



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
La Universidad Católica de Loja

ÁREA TÉCNICA

TÍTULO DE INGENIERO EN INFORMÁTICA

Desarrollo de una solución software empresarial para control de vacaciones anuales en la Empresa Importadora Comercial El Hierro Cia.Ltda.

TRABAJO DE TITULACIÓN.

AUTOR: Lloret Orellana, Antonio

DIRECTOR: Guamán Coronel, Daniel Alejandro, Mgtr

CENTRO UNIVERSITARIO CUENCA - ECUADOR

2015



Esta versión digital, ha sido acreditada bajo la licencia Creative Commons 4.0, CC BY-NY-SA: Reconocimiento-No comercial-Compartir igual; la cual permite copiar, distribuir y comunicar públicamente la obra, mientras se reconozca la autoría original, no se utilice con fines comerciales y se permiten obras derivadas, siempre que mantenga la misma licencia al ser divulgada. <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>

Septiembre, 2015

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

Magister.
Daniel Alejandro Guamán Coronel.

DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de titulación: Desarrollo de una solución software empresarial para control de vacaciones anuales en la Empresa Importadora Comercial El Hierro Cia.Ltda.; realizado por Antonio Lloret Orellana; ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

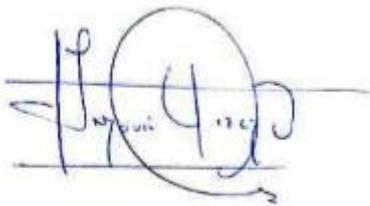
Loja, octubre de 2015.

f)

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

“ Yo Lloret Orellana Antonio declaro ser autor del presente trabajo de titulación: Desarrollo de una solución software para control de vacaciones anuales en la Empresa Importadora Comercial El Hierro Cía. Ltda., de la Titulación Ingeniero en Informática, siendo Daniel Alejandro Guamán Coronel director del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o institucional (operativo) de la Universidad”

A handwritten signature in blue ink, appearing to read 'Lloret Orellana Antonio', written over a horizontal line.

Autor: Lloret Orellana Antonio
Cédula: 0101562643

DEDICATORIA

A mi esposa Jacqueline y mi hija María Rosa, porque sacrificaron su tiempo, su espacio y me apoyaron con su sonrisa permanente, su entusiasmo diario, su complicidad y me dieron la fuerza y confianza necesaria para culminar una etapa de vida que estaba en deuda.

A mi familia, porque en ella nace y se desarrolla la gratitud, la solidaridad, el respeto, los valores, porque es ese espacio de vida compartida, donde se aplauden los éxitos y se recomponen los fracasos; porque es lo más auténtico que hemos creado.

AGRADECIMIENTO

A la Universidad Técnica Particular de Loja por brindarme la oportunidad de aprender y formarme como profesional, de lograr nuevos desafíos y proponer soluciones que permitan alcanzar mejores días para nuestro país.

A la Empresa Importadora Comercial El Hierro por facilitar sus equipos e instalaciones, a su equipo de directivos y profesionales que coadyuvaron a la terminación de este proyecto académico.

Mi gratitud y agradecimiento al Magister Daniel Guamán Coronel Director de Tesis.

“El agradecimiento es la memoria del corazón” Lao Tsé

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN.....	¡Error! Marcador no definido.
DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS.....	ii
DEDICATORIA.....	iii
AGRADECIMIENTO.....	iv
INDICE DE CONTENIDOS.....	v
RESUMEN.....	8
ABSTRACT.....	9
INTRODUCCIÓN.....	10
CAPÍTULO 1.....	14
MARCO TEÓRICO.....	14
1.1. Metodología de desarrollo aplicada.....	15
1.2. Arquitectura de información.....	15
1.2.1. Fase de inicio.....	17
1.2.2. Fase de consolidación.....	18
1.2.3. Fase de validación de la AI.....	19
1.3. Metodología RUP.....	20
1.3.1. Definiciones de RUP.....	20
1.3.2. Dimensiones de RUP.....	21
1.3.3. Características fundamentales de RUP.....	22
1.3.4. Artefactos de RUP.....	24
1.3.4.1. <i>Artefactos a desarrollar</i>	24
1.3.5. Ciclo de vida de la metodología de desarrollo de software RUP.....	27
1.4. Arquitectura de Software.....	28
1.4.1. Importancia de la arquitectura de software.....	29
1.4.2. Factores que influyen en la arquitectura de software.....	30
1.4.3. Patrones arquitectónicos.....	30
1.4.4. Estilos arquitectónicos.....	31
1.4.4.1. <i>Estilo Cliente-Servidor</i>	33
1.4.4.2. <i>Modelo de programación por capas</i>	35
1.4.4.3. <i>Arquitectura Cliente-Servidor en tres niveles</i>	38
1.4.4.4. <i>Cuadro resumen de los estilos arquitectónicos de software</i>	40
1.4.5. Selección del estilo arquitectónico de software.....	40
1.4.6. Lenguajes de cuarta generación (4GL).....	42
1.5. Informix.....	43
1.5.1. Definiciones de Informix.....	43
1.5.2. Informix 4GL.....	46
CAPÍTULO 2.....	48

DEFINICIÓN DEL PROBLEMA Y ANÁLISIS INICIAL	48
2.1. Introducción.	49
2.2. Descripción del proceso actual.....	49
2.3. Problemática.	52
2.4. Plan de solución.....	53
2.5. Descripción de la arquitectura.	56
2.6. Beneficios de la solución.....	58
2.7. Resultados esperados.....	59
CAPÍTULO 3.....	60
DESARROLLO DE LA SOLUCIÓN.....	60
3.1. Introducción.	61
3.2. Metodología de desarrollo de software.....	61
3.3. Fases de la metodología de desarrollo de software.....	62
3.4. Fase de inicio.....	63
3.4.1. Estudio del proceso actual.	63
3.4.2. Entrevistas.	65
3.4.3. Observación directa.	65
3.4.4. Documento Visión.....	65
3.4.4.1. <i>Definición del problema actual.</i>	65
3.4.4.2. <i>Funciones del personal involucrado en el proyecto.</i>	66
3.4.5. Especificación de requerimientos.....	67
3.4.5.1. <i>Visión de la solución software.</i>	69
3.4.5.2. <i>Características de la solución software.</i>	69
3.5. Fase de elaboración.	70
3.5.1. Especificaciones de casos de uso.	70
3.5.2. Arquitectura de software propuesta.....	71
3.5.3. Descripción del patrón arquitectónico Cliente–Servidor.	73
3.6. Fase de construcción.....	75
3.6.1. Cliente.....	76
3.6.2. Diseño de la interfaz del usuario.	76
3.6.3. Servidor.	79
3.7. Fase de transición.....	81
CAPÍTULO 4.....	82
PRUEBAS DE VALIDACIÓN	82
4.1. Introducción.	83
4.2. Tipos de pruebas.	83
4.3. Estrategia de pruebas.	84
4.4. Herramientas para pruebas.....	85
4.5. Configuración del ambiente de pruebas.	86

