prioridad (Alta, Media, Baja) a las HU de acuerdo a la necesidad de desarrollo.
- Riesgo en Desarrollo: se otorga una medida (Alto, Medio, Bajo) a la ocurrencia de errores en el proceso de desarrollo de la HU.
- Iteración asignada: número de la iteración donde va a desarrollarse la HU.
- Programador responsable: encargado de programar la HU.
- Puntos Estimados: tiempo estimado en semanas (de lunes a viernes) que demora el desarrollo de la HU.
- Descripción: breve descripción de la HU.
- Observaciones: señalamiento o advertencia del sistema.
- Prototipo de interfaz: imagen de cada interfaz relacionada con la HU.

Durante esta fase de identificaron 36 HU definidas por el cliente en conjunto con el equipo de desarrollo, a continuación, se muestran las HU correspondientes a las funcionalidades Insertar incidencia, Insertar Técnico, Seleccionar opción de distribución y Distribuir carga. (el resto ver en [Anexos](#)):

Tabla 6 HU 1 Insertar incidencia

Historia de usuario	
Número: 1	Usuario: Operador
Título: Insertar incidencia	
Prioridad en negocio: Alta	Iteración:1
Riesgo de desarrollo: Alto	Puntos Estimados: 0.2
Programador responsable: Sheila Tiá Ubeda	
Descripción: El operador adiciona las incidencias al sistema llenando los campos requeridos. (Tipo, Estado, Servicio afectado, Descripción, Localización, Fecha de ocurrencia, Prioridad, introducir datos del solicitante; Nombre y Apellidos, Dirección particular y Teléfono de contacto).	
Observaciones: El operador tiene que estar autenticado en el sistema.	

Interfaz:

The image shows a web form titled "Adicionar" with the following fields and controls:

- Descripción: Text input field.
- Estado: Dropdown menu with "--Seleccione--".
- Tipo: Dropdown menu with "--Seleccione--".
- Servicio afectado: Text input field.
- Localización: Text input field.
- Fecha de ocurrencia: Date input field.
- Nombre del solicitante: Text input field.
- Apellidos del solicitante: Text input field.
- Dirección particular del solicitante: Text input field.
- Teléfono de contacto del solicitante: Text input field.
- Prioridad: Dropdown menu with "--Seleccione--".
- Buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

Tabla 7 HU 21 Insertar Técnico

Historia de usuario	
Número: 21	Usuario: Administrador
Título: Insertar técnico	
Prioridad en negocio: Alta	Iteración: 1
Riesgo de desarrollo: Alto	Puntos Estimados: 0.2
Programador responsable: Sheila Tiá Ubeda	
Descripción: El administrador del sistema adiciona el técnico en el sistema llenando los campos requeridos. (Nombre, Apellidos, Área, Teléfono de contacto, Ping)	
Observaciones: El administrador tiene que estar autenticado en el sistema.	
Interfaz:	

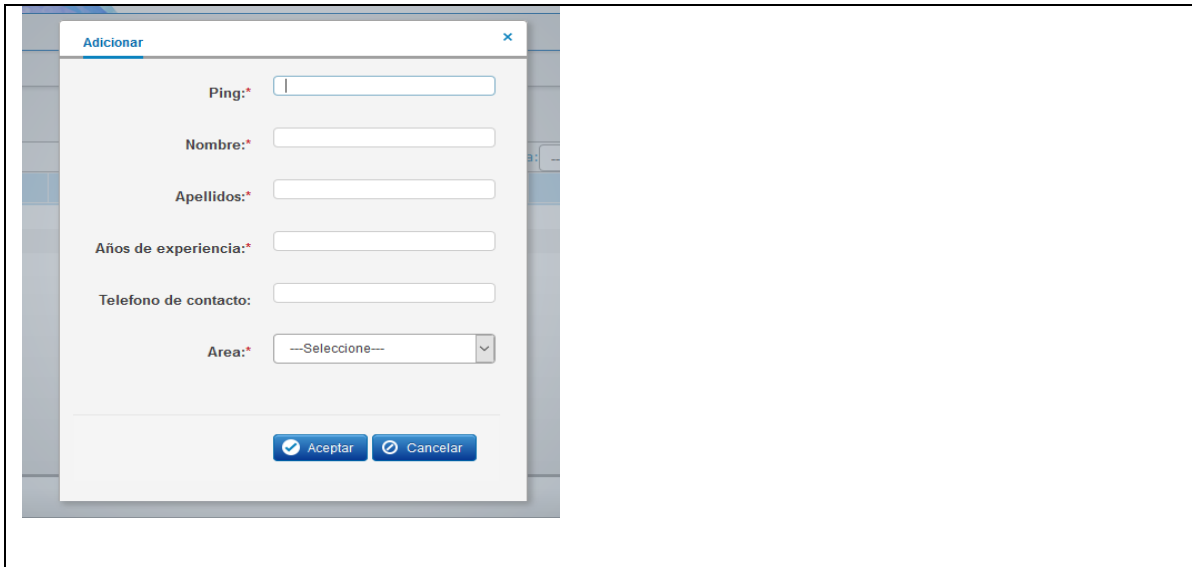
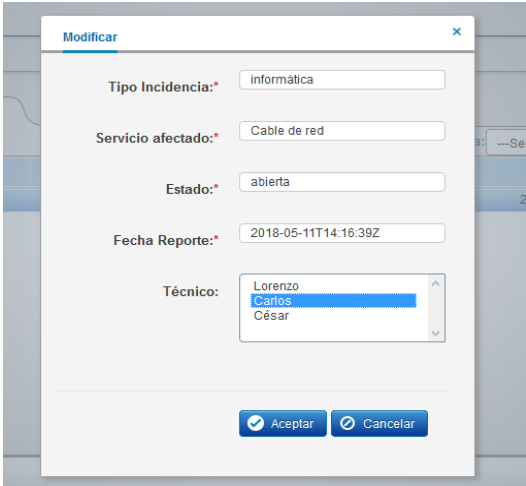


Tabla 8 HU 31 Seleccionar opción de distribución

Historia de usuario	
Número: 31	Usuario: Administrador
Título: Seleccionar opción de distribución	
Prioridad en negocio: Alta	Iteración:2
Riesgo de desarrollo: Alto	Puntos Estimados: 1
Programador responsable: Sheila Tiá Ubeda	
Descripción: El administrador del sistema selecciona la opción de balanceo de carga: "Automática" o "Manual".	
Observaciones: El administrador tiene que estar autenticado en el sistema.	
Interfaz:	

Tabla 9 HU 32 Distribuir carga de trabajo

Historia de usuario	
Número: 32	Usuario: Operador

Título: Distribuir carga de trabajo	
Prioridad en negocio: Alta	Iteración:2
Riesgo de desarrollo: Alto	Puntos Estimados: 1
Programador responsable: Sheila Tiá Ubeda	
<p>Descripción: El operador selecciona el menú “Distribuir carga” y el sistema lista las incidencias registradas, selecciona una de estas y oprime la opción “Asignar” y se muestra una ventana mostrando breves datos de la incidencia seleccionada y el listado de técnicos registrados en la base de datos.</p> <p>En la asignación “Automática” se aplica el algoritmo de distribución de la carga de trabajo.</p>	
Observaciones: El operador tiene que estar autenticado en el sistema. Haber seleccionado la opción “Manual” en la HU 31.	
Interfaz:	
	

2.4 Fase de Planificación

Durante la fase de planificación se realiza una estimación del esfuerzo que cuesta implementar cada HU. Se toman acuerdo sobre el contenido de la primera entrega y se establece un cronograma en conjunto con el cliente. Además, se deben incluir varias iteraciones para lograr una entrega, siendo así el plan de entregas el resultado de esta fase.

2.4.1 Estimación de esfuerzo

Las estimaciones de esfuerzo en semanas de cada historia de usuario identificada se muestran a continuación, considerando la estimación de 0 a 0.7 en días y de 1 en adelante en semanas.