4.6.	Ejecución y resultados de las pruebas.	87
4.6.1.	Pruebas unitarias.....	88
4.6.2.	Pruebas de integración.....	88
4.6.3.	Pruebas de sistema.....	90
4.6.4.	Pruebas funcionales.....	91
4.7.	Métricas del producto software.....	92
4.7.1.	Métricas en la fase de análisis.....	93
4.7.2.	Métricas en la fase de diseño e implementación.....	96
4.7.3.	Eficacia de la eliminación de defectos.....	97
	CAPÍTULO 5.....	99
	CONCLUSIONES, RECOMENDACIONES Y ACCIONES FUTURAS	99
	Conclusiones	100
	Recomendaciones	101
	Acciones futuras	102
	BIBLIOGRAFÍA.....	103
	GLOSARIO.....	106
	A N E X O S.....	107
	Anexo A: Cronograma anual de vacaciones	108
	Anexo B: Formulario de solicitud de vacaciones	109
	Anexo C: Reglamento interno	110
	Anexo D: Documento de visión.....	111
	Anexo E: Documento de especificación de requerimientos.....	123
	Anexo F: Documento de especificación de casos de uso	135
	Anexo G: Documento de arquitectura de software.....	144
	Anexo H: Documento de plan de pruebas.....	159
	Anexo I: Documento de manual de usuario	194

RESUMEN

El proyecto de fin de titulación ha sido desarrollado para automatizar el control de vacaciones anual de los empleados de la Empresa Importadora Comercial El Hierro, el mismo que venía ejecutándose de manera manual, lo cual evidentemente afectaba en la elaboración oportuna, confiable y segura del proceso global del control de nómina y liquidación de haberes por parte del Departamento de Talento Humano de la Empresa, además no estaba integrado al módulo principal de rol de pagos que la Empresa tiene automatizado.

Este proyecto se concentra en el: desarrollo e implementación de un módulo de programación en lenguaje 4GL de Informix que permita automatizar el registro de vacaciones anuales de los empleados de la Empresa y que se integre al sistema principal de Rol de Pagos; para lo cual se ha utilizado como metodología de desarrollo de software RUP, de tal manera que permita cumplir las diferentes fases de ejecución en función de las necesidades de la Empresa, con el objetivo final de brindar una solución de software de calidad, fácil mantenimiento, flexible a la normativa legal vigente y escalable de acuerdo al crecimiento futuro de la Empresa.

PALABRAS CLAVE: Automatizar, proceso, nómina, módulo, rol de pago, vacaciones, programación, lenguaje, software, Informix, 4GL, RUP, metodología, normativa, sistema principal, fases.

ABSTRACT

The end of degree project has been developed to automate the annual holiday control of the employees of the company Importing Commercial El Hierro, the same that had been running manually, which obviously affected in the development of timely, reliable and secure the global process of payroll and payroll by the Department of human resources of the company also was not integrated into the main module role of payments the company has automated. This project focuses on the: development and implementation of a module of programming in language 4GL Informix that allow to automate the register of annual leave of employees of the company and to integrate to the main role of payments system; It has been used as a methodology of software development RUP, in such a way to allow meet different phases of execution depending on the needs of the company, with the ultimate goal of providing a solution in quality software, easy to maintain, flexible, scalable and applicable legal regulations according to the future growth of the company.

Keywords: Automate, process payroll, module, role of payment, holiday, programming, language, software, Informix, 4GL, RUP, methodology, standards, master system, phases.

INTRODUCCIÓN

La automatización de procesos en el área de gestión administrativa y Talento Humano vinculadas con las tareas de control y supervisión tanto en las Empresas del sector privado como públicas se ha convertido en una prioridad por parte de los responsables de Talento Humano en tomar la decisión oportuna de implementar y ejecutar soluciones de software que permitan mejorar, agilizar y controlar las diferentes tareas de orden administrativo que se presentan en las diferentes Empresas.

El propósito general del Trabajo de Fin de Titulación es desarrollar una solución de software que permita automatizar el registro de control de vacaciones a que tienen derecho los empleados y trabajadores de la Empresa Importadora Comercial El Hierro de la ciudad de Cuenca.

Importadora Comercial El Hierro Cía. Ltda. Es una Empresa cuencana que se dedica a la importación y comercialización de materiales de construcción y artículos de hogar a nivel nacional, al momento cuenta con una nómina de 300 empleados.

La Empresa dispone de una plataforma tecnológica de punta que permite manejar de manera apropiada los sistemas de comercialización, contabilidad y Talento Humano; con soluciones de software desarrollado y diseñado de acuerdo a sus necesidades por el departamento de sistemas; sin embargo en el módulo de rol de pagos que tiene implementada la Empresa, carece de un proceso automatizado que permita llevar un control del registro de vacaciones anuales de los empleados; pues actualmente dicho proceso se lleva de forma manual, el mismo que de acuerdo al levantamiento de información y los datos proporcionados por el departamento de Talento Humano se destaca lo siguiente:

- Formulario de registro de vacaciones pre-impreso con información deficiente.
- Archivo o carpeta de documentos desordenada.
- Desconfianza de cálculo de vacaciones por parte del personal.
- Limitada información de los empleados sobre el derecho a vacaciones.
- Cronograma de vacaciones del personal no se cumple de acuerdo a la fecha de ingreso.
- Falta de reportes mensuales de seguimiento del personal.

- Limitado conocimiento por parte del Dpto. Sistemas y Talento Humano sobre la normativa para el cálculo y asignación de vacaciones emitida por el Ministerio de Trabajo.
- Pérdida de tiempo laboral por búsqueda de información en forma manual relacionada al empleado.
- El tradicionalismo en el que tiende a desenvolverse la Empresa.
- El asentamiento en el trabajo que conlleva una pasividad del empleado.
- Estructura organizativa del Departamento de Talento Humano.

Ante este panorama es necesario y prioritario desarrollar e implementar una solución de software que permita complementar e integrarse al módulo principal de rol de pagos que dispone actualmente la Empresa., de tal manera que permita optimizar y mejorar el control de vacaciones anuales de la Empresa.

El Departamento de Talento Humano de la Empresa controla una nómina promedio de trescientos empleados, sin embargo el control del registro de vacaciones de su personal lo elabora de forma manual, este procedimiento al no estar automatizado y vinculado con la emisión final del rol mensual de cada empleado puede ser causa de errores en el cálculo de liquidación de haberes del empleado cesante, así como en la programación anual de vacaciones de los empleados vigentes y control de asistencia del personal.

En consideración a la información levantada y los recursos de tecnología que dispone la Empresa, vale la pena mencionar aspectos fundamentales que justifican la necesidad de implementar una solución de software que permita facilitar las tareas del Departamento de Talento Humano:

- Disponer de la información registrada en una base de datos Informix
- Integrar la información registrada en el Menú Principal del Módulo De Rol de Pagos de la Empresa.
- Mejorar su interpretación y presentación del cronograma general de vacaciones.
- Emitir un correcto cálculo de asignación de vacaciones al personal apoyado en una fundamentación científica del proceso basado en las normativas y reglas proporcionadas por el Ministerio de Trabajo del Ecuador.
- Gestionar y organizar los registros individuales de vacaciones de cada uno de los empleados de la Empresa.
- Asignar automáticamente los días adicionales de vacación en función del tiempo de servicio.

- Calcular automáticamente los días utilizados y disponibles por concepto de vacaciones acreditadas.

En base a lo antes descrito se pretende desarrollar el trabajo de fin de titulación; partiendo de la información y recursos de software y tecnología proporcionados por los departamentos de Sistemas y Talento Humano de la Empresa. Los componentes de la solución de software serán desarrollados en lenguaje de programación de cuarta generación 4GL soportados en el motor de base de datos Informix; complementados con reportes generados en lenguaje de consulta estructurado SQL de la base de datos Informix; estos componentes se encuentran agrupados en el motor de la base de datos, lo que posibilita su fácil desarrollo, mantenimiento y uso.

Objetivos

Objetivo general

Desarrollar e implementar una solución software que automatice el registro de vacaciones anual del personal de empleados de la Empresa Imp. Com. El Hierro, utilizando como tecnología base la infraestructura de tratamiento de información que dispone el Departamento de Sistemas de la Empresa.

Objetivos específicos

- Levantamiento de requerimientos del Departamento de Sistemas y Talento Humano referentes al cálculo y asignación de vacaciones anuales de los empleados de Imp. Com. El Hierro.
- Aplicar la metodología de desarrollo RUP de aquellos componentes que intervienen directamente en la solución de software.
- Aplicar una arquitectura cliente-servidor en 3 capas para la solución software.
- Validación del aplicativo en un ambiente de pruebas de la solución software utilizando la infraestructura de TI de la Empresa Imp.Com. El Hierro.

El soporte de este proyecto está conformado por los siguientes capítulos:

Capítulo 1: Refleja el marco teórico que muestra una perspectiva global de los temas a tratar en el proyecto tales como: Arquitectura de la información; metodología de desarrollo de

software, selección de la arquitectura del software, tecnologías y herramientas utilizadas en el desarrollo de la solución software.

Capítulo 2: Este capítulo está conformado por el método de asignación y cálculo de vacaciones anuales que actualmente realiza la Empresa; análisis del método actual, planteamiento de la solución, beneficios de la automatización y resultados esperados de la solución de software a implementarse.

Capítulo 3: Este capítulo aborda el desarrollo de la solución, el cual contempla la aplicación de las diferentes fases de la metodología de desarrollo de software y demuestra la implementación de la arquitectura elegida.

Capítulo 4: Este capítulo contiene toda la documentación relativa a las pruebas y resultados que convalidan la solución de software desarrollada en el trabajo de fin de titulación.

Capítulo 5: En este capítulo se aborda todas las conclusiones, recomendaciones e integraciones futuras que se requieren en el proyecto.

Es importante resaltar el apoyo brindado, además de las facilidades dadas por los directivos y personal de los departamentos de Sistemas y Talento Humano de la Empresa Imp. Com. El Hierro para la implementación y puesta a punto del proyecto; la utilización de sus equipos informáticos, uso de licencias de la base de datos y sistema operativo, así como toda la documentación necesaria para la realización de pruebas y validaciones de la solución software a implementar.

CAPÍTULO 1.

MARCO TEÓRICO

1.1. Metodología de desarrollo aplicada.

En el presente capítulo se plantea un modelo de integración de la ingeniería del software, en el proceso unificado de desarrollo de software bajo la metodología RUP, poniendo especial interés en los roles, actividades y etapas que integran el desarrollo del módulo de control de vacaciones en la aplicación global del Rol de Pagos de la Empresa. El objetivo es asegurar que el producto final ofrezca una experiencia de usuario satisfactoria.

En la ingeniería del software el objetivo es construir un producto software o mejorar uno existente, como es el caso del modelo propuesto que es parte integrante de una aplicación real que se halla en ejecución en la Empresa Importadora Comercial El Hierro; con resultados que aspiran sean satisfactorios; de tal forma que este proceso de desarrollo sea capaz de evolucionar durante muchos años; para lo cual esta evolución debería limitar su alcance, en un momento del tiempo dado, a las realidades que permitan las personas, patrones y organización de la Empresa, tecnologías y herramientas en uso.

El diseño de software en la actualidad plantea un reto para el desarrollador, en razón de que debido a las múltiples metodologías y tipos existentes es necesario tener claro los conceptos y adicional a ello saber cómo funciona, de tal manera que la selección sea la más adecuada para la finalidad del proyecto.

La confiabilidad y robustez del motor de la base de datos Informix demostrada en el uso de las aplicaciones (15 años) que se ejecutan actualmente en la Empresa Importadora Comercial El Hierro han permitido continuar con la implementación y desarrollo de un nuevo módulo de programación que se integre y adapte a las necesidades de mejorar el control de vacaciones planteados en este proyecto. Por ello en el presente capítulo se describe con detalle la metodología de desarrollo de software, sus características básicas, ciclo de vida y los artefactos a ser usados en la solución de software, de igual forma se describe las potencialidades de la base de datos y su lenguaje de programación 4GL como herramienta para la implementación de la aplicación de control de vacaciones, adicional a ello se realiza el análisis para la selección de la arquitectura de software y el modelo de tres capas, lo cual considero importante para el desarrollo del proyecto de fin de titulación.

1.2. Arquitectura de información.

El término Arquitectura de Información (AI) según los autores (Rosenfield y Morville, 1999), (Montes de Oca, 2004) y (Ronda, 2008) se difunde y se aplica cada vez con más énfasis por

parte de las Empresas desarrolladoras de software en todo el mundo, evidenciando la creciente preocupación por que los sistemas informáticos sean usables y accesibles¹.

La AI es una disciplina íntimamente relacionada con la usabilidad y la accesibilidad (Montes de Oca, 2004), (Hassan, Martín e Iazza, 2004), haciendo prácticamente imposible concebir una sin la otra. Si bien las técnicas y el desarrollo de cada actividad son diferentes, comparten un mismo objetivo: que la interacción de los usuarios con el sistema sea satisfactoria.

La arquitectura de información, cuyo objetivo es la permanente búsqueda de mejores diseños para la presentación de la información y su comprensión, así como la usabilidad, que estudia el conjunto de características del diseño y funcionamiento de una interfaz de usuario, con el propósito de obtener una correcta operación y comprensión de los contenidos, son disciplinas cuya actividad está orientada a lograr la satisfacción total del usuario durante el proceso de interacción con los productos de información. Una información estructurada y coherente, sin dudas, facilita, tanto su consulta como el proceso de asimilación e introducción en la práctica. Ambas disciplinas surgieron y, a menudo, se estudian por separado. Sin embargo, son complementarias: los primeros (los contenidos) son la columna vertebral del producto de información y la segunda, (la interfaz) es el medio por donde se llega a ellos.

En 1989, Tim Berners-Lee², un especialista del Laboratorio Europeo de Física de Partículas (CERN), desarrolló un conjunto de especificaciones para facilitar el acceso a sus bases de datos. Estas se denominaron protocolo de transferencia de hipertexto (HTTP) y permitía acceder a documentos que contenían enlaces con otros documentos, ello simplificó la localización de información. A partir de este protocolo, nació World Wide Web (WWW), y se popularizó finalmente Internet, debido a la variedad y versatilidad de las páginas, aunque sólo se trate de uno de sus servicios.