Tabla 10 Estimación de esfuerzo

No	HU	Puntos de estimación
1	Insertar incidencia	0.2
2	Modificar incidencia	0.2
3	Eliminar incidencia	0.1

4	Listar incidencia	0.1
5	Buscar incidencia	0.1
6	Insertar tipo de incidencia	0.2
7	Modificar tipo de incidencia	0.2
8	Eliminar tipo de incidencia	0.1
9	Listar tipo de incidencia	0.1
10	Buscar tipo de incidencia	0.1
11	Insertar estado de incidencias	0.2
12	Modificar estado de incidencias	0.2
13	Eliminar estado de incidencias	0.1
14	Listar estados de incidencias	0.1
15	Buscar estados de incidencias	0.1
16	Insertar prioridad	0.2
17	Modificar prioridad	0.2
18	Eliminar prioridad	0.1
19	Listar prioridad	0.1
20	Buscar prioridad	0.1
21	Insertar técnico	0.2
22	Modificar técnico	0.2
23	Eliminar técnico	0.1
24	Listar técnico	0.1
25	Buscar técnico	0.1
26	Insertar área	0.2
27	Modificar área	0.2
28	Eliminar área	0.1
29	Listar área	0.1
30	Buscar área	0.1
31	Seleccionar opción de distribución	1
32	Distribuir de carga de trabajo	1
33	Exportar datos de incidencia	0.3
34	Detalles de incidencia	0.1
35	Detalles de técnico	0.1
36	Cambiar ping de técnico	0.2
Total		7 semanas

2.4.2 Plan de iteraciones

Después de ser identificadas las HU y estimado el esfuerzo dedicado a la realización de cada una de estas, se procede a la realización de la planificación de la etapa de implementación. El orden de las HU y de las iteraciones va a estar sujetos al orden lógico en que se deben realizar las acciones, aunque en

la medida de lo posible se respeta la prioridad en el negocio. A continuación, se describen las 3 iteraciones establecidas:

- **Iteración 1:** en esta iteración se realizan las HU con prioridad para el negocio alta, las HU correspondientes a los “Adicionar”, “Modificar” y “Eliminar” de cada gestionar, excepto el “Gestionar distribución”. Quedando así culminada la primera versión del sistema con un 83% de la aplicación terminada.
- **Iteración 2:** en esta se realizan el segundo bloque de las HU con prioridad para el negocio alta, las HU “Seleccionar opción de distribución” y “Distribuir de carga de trabajo” teniendo así una versión más completa del sistema, con un 88% de la aplicación terminada.
- **Iteración 3:** se realizan las HU con prioridad para el negocio media y baja, las HU correspondientes a los “Listar” y “Buscar” de cada gestionar, “Exportar datos de incidencia”, “Detalles de incidencia y técnico” y “Cambiar ping de técnico”. Teniendo la aplicación terminada por completo.

2.4.2.1 Plan de duración de las iteraciones

A continuación, se muestra una tabla con la duración en semanas de cada iteración, así como el orden de implementación de cada HU:

Tabla 11 Plan de iteraciones

Iteración	HU	Duración
1	Insertar incidencia	3 semanas
	Modificar incidencia	
	Eliminar incidencia	
	Insertar tipo de incidencia	
	Modificar tipo de incidencia	
	Eliminar tipo de incidencia	
	Insertar estado de incidencias	
	Modificar estado de incidencias	
	Eliminar estado de incidencias	
	Insertar prioridad	
	Modificar prioridad	
	Eliminar prioridad	
	Insertar técnico	
	Modificar técnico	
	Eliminar técnico	
Insertar área		

	Modificar área	
	Eliminar área	
2	Seleccionar opción de distribución	2 semanas
	Distribuir de carga de trabajo	
3	Listar incidencias	2 semanas
	Listar tipos de incidencias	
	Listar estados de incidencia	
	Listar técnicos	
	Listar prioridad	
	Listar áreas	
	Buscar incidencia	
	Buscar tipo de incidencia	
	Buscar estado de incidencia	
	Buscar prioridad	
	Buscar técnico	
	Buscar área	
	Exportar datos de incidencia	
	Detalles de incidencia	
	Detalles de técnico	
	Cambiar ping de técnico	

2.4.3 Plan de entrega

El plan de entrega detalla la fecha fin de cada iteración para la fase de implementación, la cual comienza el 1ro de marzo de 2018.

Tabla 12 Plan de entrega

Iteración	Fecha de entrega
1	22 de marzo de 2018
2	5 de abril de 2018
3	19 de abril de 2018

2.5 Fase de Diseño

XP propone la fase de diseño utilizada para la implementación del sistema que se propone siguiendo el principio de la elaboración de diseños sencillos para lograr un mejor entendimiento en la fase de desarrollo, se generan los diferentes artefactos y se procede al diseño de la base de datos.

2.5.1 Arquitectura

La arquitectura es una representación que permite analizar la efectividad del diseño para cumplir los requerimientos establecidos, considerar alternativas arquitectónicas en una etapa en la que hacer cambios al diseño todavía es relativamente fácil y reducir los riesgos asociados con la construcción del software. (Pressman y Ph, 2010)

2.5.1.1 Arquitectura Cliente-Servidor

En esta arquitectura la computadora de cada uno de los usuarios, llamada cliente, produce una demanda de información a cualquiera de las computadoras que proporcionan información, conocidas como servidores, estos últimos responden a la demanda del cliente que la produjo. Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla como según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura. (Camacho, Cardeso y NuñezErikaz, 2004)

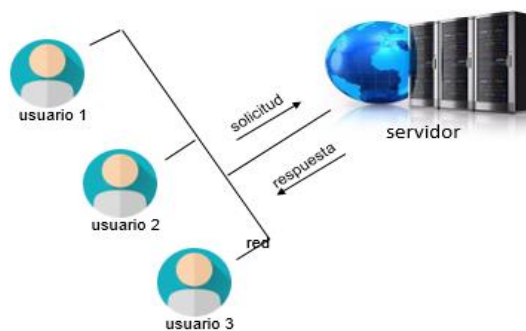


Fig. 4 Arquitectura Cliente-Servidor

Se decide optar por esta arquitectura debido a que favorece el uso de interfaces gráficas interactivas, teniendo una mayor interacción con el usuario. La estructura modular facilita la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.

2.5.2 Patrón arquitectónico

Los patrones arquitectónicos o patrones de arquitectura, también llamados arquetipos ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados.

En el patrón Modelo-Vista-Controlador(MVC) el "Modelo" hace referencia al acceso a la capa de datos, la "Vista" se refiere a la parte del sistema que selecciona qué mostrar y cómo mostrarlo y el "Controlador" implica la parte del sistema que decide qué vista usar, dependiendo de la entrada del usuario, accediendo al modelo si es necesario. («LibrosWeb» 2006)

Django sigue el patrón MVC tan al pie de la letra que puede ser llamado un framework MVC. Someramente, la M, V y C se separan en Django de la siguiente manera:

- M, la porción de acceso a la base de datos, es manejada por la capa de la base de datos de Django.
- V, la porción que selecciona qué datos mostrar y cómo mostrarlos, es manejada por la vista y las plantillas.
- C, la porción que delega a la vista dependiendo de la entrada del usuario, es manejada por el framework mismo siguiendo el URLconf y llamando a la función apropiada de Python para la URL obtenida.

Debido a que la "C" es manejada por el mismo framework y la parte más importante se produce en los modelos, las plantillas y las vistas, Django es conocido como un Framework MTV. En el patrón de diseño MTV:

M significa "Model" (Modelo), la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene y las relaciones entre los datos.

- T significa "Template" (Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web u otro tipo de documento.
- V significa "View" (Vista), la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: como un puente entre el modelo y las plantillas.

Debido a que el framework de desarrollo empleado es Django, se opta por el patrón arquitectónico MTV. Se muestra a continuación la estructura del sistema según la arquitectura seleccionada descrita mediante sus diagramas de clases.