La deficiente aplicación de la AI en general, y su no vinculación al proceso de desarrollo de software en particular, impacta negativamente en la experiencia del usuario. Por estas razones, en el presente trabajo de fin de titulación se propone un proceso para vincular la Arquitectura de Información a las distintas fases de desarrollo de software.

¹ http://nosolousabilidad.com/articulos/ai_rup.htm#biblio

² http://es.wikipedia.org/wiki/Tim_Berners-Lee

1.2.1. Fase de inicio.

La fase de inicio del proceso de AI se desarrollará en esta primera etapa, estudiando todo el material disponible asociado al proyecto de fin de titulación, de manera que se defina claramente su alcance y se asegure que tanto la Empresa como el desarrollador del producto compartan los mismos conceptos, principios, objetivos y estrategias para conseguirlos. Se realiza un estudio pormenorizado de los usuarios potenciales que tendrá la aplicación, con el fin de conocer sus características, necesidades y alcances; para ello es necesario clasificar y organizar los contenidos de información que se usarán; para lo cual se aplican técnicas de recopilación de información como la entrevista, la encuesta y otras técnicas cuantitativas. Concretamente en esta fase centramos el mayor esfuerzo en el desarrollo del conjunto de actividades y artefactos representados en la Figura 1.

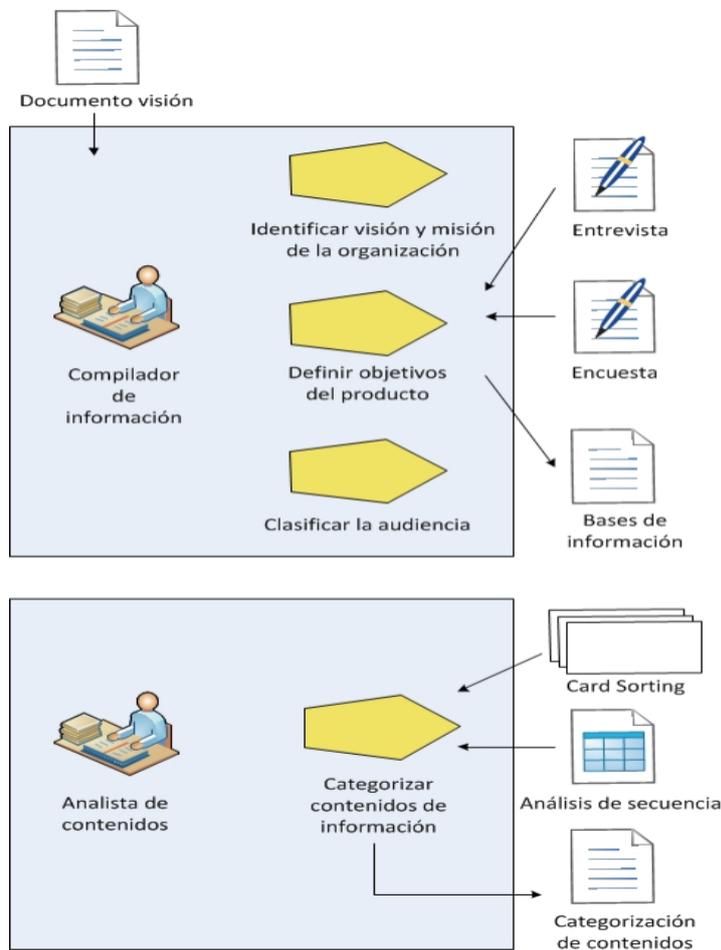


Figura 1. Relación entre actividades, trabajadores y artefactos de la Fase de Inicio del proceso de AI.

Fuente: (Sablón Fernández, Yusel; Hernández Aballe, Denny 2013).

Elaborado por: Lloret Antonio.

1.2.2. Fase de consolidación.

La AI efectúa su fase de consolidación, especificando la estructuración de los contenidos de información e introduciéndose en el diseño inicial del producto. En esta fase se hace la definición general o de alto nivel de la AI. Se asignan los nombres o etiquetas más acertados y relacionados entre sí a cada sección, conformando el esquema de contenidos definitivo. Se definen las directrices generales de la estructura y organización de la información.

Se definirán además las normas de representación de la información, y finalmente se ubicarán los elementos de forma gráfica (Ronda, 2004) en las pantallas de la aplicación. En esta fase de la AI se enfocara el trabajo de fin de titulación a elaborar el conjunto de actividades y artefactos que se representan en la Figura 2.

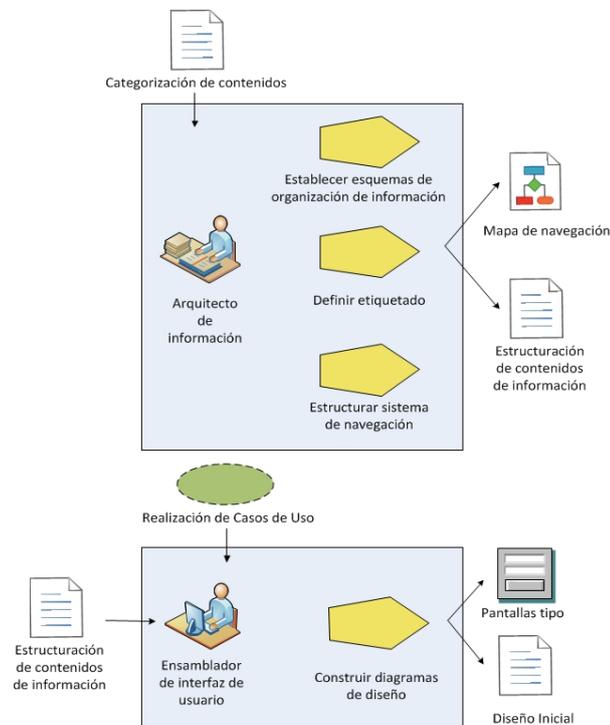


Figura 2. Relación entre actividades, empleados y artefactos de la fase de consolidación del proceso de AI.

Fuente: (Sablón Fernández, Yusel; Hernández Aballe, Denny 2013).
Elaborado por: Lloret Antonio.

1.2.3. Fase de validación de la AI.

Terminado todo el trabajo de arquitectura de información, una vez que desarrollaremos el primer prototipo del proyecto -generalmente un prototipo funcional previo al producto final-, éste es sometido a una revisión minuciosa, para evaluar que efectivamente cumplirá con las expectativas creadas. El equipo de fin de titulación revisa si efectivamente se están cumpliendo los objetivos, haciendo para ello pruebas con usuarios que permiten, además, anticipar el comportamiento real del producto y el nivel de satisfacción del usuario.

En esta fase se implementara las actividades y los artefactos que se muestra en la Figura 3.