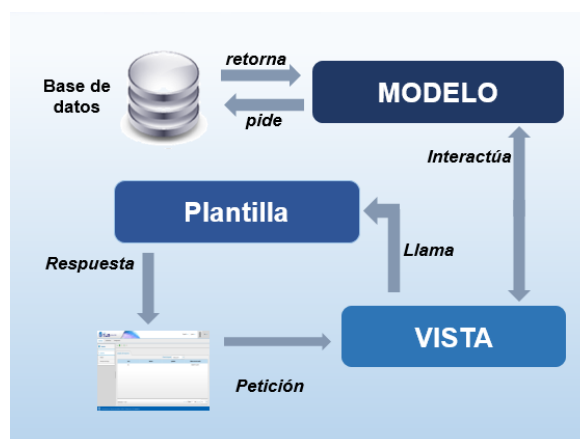


Fig. 5 Patrón arquitectónico MTV

A continuación, se presentan los diagramas de paquetes para englobar las clases que describen la constitución de las capas de la arquitectura MTV:

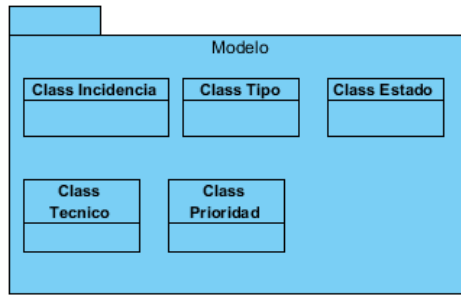


Fig. 6 Clases del Modelo

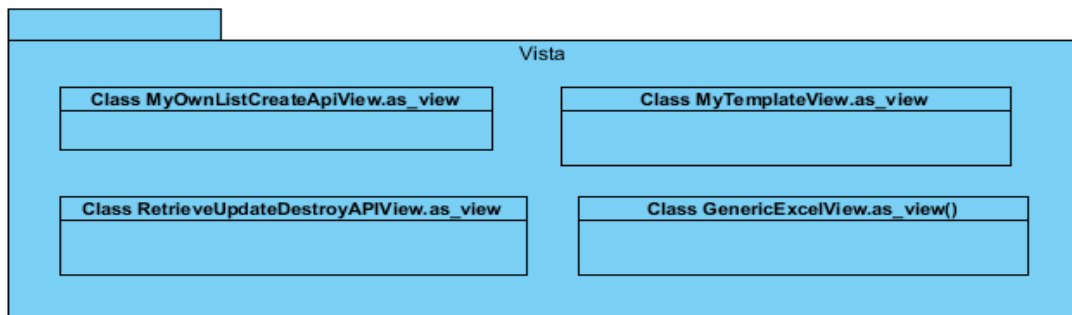


Fig. 7 Clases de la Vista

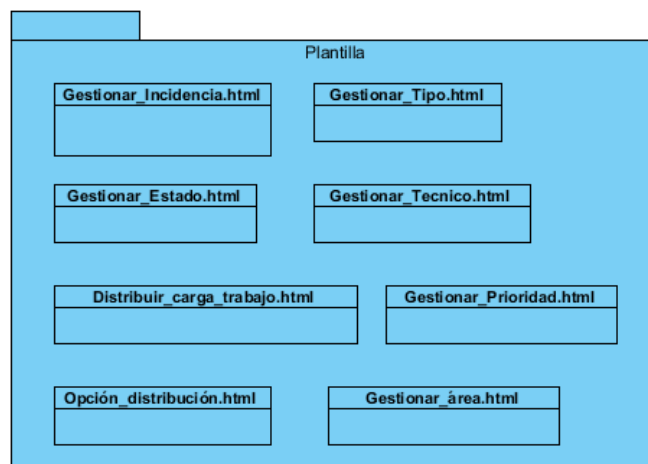


Fig. 8 Clases de la Plantilla

2.5.3 Patrones de diseño

Un patrón de diseño constituye un esquema para refinar subsistemas o componentes. Es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Un patrón de diseño identifica: clases, instancias, roles, colaboraciones y la distribución de responsabilidades, además de que ayuda a construir clases y a estructurar sistemas de clases. Son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. («Academia» 2018)

GRAPS (General Responsibility Assignment Software Patterns, Patrones Generales de Software de Asignación de Responsabilidades).

- **Bajo Acoplamiento:** Consiste en crear pocas dependencias entre las clases con el objetivo que un cambio en alguna de ellas no afecte al resto o dicha afectación tenga una mínima repercusión en el resto de las clases potenciando la reutilización y disminuyendo la dependencia, esto se ve claramente cuando se aplica el estilo MVT, de esta manera se garantiza una dependencia mínima entre las capas Vista y Modelo.
- **Alta Cohesión:** Consiste en que la información que almacena una clase debe de ser coherente y estar en la mayor medida de lo posible relacionada con la clase de modo que cada elemento de nuestro diseño debe realizar una labor única dentro del sistema. Reflejado en todas las clases garantizando que estas posean solo la información pertinente a ellas.
- **Controlador:** está representado por una clase a la cual se le asigna la responsabilidad de las operaciones del sistema. Este patrón se refleja en la clase: Incidencia, encargada de controlar las acciones que realiza el usuario con las interfaces de la aplicación y dar respuesta a las peticiones realizadas.

GoF (Gang of Four)

- **Front Controller:** patrón usado por el framework Django. Reflejado en el fichero de url.py, recibe las peticiones desde los templates y hace las vistas correspondientes.

2.5.4 Tarjetas CRC

Las tarjetas CRC (Clase-Responsabilidad-Colaborador) es una técnica para la representación de sistemas Orientados a Objetos. Son usadas para que los diseñadores piensen en términos abstractos sobre asignación de responsabilidades y de las colaboraciones.

Tabla 13 Tarjeta CRC Técnico

Técnico	
Responsabilidad	Colaborador
Adiciona un técnico	Área
Modifica un técnico	Incidencia
Elimina un técnico	
Detalla información de un técnico	
Cambia el ping de un técnico	
Guarda los datos pertinentes de un técnico	

Tabla 14 Tarjeta CRC Incidencia

Incidencia	
Responsabilidad	Colaborador
Adiciona una incidencia	Técnico
Modifica una incidencia	Prioridad
Elimina una incidencia	Estado
Detalla información de una incidencia	Tipo

Exporta las incidencias	User
Guarda los datos pertinentes de una incidencia	Opción_distribución Área

2.5.5 Diseño de la base de datos

La construcción de la base de datos es una de las funciones principales en la fase de diseño de un sistema, ya que se exponen los datos necesarios para el correcto desempeño del mismo.

2.5.5.1 Modelo lógico de la base de datos

A continuación, se muestra el modelo lógico de la aplicación:

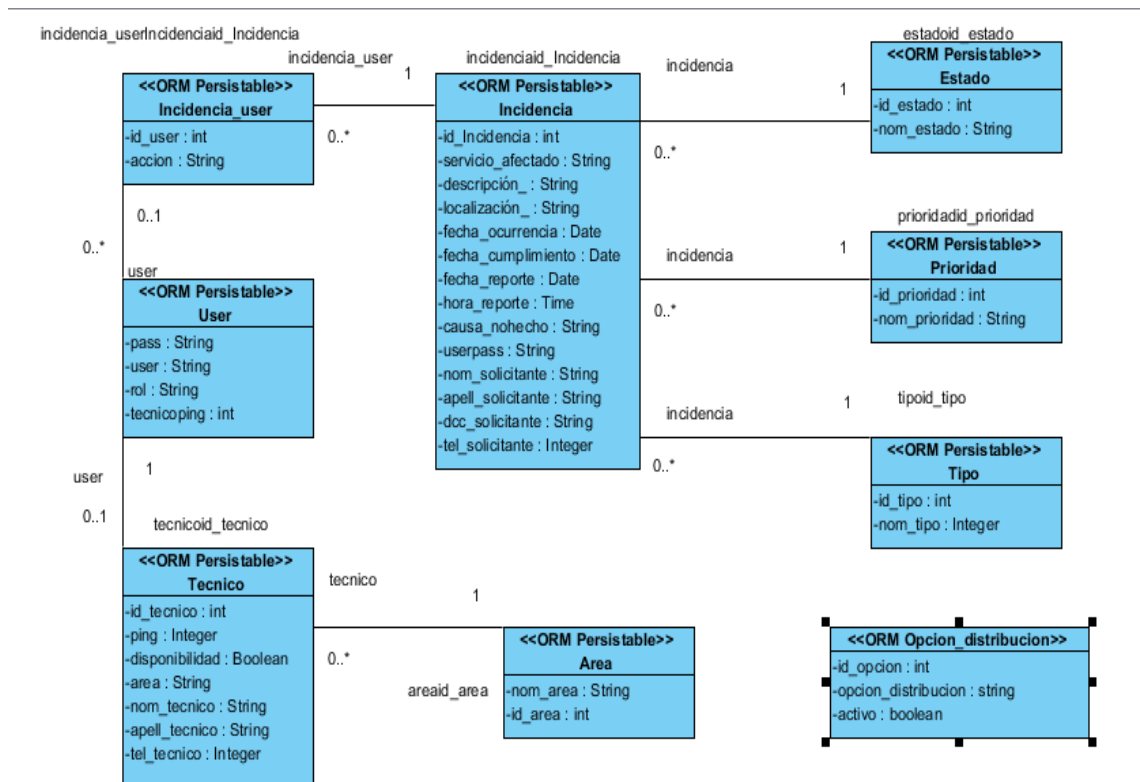


Fig. 9 Modelo lógico de la base de datos

2.5.5.2 Modelo físico de la base de datos

A continuación, se muestra el DER (Entity Relationship Diagrams, Diagrama Entidad Relación) diseñado para la aplicación:

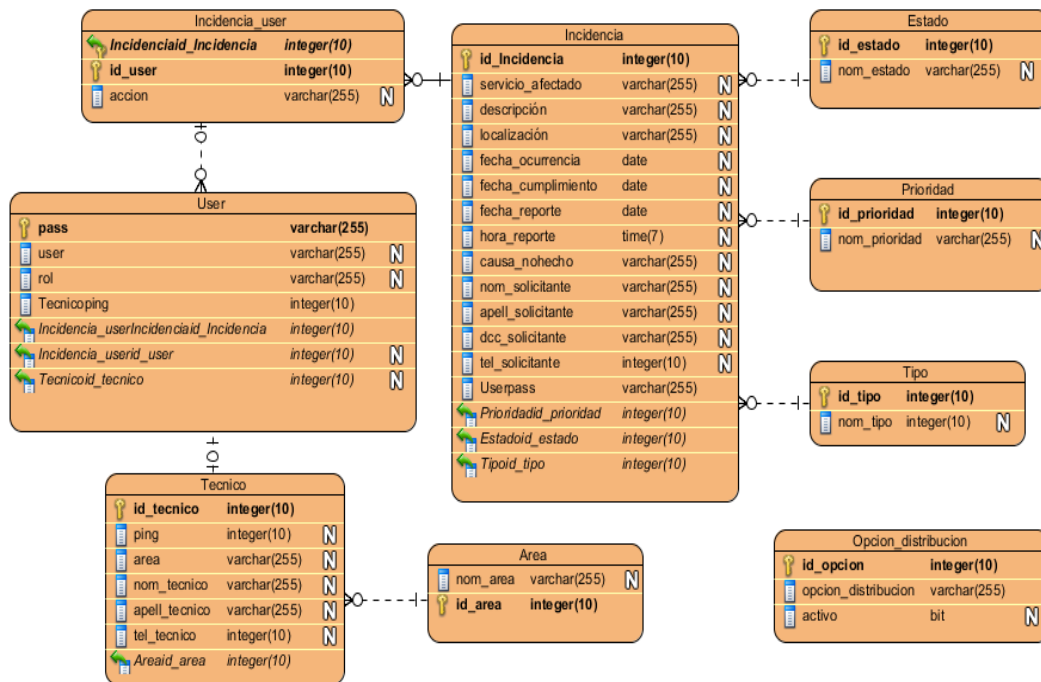


Fig. 10 Modelo físico de la base de datos

2.6 Conclusiones parciales

En este capítulo se describieron las HU que permite guiar el proceso de desarrollo a través de la descripción detallada de las funcionalidades, además de realizar una estimación de tiempo fiable para ajustarse a los planes de entrega y cumplir con las demandas del cliente en tiempo y forma. Se seleccionó la arquitectura MTV para analizar la efectividad del diseño para cumplir los requerimientos establecidos. Se identificaron los patrones de diseño empleados para solucionar problemas de arquitectura de software en ingeniería de software que puedan surgir. Se elaboraron las tarjetas CRC para esclarecer la asignación de responsabilidades de las clases y las colaboraciones. Se realizó el diseño lógico y físico de la base de datos para exponer los datos necesarios para el correcto desempeño del sistema.

Capítulo 3: Implementación y pruebas del Sistema para la gestión de incidencias y distribución de la fuerza de trabajo asociada

En el presente capítulo se muestra el desarrollo de las fases de implementación y pruebas que propone la metodología XP. Se describen los principales artefactos a generar, como son: las tareas de ingeniería por cada HU identificada y las pruebas empleadas para validar la solución propuesta.

3.1 Implementación

Como parte de la metodología ágil escogida durante el inicio de cada iteración se revisa el plan de iteraciones y se expresan las tareas de programación, donde a cada una de ellas son asignadas a los programadores responsables. Como parte de la planificación realizada en el capítulo anterior se detallan a continuación las iteraciones y las historias de usuario implementadas en cada iteración.

3.1.1 Tareas de ingeniería (TI)

Una historia de usuario se descompone en varias tareas de ingeniería, las cuales describen las actividades que se realizan en cada una, así mismo las tareas de ingeniería se vinculan más al desarrollador, ya que permite tener un acercamiento con el código. (Meléndez Valladarez, Elizabeth Gaitan y Pérez Reyes, 2016)

Tabla 15 Tareas de Ingeniería

Iteración	HU	Tareas de ingeniería
1	Insertar incidencia	TI 1: Insertar incidencia
	Modificar incidencia	TI 2: Modificar incidencia
	Eliminar incidencia	TI 3: Eliminar incidencia
	Insertar tipo de incidencia	TI 4: Insertar tipo de incidencia
	Modificar tipo de incidencia	TI 5: Modificar tipo de incidencia
	Eliminar tipo de incidencia	TI 6: Eliminar tipo de incidencia
	Insertar estado de incidencias	TI 7: Insertar estado de incidencias
	Modificar estado de incidencias	TI 8: Modificar estado de incidencias
	Eliminar estado de incidencias	TI 9: Eliminar estado de incidencias
	Insertar prioridad	TI 10: Insertar prioridad
	Modificar prioridad	TI 11: Modificar prioridad
	Eliminar prioridad	TI 12: Eliminar prioridad
	Insertar técnico	TI 13: Insertar técnico
	Modificar técnico	TI 14: Modificar técnico
	Eliminar técnico	TI 15: Eliminar técnico
	Insertar área	TI 16: Insertar área
	Modificar área	TI 17: Modificar área
	Eliminar área	TI 18: Eliminar área
2	Seleccionar opción de distribución	TI 19: Seleccionar opción de distribución

	Distribuir carga de trabajo	TI 20: Distribuir carga de trabajo (Automática)
		TI 21: Distribuir carga de trabajo (Manual)
3	Listar incidencias	TI 22: Listar incidencias
	Listar tipos de incidencias	TI 23: Listar tipos de incidencias
	Listar estados de incidencia	TI 24: Listar estados de incidencia
	Listar técnicos	TI 25: Listar técnicos
	Listar prioridad	TI 26: Listar prioridad
	Listar área	TI 27: Listar área
	Buscar incidencia	TI 28: Buscar incidencia
	Buscar tipo de incidencia	TI 29: Buscar tipo de incidencia
	Buscar estado de incidencia	TI 30: Buscar estado de incidencia
	Buscar técnico	TI 31: Buscar técnico
	Buscar prioridad	TI 32: Buscar prioridad
	Buscar área	TI 33: Buscar área
	Exportar datos de incidencia	TI 34: Exportar datos de incidencia
	Detalles de incidencia	TI 35: Detalles de incidencia
Detalles de técnico	TI 36: Detalles de técnico	
Cambiar ping de técnico	TI 37: Cambiar ping de técnico	

Seguidamente , se muestra las TI referentes a las HU Insertar incidencia, Insertar Técnico, Seleccionar opción de distribución y Distribuir carga (el resto ver en [Anexos](#))

Tabla 16 Tarea de Ingeniería HU Insertar incidencia

Tarea de ingeniería	
Número de tarea: 1	Número de HU: 1
Nombre de la tarea: Insertar incidencia	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Programador responsable: Sheila Tía Ubeda	
Descripción: El operador, en el menú "Gestionar Incidencias" selecciona la opción "Insertar" y el sistema le muestra una interfaz donde debe llenar los campos necesarios (Tipo, Estado, Servicio afectado, Inmueble, Descripción, Localización, Fecha de ocurrencia, Prioridad, Nombre del solicitante, Apellidos del solicitante, Teléfono de contacto, Dirección particular). Luego de haber llenado los campos, se guarda la misma en la base de datos del sistema.	