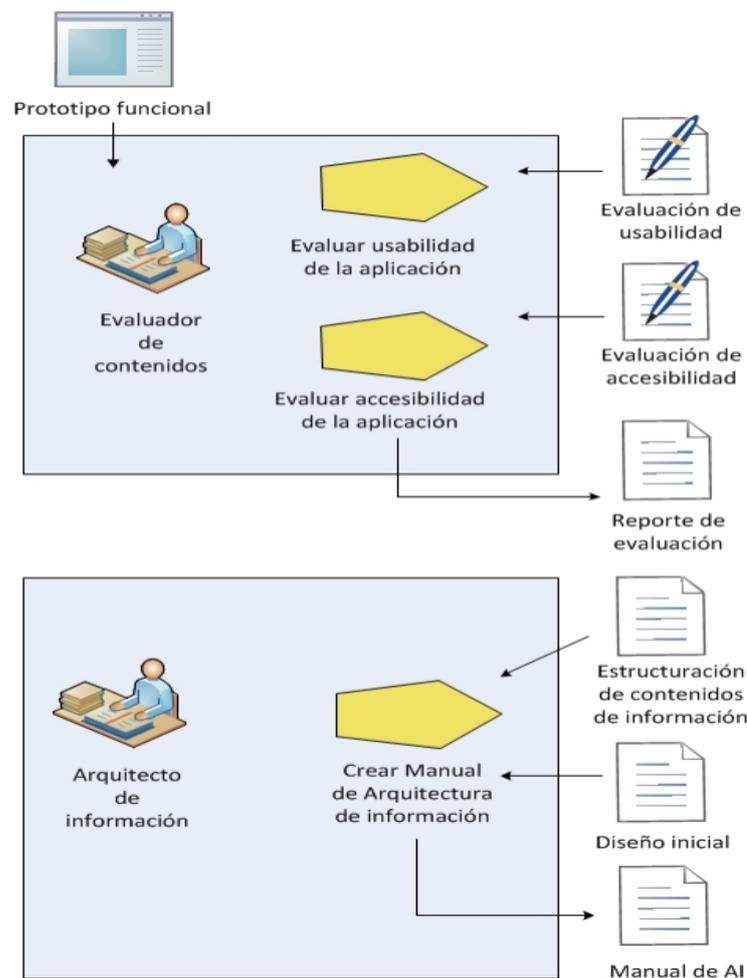


Figura 3. Relación entre actividades, empleados y artefactos de la fase de validación del proceso de AI.

Fuente: (Sablón Fernández, Yusel; Hernández Aballe, Denny 2013).

Para efectos del proyecto y con la finalidad de concentrar el máximo esfuerzo en el módulo de integración de la solución software se utilizara las disciplinas necesarias que faciliten el rápido desarrollo de la solución planteada y que a continuación se presenta un resumen de cada una de ellas:

Requisitos: Se refiere básicamente a detallar los requerimientos, indicar los límites y realizar una estimación de costo y tiempo empleado en el desarrollo de la solución software.

Análisis y Diseño: se refiere al uso de los artefactos de la metodología de desarrollo de software RUP, con el propósito de definir la arquitectura de la solución software, lo cual implica transferir los requisitos en especificaciones de implementación.

Implementación: Su objetivo consiste en implementar el diseño, evaluar las funcionalidades, integrar los componentes y utilizar un modelo de implementación para comprender el entorno global de la solución software.

Pruebas: Consiste en verificar la integración de las funcionalidades de la solución software, adicional a ello comprobar que todos los requisitos han sido implementados y asegurar que los defectos detectados han sido resueltos.

Despliegue: Confirma que la solución software esta lista para el uso del cliente, empaquetarlo, configurarlo, instalarlo y proceder a su entrega y recepción.

1.3. Metodología RUP.

1.3.1. Definiciones de RUP.

“Las siglas RUP en inglés significan Rational Unified Process, un producto del proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de alta calidad que resuelve las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos”, según el autor (Rueda Chacón, 2006).

RUP define la gestión de proyectos de software como el arte de balancear los objetivos, administrar riesgos y superar restricciones para entregar un producto que cumpla con las necesidades de los clientes y usuarios, según manifiesta el autor (Charbonneau S, 2004).

El **Proceso Racional Unificado** es un proceso de desarrollo de software desarrollado por la empresa Rational Software, actualmente propiedad de IBM. Junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. También se conoce por este nombre al software, también desarrollado por Rational, que incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. Está incluido en el **Rational Method Composer** (RMC), que permite la personalización de acuerdo con las necesidades.³

El RUP se fundamenta en seis mejores prácticas: desarrollo iterativo, administración de requerimientos, uso de arquitectura basada en componentes, modelamiento visual, verificación continua de la calidad y administración del cambio. Estas seis mejores prácticas orientan el modelo y con ellas se pretende solucionar muchos de los problemas asociados al software.

La implementación de estas mejores prácticas han sido probadas con éxito y dado que actualmente las organizaciones poseen y demandan cada vez más sofisticados sistemas de software, no es posible definir secuencialmente todo el problema que se plantea, el diseño de toda la solución, construir el software y probar el producto al final, por tal motivo se requiere trabajar con un enfoque iterativo cuya ventaja se da en una mayor comprensión del problema a través de sucesivas mejoras, y un crecimiento incremental de una solución cada vez mayor a través de múltiples iteraciones⁴.

Dada la eficiencia y flexibilidad en lo que se refiere a plazos y presupuesto, muchos desarrolladores de software han optado por aplicar esta metodología en base a su experiencia con el cliente, por lo cual considero que esta metodología es perfectamente aplicable para garantizar tanto a la Empresa como al autor del proyecto, la oportuna entrega y culminación del aplicativo.

1.3.2. Dimensiones de RUP.

En la figura 4 mostrada a continuación se puede observar la relación existente entre las fases del proceso propuesto para desarrollar la AI con los flujos de trabajo de la metodología RUP, en dependencia de las actividades que se llevan a cabo en cada uno de estos.

³ Tomado de http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational

⁴ Tomado de <http://www.iis.org/CDs2008/CD2009CSC/CISCI2009/PapersPdf/C690MI.Pdf>

De esta manera según el autor (Vargas Cano, 2014) El RUP tiene dos dimensiones:

- El eje horizontal representa tiempo y demuestra los aspectos del ciclo de vida del proceso.
- El eje vertical representa las disciplinas, que agrupan actividades definidas lógicamente por la naturaleza.

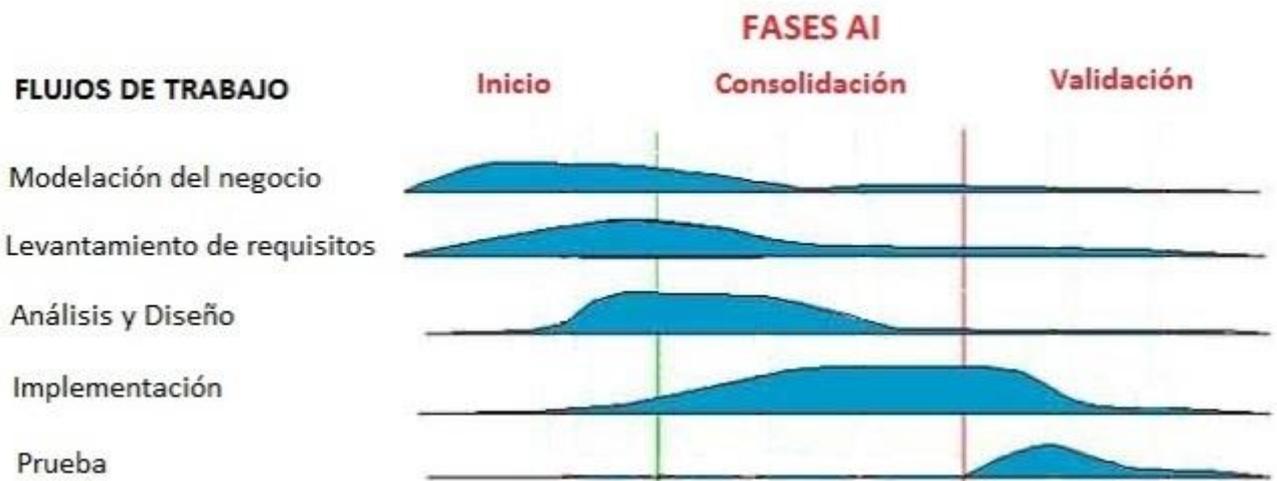


Figura 4. Relación entre las fases del proceso de AI y los flujos de trabajo de RUP.