Tabla 17 Tarea de Ingeniería HU Insertar técnico

Tarea de ingeniería	
Número de tarea: 13	Número de HU: 13
Nombre de la tarea: Insertar técnico	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2

Programador responsable: Sheila Tiá Ubeda
Descripción: El administrador, en el submenú, “Gestionar Técnico” selecciona la opción “Insertar” y el sistema le muestra una interfaz donde debe llenar los campos solicitados (Nombre, Apellidos, Área, Teléfono de contacto, Ping). Luego de haber llenado los campos, adicionado el nuevo técnico, se guarda el mismo en la base de datos del sistema.

Tabla 18 Tarea de Ingeniería HU Seleccionar opción de distribución

Tarea de ingeniería	
Número de tarea: 19	Número de HU: 19
Nombre de la tarea: Seleccionar opción de distribución	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Sheila Tiá Ubeda	
Descripción: El administrador selecciona en el submenú “Opción de distribución”, se listan las opciones: “Automática” y “Manual”. Selecciona la opción deseada, y el sistema muestra una interfaz mostrando el nombre de la opción seleccionada y debajo el campo Checkbox “Activo” para marcar o desmarcar la opción. Una vez que active una la otra de desactivará.	

Tabla 19 Tarea de ingeniería HU Distribuir carga de trabajo (Manual)

Tarea de ingeniería	
Número de tarea: 21	Número de HU: 21
Nombre de la tarea: Distribuir carga de trabajo (Manual)	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Sheila Tiá Ubeda	
Descripción: Luego de previamente haber seleccionado la opción de balanceo de carga “Manual” por el administrador, el operador selecciona el menú “Distribuir carga”, se listan las incidencias registradas, selecciona una de estas y oprime la opción “Asignar” y se muestra una interfaz mostrando breves datos de la incidencia seleccionada (Servicio afectado, Estado, Tipo) y el listado de técnicos registrados en la base de datos. La selección puede ser múltiple. Si el técnico seleccionado ya ha sido asignado anteriormente, se muestra un cartel avisándolo. Luego de seleccionar correctamente el técnico, en el listado de incidencias se mostrará un campo “Técnico” mostrando el nombre y apellidos del mismo.	

Tabla 20 Tarea de Ingeniería 20 HU Distribuir carga de trabajo (Automática)

Tarea de ingeniería	
Número de tarea: 20	Número de HU: 20
Nombre de la tarea: Distribuir carga de trabajo (Automática)	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Sheila Tiá Ubeda	

Descripción: Luego de previamente haber seleccionado la opción de distribución de carga “Automática” por el administrador, todo el proceso de asignación aplicando el algoritmo de distribución de la carga de trabajo.

Inicialmente se capturan los datos necesarios de la incidencia y de los técnicos registrados. Si hay técnicos disponibles, más de una incidencia pendiente con iguales prioridades y tiempos de espera, se aplica el RR, si el área del técnico seleccionado es igual al tipo de incidencia en cuestión, se asigna la incidencia. De no haber más de una incidencia pendiente, se aplica el RR y si el área del técnico seleccionado es igual al tipo de incidencia en cuestión, se asigna la incidencia. Si hay más de una incidencia pendiente y no tienen igual prioridad, se asigna por el mayor valor de esta, comprobando luego que el área del técnico seleccionado es igual al tipo de incidencia en cuestión. Si hay más de una incidencia pendiente, con iguales prioridades y diferentes tiempos de espera en el servidor se asigna según el mayor valor de este parámetro, comprobando luego que el área del técnico seleccionado es igual al tipo de incidencia en cuestión. Si no hay técnicos disponibles se espera a que se inserte una nueva incidencia o un nuevo técnico para volver a ejecutar el algoritmo.

3.2 Pruebas

La calidad del software es un proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan. (Pressman y Ph, 2010)

Las pruebas representan el último bastión desde donde puede valorarse la calidad y, de manera más pragmática, descubrirse errores. La calidad se incorpora en el software a lo largo de todo el proceso de ingeniería del software. La adecuada aplicación de métodos y herramientas, revisiones técnicas efectivas y gestión y medición sólidas conducen a la calidad que se confirma durante las pruebas. (Pressman y Ph, 2010)

Propio de la metodología XP, se lleva a cabo la Fase de Prueba. Durante el desarrollo de software, establece probar constantemente como sea posible, esto permite un aumento de la calidad del sistema desarrollado, reduciendo el número de errores no detectados. XP divide las pruebas en dos grupos: las pruebas unitarias y las pruebas de aceptación.

3.2.1 Pruebas Unitarias

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Las pruebas deben ser definidas antes de realizar el código. Que todo código liberado pase correctamente

las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código. (Somerville, 2006)

Después de que las historias han sido desarrolladas y de que se ha hecho el trabajo de diseño preliminar, el equipo no inicia la codificación, sino que desarrolla una serie de pruebas unitarias a cada una de las historias que se van a incluir en la entrega en curso. Una vez creada la prueba unitaria, el desarrollador está mejor capacitado para centrarse en lo que debe implementarse para pasar la prueba. Una vez que

el código está terminado, se le aplica de inmediato una prueba unitaria, con lo que se obtiene retroalimentación instantánea para los desarrolladores. (Pressman y Ph, 2010)

Se aplicaron 4 pruebas unitarias, a las funcionalidades de “Distribuir carga de trabajo” con sus dos opciones, “Automática” y “Manual”, “Insertar incidencia” e “Insertar técnico”, haciendo uso del propio PyCharm que permite realizar este tipo de pruebas al código.

A continuación, se muestra el código resultante.

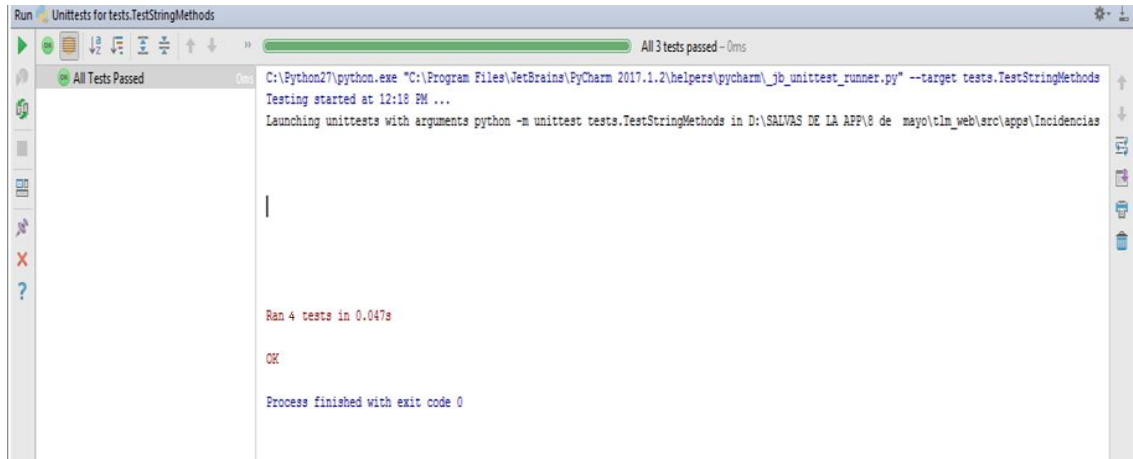


Fig. 11 Resultado de las pruebas unitarias

En la ejecución de las pruebas se obtuvieron en la primera iteración 9 errores, en la segunda 5 en la tercera 1, todas resueltas. Seguido se muestra la descripción de los resultados.



Fig. 12 Descripción de los resultados de las pruebas unitarias

3.2.2 Pruebas de Aceptación

Las pruebas de aceptación son creadas en base a las historias de usuario, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Las pruebas de aceptación son consideradas como

“pruebas de caja negra”. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. (Somerville, 2006)

Se realizaron pruebas de aceptación a cada HU, respondiendo a los siguientes aspectos:

- Nombre: Identifica la prueba descrita.
- Código: Identificador.
- Descripción: Breve descripción de la prueba de la funcionalidad a probar.
- Condiciones de ejecución: Condiciones que deben cumplirse para poder realizar la prueba.
- Pasos/ Entradas: descripción de cada uno de los pasos realizados durante el desarrollo de la prueba, se tendrá en cuenta cada una de las entradas que hace el usuario con el objetivo de verificar si se obtiene el resultado esperado.
- Resultado esperado: breve descripción del resultado que se espera obtener con la prueba.
- Resultado Obtenido: breve descripción del resultado obtenido con la prueba.
- Evaluación de la prueba: acorde al resultado de la prueba realizada se emitirá una evaluación.

A continuación, se muestran las pruebas de aceptación correspondientes a las HU “Insertar incidencia”, “Insertar Técnico”, “Seleccionar opción de distribución” y “Distribuir carga”. El resto puede encontrarse en [Anexos](#).

Tabla 21 Prueba de aceptación HU Insertar incidencia

Prueba de aceptación	
Código:HU1_P1	HU: Insertar Incidencia
Nombre: Insertar Incidencia	
Descripción: El operador adiciona las incidencias al sistema llenando los campos requeridos. (Tipo, Estado, Servicio afectado, Descripción, Localización, Fecha de ocurrencia, Prioridad, introducir datos del solicitante; Nombre y Apellidos, Dirección particular y Teléfono de contacto).	
Condiciones de ejecución: El operador tiene que estar autenticado en el sistema.	
Pasos/ Entradas: En el menú “Gestionar incidencia”	
<ol style="list-style-type: none"> 1. Oprimir opción “Adicionar”. 2. Llenar campos (Tipo, Estado, Servicio afectado, Descripción, Localización, Fecha de ocurrencia, Prioridad, introducir datos del solicitante: Nombre y Apellidos, Dirección particular y Teléfono de contacto). 3. Oprimir opción “Aceptar”. 	
Resultado esperado: El sistema adiciona la incidencia.	
Resultado Obtenido: El sistema adiciona la incidencia.	
Evaluación de la prueba: Satisfactoria	

Tabla 22 Prueba de aceptación HU Insertar técnico

Prueba de aceptación	
Código:HU21_P21	HU: Insertar técnico
Nombre: Insertar técnico	
Descripción: El administrador del sistema adiciona el técnico en el sistema llenando los campos requeridos. (Nombre, Apellidos, Área, Teléfono de contacto, Ping)	
Condiciones de ejecución: El operador tiene que estar autenticado en el sistema.	
Pasos/ Entradas: En el menú "Gestionar técnico" <ol style="list-style-type: none"> 1. Oprimir opción "Adicionar". 2. Llenar campos (Nombre, Apellidos, Área, Teléfono de contacto, Ping). 3. Oprimir opción "Aceptar". 	
Resultado esperado: El sistema adiciona el técnico.	
Resultado Obtenido: El sistema adiciona el técnico.	
Evaluación de la prueba: Satisfactoria	

Tabla 23 Prueba de aceptación HU Seleccionar opción de distribución

Prueba de aceptación	
Código:HU31_P31	HU: Seleccionar opción de distribución
Nombre: Seleccionar opción de distribución	
Descripción: El administrador del sistema selecciona la opción de balanceo de carga: "Automática" o "Manual".	
Condiciones de ejecución: El operador tiene que estar autenticado en el sistema.	
Pasos/ Entradas: En el menú "Seleccionar opción de distribución" <ol style="list-style-type: none"> 1. Seleccionar opción deseada. 2. Oprimir opción "Seleccionar". 3. Marcar o desmarcar la opción "Activo" según se desee. 4. Oprimir opción "Aceptar". 	
Resultado esperado: El sistema marca el ícono "Activo" según la acción realizada.	
Resultado Obtenido: El sistema marca el ícono "Activo".	
Evaluación de la prueba: Satisfactoria	

Tabla 24 Prueba de aceptación HU Distribuir carga de trabajo

Prueba de aceptación	
Código:HU32_P32	HU: Seleccionar opción de distribución
Nombre: Distribuir carga de trabajo	

<p>Descripción: Luego de previamente haber seleccionado la opción de balanceo de carga “Manual”, el administrador selecciona el menú “Distribuir carga” y el sistema lista las incidencias registradas, selecciona una de estas y oprime la opción “Asignar” y se muestra una ventana mostrando breves datos de la incidencia seleccionada y el listado de técnicos registrados en el sistema.</p>
<p>Condiciones de ejecución: El operador tiene que estar autenticado en el sistema. El operador tiene que estar autenticado en el sistema. Deben de existir incidencias registradas en la base de datos.</p>
<p>Pasos/ Entradas: En el menú “Distribuir carga”</p> <ol style="list-style-type: none"> 1. Seleccionar incidencia deseada. 2. Oprimir opción “Asignar”. 3. Seleccionar el(los) técnico(s) deseado(s). 4. Oprimir opción “Aceptar”.
<p>Resultado esperado: El sistema asigna la incidencia al técnico seleccionado.</p>
<p>Resultado Obtenido: El sistema asigna la incidencia al técnico seleccionado.</p>
<p>Evaluación de la prueba: Satisfactoria</p>

Las pruebas realizadas favorecieron la calidad y las funcionalidades del sistema, en su ejecución se obtuvieron en la primera iteración 12 no conformidades, en la segunda 6 y en la tercera 2, todas resueltas. Logrando así la satisfacción del cliente y el cumplimiento de los objetivos planteados. A continuación, se describen los resultados.



Fig. 13 Descripción de los resultados de las pruebas de aceptación

3.3 Conclusiones parciales

En este capítulo se implementaron las Tareas de la Ingeniería definidas para cada HU, teniendo así un mejor acercamiento con el código. Finalmente, consumado el diseño de los casos de prueba funcionales,

la aplicación fue probada y seguidamente fueron resueltas todas las no conformidades. Obteniendo así el producto terminado que cumple con los requerimientos especificados por el cliente.

Conclusiones generales

En la presente investigación se estudiaron los procesos relacionados con la gestión de incidencias y la distribución de la carga de trabajo asociada, obteniendo una mejor comprensión de cómo estos funcionan. Se seleccionaron las herramientas y la metodología de desarrollo de software a utilizar. Se aplicaron los patrones arquitectónicos y de diseño para una correcta implementación de la aplicación. Se realizaron las pruebas definidas por la metodología que permitió la validación del cliente. Se logró crear un sistema de gestión de incidencias genérico capaz de distribuir la fuerza de trabajo asociada. Con su empleo las empresas que brindan servicios podrán prestarlos con mayor eficiencia ya que se atenderán las incidencias en menor tiempo. Los trabajadores serán beneficiados ya que la carga de trabajo se distribuirá de manera equitativa. Todo esto redundará en beneficio para los clientes ya que recibirán un servicio completo y de calidad y a la vez para la empresa que fidelizará a su clientela. Por todo lo anteriormente expuesto, se concluye que los objetivos propuestos para el presente trabajo se han cumplido satisfactoriamente, desarrollando las tareas propuestas para el desarrollo del sistema.

Recomendaciones

A partir de las conclusiones abordadas se listan las recomendaciones en vistas de posibles mejoras:

- Desarrollar el sistema que emplea la tecnología VR, integrándolo con el presente para explotar al máximo las posibilidades de este tipo de integración y de esta manera contar con un sistema genérico.

Referencias bibliográficas

Academia. [en línea], 2018. Disponible en:

http://www.academia.edu/4726489/Patrones_de_Diseño_Contenido_Patrones_GRASP_Experto_Creador_Bajo_Acoplamiento.

Acerca de JavaScript. [en línea], 2018. Disponible en:

https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca_de_JavaScript.

BBVA OPEN4U, [sin fecha]. BBVA APi-Marquet. [en línea]. Disponible en:

<https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>.

CAMACHO, E., CARDESO, F. y NUÑEERIKAZ, G., 2004. Arquitecturas de Software. S.l.: s.n.

CONTRIBUTORS, D.S.F. and individual, 2018. django. [en línea]. Disponible en:

<https://www.djangoproject.com/>.

DESK-SPANISH-SPAIN, C.N.S., 2017. Tratamiento del ticket y conceptos relacionados. [en línea].