Fuente: (Sablón Fernández, Yusel; Hernández Aballe, Denny 2013).

Elaborado por: Lloret Antonio.

1.3.3. Características fundamentales de RUP.

El conocer todo el proyecto implica considerar una documentación detallada que dentro de la metodología de desarrollo de software RUP se fundamenta en la definición de procesos, tareas y herramientas, además se caracteriza por ser interactiva e incremental y se guía por medio de los denominados casos de uso. Según manifiesta el autor (Vargas Cano, 2014) en el cual explica las siguientes características fundamentales de la metodología RUP:

- **Casos de uso:** son una técnica de captura de requisitos que involucra a todos los usuarios del proyecto, definiendo un caso de uso como un fragmento de funcionalidad del sistema y que representa los requisitos funcionales del sistema. Los casos de uso son utilizados como artefacto que permite establecer un comportamiento deseado del sistema, además de validar y verificar la arquitectura empleada en el sistema.

- **Proceso centrado en la arquitectura:** Los casos de uso deben encajar en la arquitectura cuando se llevan a cabo y la cual debe permitir el desarrollo de todos los casos de uso requerido. Esto provoca que tanto la arquitectura como los casos de uso deban evolucionar en paralelo durante todo el proceso de desarrollo de software, esto debe hacerse tomando en cuenta que la arquitectura es un artefacto utilizado para diseñar, construir, gestionar y evolucionar el desarrollo del sistema.
- **Proceso iterativo e incremental:** La estrategia que propone RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos, permitiendo el equilibrio entre los casos de uso y la arquitectura que se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede observar como una interacción que involucra pasar por los flujos de trabajo como requisitos, análisis, diseño, implementación y pruebas del cual se produce un crecimiento en el producto software.

Para efectos del presente proyecto se remitirá al uso de las características esenciales de casos de uso, como herramienta fundamental para especificar los requisitos funcionales de la solución software, lo cual permitirá aplicar en el diseño e implementación de la solución software; adicional a ello proporcionara una guía conductora durante la ejecución del proyecto y permitirá establecer trazabilidad entre los artefactos generados en las fases de la metodología de desarrollo. De igual manera nos apoyaremos en el proceso centrado en la arquitectura considerando las iteraciones necesarias en los casos de uso y las especificaciones de requerimientos del usuario, estos artefactos nos permitirá evolucionar en conjunto durante el proceso de desarrollo; también vale la pena mencionar la característica esencial del proceso iterativo e incremental que se usara para llevar adelante las interacciones necesarias que involucra a cada una de las disciplinas del ciclo de vida de la metodología de desarrollo de software, en otras palabras cumplir con los flujos fundamentales como requisitos, análisis y diseño, implementación, pruebas e integración, tal como se muestra en la figura 5; con el objetivo de obtener un producto software completamente desarrollado y listo para el uso de la Empresa Importadora Comercial El Hierro.



Figura 5. Interacción de RUP.
 Fuente: (Vargas Cano, 2014).
 Elaborado por: Lloret Antonio.

Como señala (Vargas Cano, 2014) cada interacción permitirá evolucionar la arquitectura y la implementación de los requisitos funcionales de la solución software.

En el caso concreto de este proyecto se desarrolla iteraciones en forma incremental hasta conseguir una solución software que tenga el visto bueno del cliente y cumpla con los requerimientos planteados para el desarrollo del proyecto control de vacaciones de la Empresa Imp. Com. El Hierro.

1.3.4. Artefactos de RUP.

Según el autor (Rueda Chacon, 2006) indica que “los artefactos son el resultado parcial o final que es producido y usado por los actores durante el proyecto. Son las entradas y salidas de las actividades realizadas por estos actores, los cuales utilizan y van produciendo estos artefactos para tener guías. Un artefacto puede ser un documento, un modelo o un elemento de modelo”.

1.3.4.1. Artefactos a desarrollar.

➤ **Fase de inicio:**

- Documento de visión (anexo D)
- Documento de especificación de requerimientos (anexo E)

- **Fase de elaboración:**
 - Documento de Especificación de casos de uso (anexo F)
 - Documento de arquitectura de software (anexo G)
- **Fase de construcción:**
 - Desarrollo de la solución software (Capítulo 3)
- **Fase de transición:**
 - Documento de plan de pruebas (anexo H)
 - Manual de usuario (anexo I)

Estos artefactos mencionados (ver Figura 6) son documentos que representan el principio y fin de las disciplinas del ciclo de vida del proceso de desarrollo de software, el cumplimiento del mismo se ejecutaran realizando las iteraciones que sean necesarias para garantizar el éxito de la solución software, adaptándose a la arquitectura de software seleccionada.