Disponible en:

<https://docops.ca.com/display/NSDFall2013ES/Tratamiento+del+ticket+y+conceptos+relacionados>

Django REST framework. [en línea], 2018. Disponible en: <http://www.django-rest-framework.org/>.

FARMER, E., BROWSON, A. y QINETIQ, 2003. Review of Workload Measurement, Analysis and Interpretation Methods. [en línea]. S.l.: Disponible en:

<http://citeseerx.ist.psu.edu/viewdoc/%0Adownload?doi=10.1.1.121.3382&rep=rep1&type=pdf>.

GIL DMÍNGUEZ, O., 2015. Fundamentos de Redes de Voz IP. S.l.: s.n.

GONZÁLEZ, H., 2011. Calidad y gestión. [en línea]. Disponible en:

<https://calidadgestion.wordpress.com/tag/satisfaccion-del-cliente-en-iso-9000/>.

HTML5. [en línea], 2018. Disponible en: <https://lenguajehtml.com/>.

IBM DEVELOPERWORKS®, 2002. Clustering: A basic 101 tutorial. [en línea]. Disponible en:

<https://www.ibm.com/developerworks/aix/tutorials/clustering/clustering.html>.

INTEGRIA, 2018. INTEGRIA IMS. [en línea]. Disponible en: <https://integriaims.com/>.

INTEGRIAIMS, 2016. Gestión de incidencias: cómo evitar clientes insatisfechos y quejas. [en línea].

Disponible en: <https://integriaims.com/gestion-incidencias-evitar-clientes-insatisfechos-quejas/>.

INTERNETARCHIVEWAYBACKMACHINE, 2007. ObjectWebOpenSource. [en línea]. Disponible en:

<https://web.archive.org/web/20050507151935/http://middleware.objectweb.org/>.

JETBRAINS, 2018. PyCharm. [en línea]. Disponible en: <https://www.jetbrains.com/pycharm/download/>.

KOTLER, P. y G.A., 2008. Fundamentos de Marketing 8va Edición. S.l.: s.n.

LETELIER, P. y PENADÉS, M.C., 2006. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Técnica administrativa [en línea], vol. 5, no. 26, pp. 17. ISSN 01882503. DOI 1666-1680. Disponible en: <http://www.jstor.org/stable/3541599?origin=crossref>.

LibrosWeb. [en línea], 2006. Disponible en:

http://librosweb.es/libro/django_1_0/capitulo_5/el_patron_de_diseno_mtv.html.

MELÉNDEZ VALLADAREZ, S.M., ELIZABETH GAITAN, M. y PÉREZ REYES, N.N., 2016. Sistema web de evaluación al desempeño docente UNANMANAGUA, empleando la metodología ágil Programación Extrema, en el II semestre del 2015. S.I.: Universidad Nacional Autónoma de Nicaragua, Managua.

OTRS, 2017. OTRS Open technology Real Services. S.I.: s.n.

PARK, J.J., ZOMAYA, A., YEO, S.-S. y SAHNI, S., 2012. Network and Parallel Computing: 9th IFIP International Conference, NPC 2012 Proceedings. S.I.: s.n.

PRESSMAN, R.S. y PH, D., 2010. Ingeniería del software. S.I.: s.n. ISBN 9786071503145.

PYTHON SOFTWARE FOUNDATION, 2018. Python. [en línea]. Disponible en:

<https://www.python.org/about/>.

Qué es BackboneJS. [en línea], 2018. Disponible en: <https://desarrolloweb.com/articulos/que-es-backbonejs.html>.

RIZO SANDOVAL, H.A., 2015. Balanceador de carga de llamadas para sistemas de call center multicola implementados sobre la plataforma de software libre asterisk. S.I.: Pontificia Universidad Javeriana.

SCIENCEDIRECT, 2017. Occupational Heat Stress Impacts on Health and Productivity in a Steel Industry in Southern India. [en línea]. Disponible en:

<https://www.sciencedirect.com/science/article/pii/S2093791116302220>.

SERVICETONIC, 2017. ITIL v3. Gestión de Incidencias. [en línea]. Disponible en:

<https://www.servicetonic.es/itil/itil-v3-gestion-de-incidencias/>.

SOMERVILLE, I., 2006. Ingeniería del Software 7ma. Edición. S.I.: s.n.

TANENBAUM, A.S., 2009. Sistemas Operativos Modernos Tercera Edición. 3ra. S.I.: s.n. ISBN 9786074420463.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2018. PostgreSQL. [en línea]. Disponible en: <https://www.postgresql.org/>.

VALOTTO PATUZZO, G., 2011. Las transformaciones en al consideración del sector de servicios. S.I.: s.n.

VÁZQUEZ, M.Y.L., PEDRO, J. y RODRÍGUEZ, F., [sin fecha]. Métodos Ágiles en el Desarrollo de Software para la Bioinformática. , no. 2.

VAZQUEZ GARCIA, J. y GARCÍA FUENTES, D., 2012. Universidad de las Ciencias Informáticas Facultad 2 Título: Gestión de la fuerza de trabajo móvil para la red de acceso de ETECSA. S.I.: Universidad de las Ciencias Informáticas.

VISUAL PARADIGM, 2018. Visual Paradigm. [en línea]. Disponible en: <https://www.visual-paradigm.com/>.

W3.CSS, 2018. W3Schools. [en línea], Disponible en: <https://www.w3schools.com/css/default.asp>.

ZOHO CORP, 2017. ManageEngine ServiceDesk Plus. [en línea]. Disponible en:
<https://www.manageengine.com/es/service-desk/>.

Bibliografía

Academia. [en línea], 2018. Disponible en:

http://www.academia.edu/4726489/Patrones_de_Diseño_Contenido_Patrones_GRASP_Experto_Creador_Bajo_Acoplamiento.

Acerca de JavaScript. [en línea], 2018. Disponible en:

https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca_de_JavaScript.

BBVA OPEN4U, [sin fecha]. BBVA APi-Marquet. [en línea]. Disponible en:

<https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>.

CAMACHO, E., CARDESO, F. y NUÑEERIKAZ, G., 2004. Arquitecturas de Software. S.l.: s.n.

CONTRIBUTORS, D.S.F. and individual, 2018. django. [en línea]. Disponible en:

<https://www.djangoproject.com/>.

DESK-SPANISH-SPAIN, C.N.S., 2017. Tratamiento del ticket y conceptos relacionados. [en línea].

Disponible en:

<https://docops.ca.com/display/NSDFall2013ES/Tratamiento+del+ticket+y+conceptos+relacionados>

Django REST framework. [en línea], 2018. Disponible en: <http://www.django-rest-framework.org/>.

FARMER, E., BROWSON, A. y QINETIQ, 2003. Review of Workload Measurement, Analysis and Interpretation Methods. [en línea]. S.l.: Disponible en:

<http://citeseerx.ist.psu.edu/viewdoc/%0Adownload?doi=10.1.1.121.3382&rep=rep1&type=pdf>.

GIL DMÍNGUEZ, O., 2015. Fundamentos de Redes de Voz IP. S.l.: s.n.

GONZÁLEZ, H., 2011. Calidad y gestión. [en línea]. Disponible en:

<https://calidadgestion.wordpress.com/tag/satisfaccion-del-cliente-en-iso-9000/>.

HTML5. [en línea], 2018. Disponible en: <https://lenguajehtml.com/>.

IBM DEVELOPERWORKS®, 2002. Clustering: A basic 101 tutorial. [en línea]. Disponible en:

<https://www.ibm.com/developerworks/aix/tutorials/clustering/clustering.html>.

INTEGRIA, 2018. INTEGRIA IMS. [en línea]. Disponible en: <https://integriaims.com/>.

INTEGRIAIMS, 2016. Gestión de incidencias: cómo evitar clientes insatisfechos y quejas. [en línea].

Disponible en: <https://integriaims.com/gestion-incidencias-evitar-clientes-insatisfechos-quejas/>.

INTERNETARCHIVEWAYBACKMACHINE, 2007. ObjectWebOpenSource. [en línea]. Disponible en:

<https://web.archive.org/web/20050507151935/http://middleware.objectweb.org/>.

JETBRAINS, 2018. PyCharm. [en línea]. Disponible en: <https://www.jetbrains.com/pycharm/download/>.