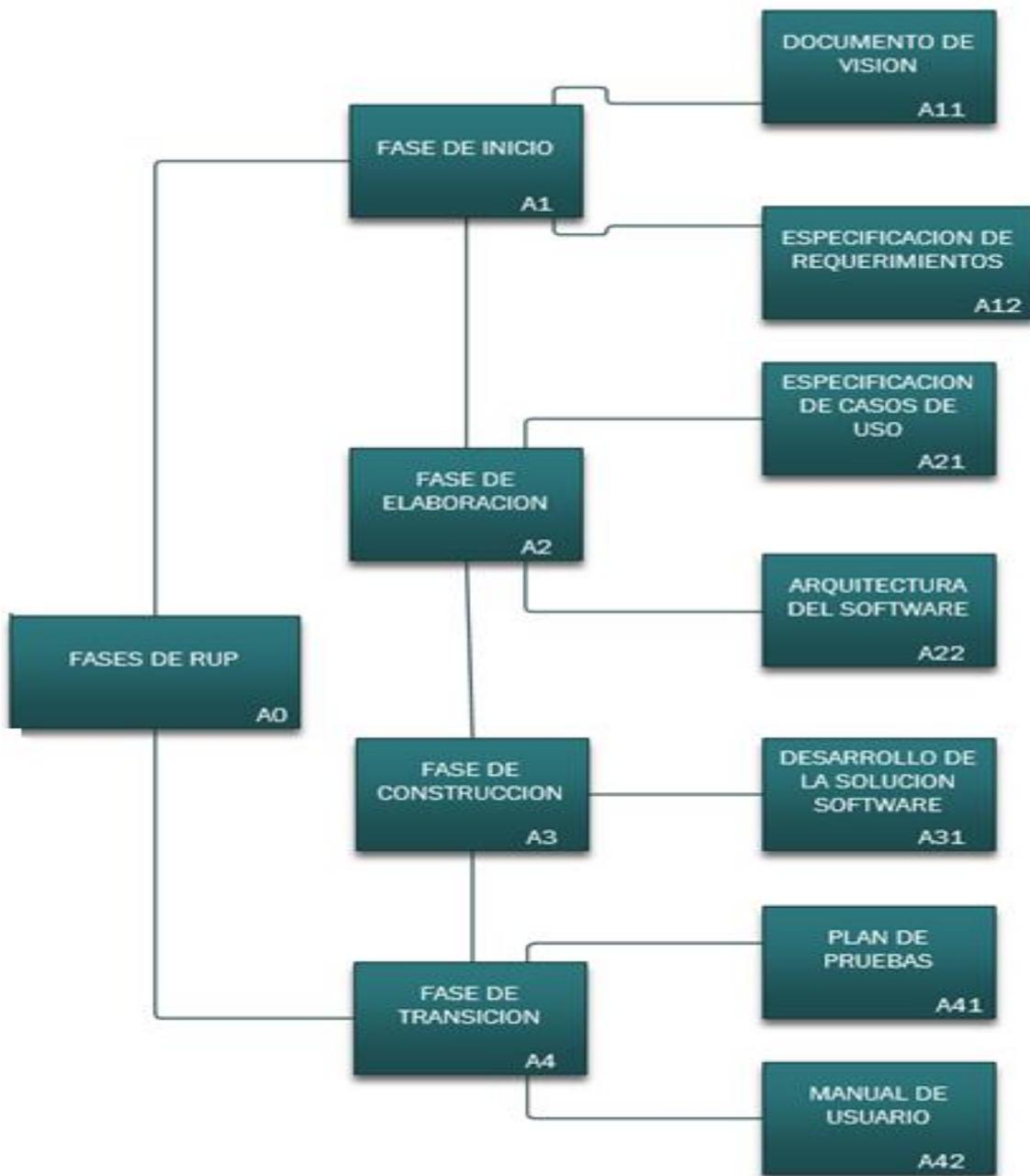


Figura 6. Fases y artefactos de RUP.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

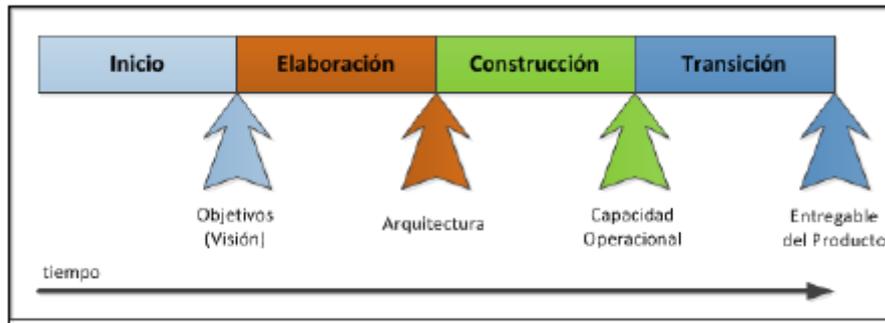


Figura 7. Fases de RUP.
Fuente: (Kruchten, 1996).
Elaborado por: Lloret Antonio.

1.3.5. Ciclo de vida de la metodología de desarrollo de software RUP.

El ciclo de vida del software del RUP se descompone en cuatro fases secuenciales. En cada extremo de una fase se realiza una evaluación (actividad: Revisión del ciclo de vida de la finalización de fase) para determinar si los objetivos de la fase se han cumplido. Una evaluación satisfactoria permite que el proyecto se mueva a la próxima fase.

Según el autor (Rueda Chacon, 2006) explica que "El ciclo de vida consiste en una serie de ciclos, cada uno de los cuales produce una nueva versión del producto, cada ciclo está compuesto por fases y cada una de estas fases está compuesta por un número de iteraciones, estas fases son:

1. Concepción, inicio o estudio de oportunidad

- Define el ámbito y objetivos del proyecto
- Se define la funcionalidad y capacidades del producto

2. Elaboración

- Tanto la funcionalidad como el dominio del problema se estudian en profundidad
- Se define una arquitectura básica
- Se planifica el proyecto considerando recursos disponibles

3. Construcción

- El producto se desarrolla a través de iteraciones donde cada iteración involucra tareas de análisis, diseño e implementación
- Las fases de estudio y análisis sólo dieron una arquitectura básica que es aquí refinada de manera incremental conforme se construye (se permiten cambios en la estructura)
- Gran parte del trabajo es programación y pruebas

- Se documenta tanto el sistema construido como el manejo del mismo
- Esta fase proporciona un producto construido junto con la documentación.

4. Transición

- Se libera el producto y se entrega al usuario para un uso real
- Se incluyen tareas de marketing, empaquetado atractivo, instalación, configuración, entrenamiento, soporte, mantenimiento, etc.
- Los manuales de usuario se completan y refinan con la información anterior
- Estas tareas se realizan también en iteraciones.

Todas las fases no son idénticas en términos de tiempo y esfuerzo. Aunque esto varía considerablemente dependiendo del proyecto, un ciclo de desarrollo inicial típico para un proyecto de tamaño mediano debe anticipar la distribución siguiente el esfuerzo y horario:

En el proyecto planteado se aplicará la utilización de las cuatro fases del ciclo de vida de RUP con la finalidad de obtener un producto de software de calidad que satisfaga los requerimientos de la Empresa; para lo cual se propone los artefactos a ser desarrollados en las cuatro fases del Proceso Unificado de Rational RUP como son: inicio, elaboración, construcción y transición empleados en la elaboración de la presente solución software.

1.4. Arquitectura de Software.

La arquitectura software se centra tanto en los elementos estructurales significativos del sistema, como subsistemas, clases, componentes y nodos, como en las colaboraciones que tienen lugar entre estos elementos a través de las interfaces.

Los casos de uso dirigen la arquitectura para hacer que el sistema proporcione la funcionalidad y uso deseados, alcanzando a la vez objetivos de rendimiento razonables. Una arquitectura debe ser completa, pero también debe ser suficientemente flexible como para incorporar nuevas funciones, y debe soportar la reutilización del software existente, según los autores (I. Jacobson, 2000).

Así mismo los autores (I. Jacobson, 2000)⁵ describen que la arquitectura se desarrolla de forma iterativa durante la fase de elaboración pasando por los requisitos, el análisis, el diseño, la implementación y las pruebas. La descripción de la arquitectura es una vista de los modelos

⁵ <http://www.freelibros.org/programacion/el-proceso-unificado-de-desarrollo-de-software-ivar-jacobson-grady-ooch-james-rumbaugh.html>

del sistema; de los modelos de casos de uso, análisis, diseño, implementación y despliegue. La descripción de la arquitectura describe las partes del sistema que es importante que comprendan todos los desarrolladores y otros interesados.