KOTLER, P. y G.A., 2008. Fundamentos de Marketing 8va Edición. S.l.: s.n.

LETELIER, P. y PENADÉS, M.C., 2006. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Técnica administrativa [en línea], vol. 5, no. 26, pp. 17. ISSN 01882503. DOI 1666-1680. Disponible en: <http://www.jstor.org/stable/3541599?origin=crossref>.

LibrosWeb. [en línea], 2006. Disponible en:

http://librosweb.es/libro/django_1_0/capitulo_5/el_patron_de_diseno_mtv.html.

MELÉNDEZ VALLADAREZ, S.M., ELIZABETH GAITAN, M. y PÉREZ REYES, N.N., 2016. Sistema web de evaluación al desempeño docente UNANMANAGUA, empleando la metodología ágil Programación Extrema, en el II semestre del 2015. S.I.: Universidad Nacional Autónoma de Nicaragua, Managua.

OTRS, 2017. OTRS Open technology Real Services. S.I.: s.n.

PARK, J.J., ZOMAYA, A., YEO, S.-S. y SAHNI, S., 2012. Network and Parallel Computing: 9th IFIP International Conference, NPC 2012 Proceedings. S.I.: s.n.

PRESSMAN, R.S. y PH, D., 2010. Ingeniería del software. S.I.: s.n. ISBN 9786071503145.

PYTHON SOFTWARE FOUNDATION, 2018. Python. [en línea]. Disponible en:

<https://www.python.org/about/>.

Qué es BackboneJS. [en línea], 2018. Disponible en: <https://desarrolloweb.com/articulos/que-es-backbonejs.html>.

RIZO SANDOVAL, H.A., 2015. Balanceador de carga de llamadas para sistemas de call center multicola implementados sobre la plataforma de software libre asterisk. S.I.: Pontificia Universidad Javeriana.

SCIENCEDIRECT, 2017. Occupational Heat Stress Impacts on Health and Productivity in a Steel Industry in Southern India. [en línea]. Disponible en:

<https://www.sciencedirect.com/science/article/pii/S2093791116302220>.

SERVICETONIC, 2017. ITIL v3. Gestión de Incidencias. [en línea]. Disponible en:

<https://www.servicetonic.es/itil/itil-v3-gestion-de-incidencias/>.

SOMERVILLE, I., 2006. Ingeniería del Software 7ma. Edición. S.I.: s.n.

TANENBAUM, A.S., 2009. Sistemas Operativos Modernos Tercera Edición. 3ra. S.I.: s.n. ISBN 9786074420463.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2018. PostgreSQL. [en línea]. Disponible en: <https://www.postgresql.org/>.

VALOTTO PATUZZO, G., 2011. Las transformaciones en al consideración del sector de servicios. S.I.: s.n.

VÁZQUEZ, M.Y.L., PEDRO, J. y RODRÍGUEZ, F., [sin fecha]. Métodos Ágiles en el Desarrollo de Software para la Bioinformática. , no. 2.

VAZQUEZ GARCIA, J. y GARCÍA FUENTES, D., 2012. Universidad de las Ciencias Informáticas Facultad 2 Título: Gestión de la fuerza de trabajo móvil para la red de acceso de ETECSA. S.I.: Universidad de las Ciencias Informáticas.

VISUAL PARADIGM, 2018. Visual Paradigm. [en línea]. Disponible en: <https://www.visual-paradigm.com/>.

W3.CSS, 2018. W3Schools. [en línea], Disponible en: <https://www.w3schools.com/css/default.asp>.

ZOHO CORP, 2017. ManageEngine ServiceDesk Plus. [en línea]. Disponible en:
<https://www.manageengine.com/es/service-desk/>.

Anexos

Entrevistas realizadas

Nombre: Lizandra Pérez Mejías.

Tipo de comunicación: Personal.

Fecha: 23 de octubre de 2017.

Preguntas:

1. ¿Qué sistemas de gestión de incidencias se emplean en la Universidad de Ciencias Informáticas (UCI)?
2. ¿Con qué sistema trabaja su centro y cómo realiza el proceso de distribución de la carga de trabajo? ¿Emplean mecanismos de balanceo de carga para el mismo?

Respuestas:

1. En la universidad ahora mismo creo que no hay muchos sistemas de incidencias. Te explico, la DGT (Dirección de Gestión Tecnológica) tenía anteriormente uno que se llama GatServer y que lo había hecho un tesista creo, que estaba adaptado al modo de trabajo de ellos ahí, no sé hasta qué punto se podía adaptar a otros negocios, pero de ese nunca tuve la administración. Este sistema se dejó de usar porque no tenía quien le diera soporte, el muchacho que lo atendía se fue de la UCI y si había que hacer algún cambio o algo no había quien lo hiciera. Entonces se llegó al acuerdo de implantar OTRS para ellos también. Porque OTRS es una plataforma libre, que descargaron de internet y que se puede adaptar fácilmente a casi cualquier negocio, por no ser generalizadora y decir a todos, claro con sus limitaciones, pero bastante adaptable.
El otro lugar donde registran incidencias es en la DSI (Dirección de Seguridad Informática), que usan OTRS también, pero en un servidor aparte al de la UCI que es el que yo administro, sin embargo, es el mismo sistema. El otro lugar donde se gestionan tickets es en el Centro de Soporte, pero se gestionan por el mismo portal de ellos soporte.uci.cu.
2. Trabajamos con OTRS. El sistema no asigna los tickets por ningún mecanismo de balanceo de carga, pero tampoco se hace de forma manual. Esto más bien se debe al sistema de trabajo que se implementa en las áreas que tienen el flujo de trabajo montado en el sistema, te explico más o menos.... En el área de mantenimiento el ticket no se asigna a un técnico en específico, se asignan los de carpintería, por ejemplo, al jefe de brigada de carpintería, no específicamente a uno de los técnicos que conforman la brigada. Esto fue lo que ellos quisieron que se hiciera y para esa asignación se usan Agentes genéricos que vienen siendo como los triggers en base de datos, cuando ocurre un evento determinado o en ciertas horas que hemos determinado estos agentes genéricos se ejecutan y distribuyen los tickets que se encuentran en la cola madre a las distintas colas específicas en las cuales se encuentran asignados los jefes de brigada. El sistema hasta ahora no aplica ningún mecanismo de balanceo de carga para esto que me dices, hasta ahora tampoco nos ha hecho falta que lo haga así por lo que te explicaba anteriormente.

Nombre: Marlen González Cruz.

Tipo de comunicación: Personal.

Fecha: 25 de octubre de 2017.

Preguntas:

1. ¿Con qué sistema se trabaja en Dirección de Gestión Tecnológica?
2. ¿Qué no conformidad le encuentra?

Respuestas:

1. OTRS.
2. No brinda la posibilidad de gestionar el local donde se localiza la incidencia, dificultando su gestión ya que el usuario escribe el mismo dando cabida a incoherencias de escrituras u otros errores de escritura que vuelven engorroso la gestión de este parámetro.

Nombre: Neybis Lago Clara.

Tipo de comunicación: Personal.

Fecha: 29 de octubre de 2017.

Preguntas:

1. ¿Con qué sistema trabaja su centro y cómo realiza el proceso de distribución de la carga de trabajo?
2. ¿Emplean mecanismos de balanceo de carga para el mismo?
3. ¿Qué inconformidades ha podido detectar desde su experiencia?

Respuestas:

1. El Centro de Soporte de la universidad gestionan sus tickets por la plataforma para la gestión de incidencias ManageEngine ServiceDesk Plus 9.0, software privativo que entre unos de sus módulos se encuentra el de análisis y solicitudes en general que gestiona sus incidencias, el registro de la incidencia se realiza de tres formas, mediante un correo que automáticamente crea el ticket, directamente el usuario en el portal web o mediante una llamada telefónica y un personal designado a esta tarea lo registra en el portal. La asignación de los tickets se realiza manualmente, mediante una lista con los especialistas que cuenta el centro se le asocia la incidencia, de igual forma la distribución de la fuerza de trabajo no se tiene informatizada.
2. No.
3. Como inconveniente cabe mencionar que el sistema es privado, por lo que no se tiene acceso al código fuente y no permite desagregar los módulos que no se necesitan.