1.4.1. Importancia de la arquitectura de software.

Como lo manifiesta el autor Scott Ambler “La arquitectura es demasiado importante para dejarla en manos de una sola persona, no importa cual brillante sea.”.⁶ De igual manera Grady Booch define⁷ "Toda la arquitectura es diseño, pero no todo el diseño es arquitectura. La arquitectura representa las decisiones de diseño significativas que le dan forma a un sistema, donde lo significativo puede ser medido por el costo del cambio". Que considero como la definición más directa y simple de arquitectura.

En un libro dedicado a la arquitectura del software, (Bass et al, 2005)⁸ identifican tres razones clave por las que es importante la arquitectura del software:

- Las representaciones de la arquitectura del software permiten la comunicación entre todas las partes interesadas en el desarrollo de un sistema basado en computadora.
- La arquitectura resalta las primeras decisiones que tendrán un efecto profundo en todo el trabajo de ingeniería de software siguiente y, también importante, en el éxito último del sistema como entidad operacional.
- La arquitectura “constituye un modelo relativamente pequeño y asequible por la vía intelectual sobre cómo está estructurado el sistema y la forma en la que sus componentes trabajan juntos”.

El modelo del diseño de la arquitectura y los patrones arquitectónicos contenidos dentro de este son transferibles. Es decir, los géneros, estilos y patrones arquitectónicos pueden aplicarse al diseño de otros sistemas y representan un conjunto de abstracciones que permite a los ingenieros de software describir la arquitectura en formas predecibles.

⁶ http://en.wikipedia.org/wiki/Scott_Ambler

⁷ Grady Booch es un ingeniero de software estadounidense. Booch es el más conocido para el desarrollo del Lenguaje Unificado de Modelado con Ivar Jacobson y James Rumbaugh.

⁸ <http://www.freelibros.org/tag/ingenieria-del-software-7ma-edicion-roger-s-pressman>

1.4.2. Factores que influyen en la arquitectura de software.

Tal como lo mencionan los autores (Ivar Jacobson, Grady Booch, James Rumbaugh, 2000), “La arquitectura está condicionada por los casos de uso que queremos que soporte el sistema; los casos de uso son directores de la arquitectura”. Sin embargo, la arquitectura no solo se ve condicionada por los casos de uso arquitectónicamente significativos, sino también por los siguientes factores:

- Sobre qué productos software del sistema queremos desarrollar, como sistemas operativos o sistemas de gestión de bases de datos concretos.
- Que productos de middleware (capa intermedia que ofrece bloques de construcción reutilizables) como por ejemplo un marco de trabajo independiente de la plataforma, es decir un subsistema “prefabricado” para construir interfaces gráficas.
- Que sistemas heredados queremos utilizar en nuestro sistema. La utilización en nuestra arquitectura de un sistema heredado, como por ejemplo un sistema de nómina existente, nos permite reutilizar gran parte de la funcionalidad existente, pero también tenemos que adaptar nuestra arquitectura para que encaje con el producto “antiguo”.
- A que estándares y políticas corporativas debemos adaptarnos.
- Requisitos no funcionales generales (no específicos en los casos de uso) como son los requisitos de disponibilidad, tiempo de recuperación, o uso de memoria.
- Las necesidades de distribución, especificar como distribuir el sistema, quizá a través de una arquitectura cliente-servidor”.

1.4.3. Patrones arquitectónicos.

A medida que se desarrolle el modelo de requerimientos, se podrá observar que el software debe enfrentar cierto número de problemas que abarcan toda la aplicación en su conjunto, más aun cuando estamos integrando un módulo de programación adicional que se adapte e integre al sistema principal de la Empresa. Los patrones arquitectónicos se ven abocados a un problema de aplicación específica dentro de un contexto dado y sujeto a limitaciones y restricciones. El patrón propone una solución arquitectónica que sirve como base para el diseño de la arquitectura.

Los patrones arquitectónicos describen problemas de diseño amplios que se resuelven con un enfoque estructural. Una estructura proporciona el marco en el que residen los patrones y

los idiomas describen los detalles de implementación específicos del lenguaje de programación para un algoritmo, o parte de él, o para una estructura de datos específica.

Según el autor (Coplien, 2005)⁹ caracteriza un patrón de diseño eficaz del modo siguiente:

- El patrón tiene un componente humano significativo (minimiza la intervención humana). Todo el software sirve para el confort humano o la calidad de vida. Los mejores patrones recurren explícitamente a la estética y a la utilidad.

Dicho en otras palabras, un buen patrón de diseño incorpora el conocimiento de diseño pragmático, ganado con dificultad, de tal forma que otros lo reutilicen **“un millón de veces sin elaborarla dos veces de la misma forma”**. Un patrón de diseño evita **“reinventar la rueda”**, o, peor aún, inventar una **“nueva rueda”**. En conclusión si los patrones de diseño se usan con eficacia harán del programador un buen diseñador de software como acertadamente lo explica el autor (Coplien, 2005).

1.4.4. Estilos arquitectónicos.

El estilo arquitectónico se considera también como una plantilla para la construcción, como lo dice G.K. Chesterton “En el fondo de la mente de todo artista hay un patrón o tipo de arquitectura”. El software construido para sistemas basados en computadora también dispone de muchos estilos arquitectónicos. Cada estilo describe una categoría de sistemas que incluye:

- Un conjunto de componentes (como una base de datos o módulos de cómputo) que realizan una función requerida por el sistema,
- Un conjunto de conectores que permiten la “comunicación, coordinación y cooperación” entre los componentes,
- Restricciones que definen como se integran los componentes para formar el sistema.
- Modelos semánticos que permiten que un diseñador entienda las propiedades generales del sistema al analizar las propiedades conocidas de sus partes constituyentes (Bass, 2003).

⁹ http://www.amazon.com/James-O.-Coplien/e/B000APJRP4/ref=dp_byline_cont_pop_book_1

Un estilo arquitectónico sostiene el autor (Bass, 2003) es una transformación que se impone al diseño de todo el sistema. El objetivo es establecer una estructura para todos los componentes del sistema. En el caso en el que ha de hacerse la reingeniería de una arquitectura ya existente, la imposición de un estilo arquitectónico dará como resultado cambios fundamentales en la estructura del software, incluida la reasignación de las funciones de los componentes. En la Tabla 1 se detalla un cuadro de los principales categorías en estilos arquitectónicos de software.

Tabla 1. Estilos de arquitectura de software.

CATEGORÍA	ESTILOS DE ARQUITECTURA
Comunicación	Arquitectura orientada a servicios (SOA), bus de mensajes
Despliegue	Arquitectura Cliente-Servidor, N-Niveles, 3-Niveles
Dominio	Dominio de Diseño
Estructura	Arquitectura basada en componentes, orientada a objetos, arquitectura en capas.

Fuente: (Conery, Rob, Hanselman, Scott; Phil, Haack; Guthrie, Scott, 2009).
Elaborado por: Lloret Antonio.

En la Tabla 2 se describe un resumen de los principales estilos arquitectónicos.

Tabla 2. Descripción de los estilos arquitectónicos.

ESTILO DE ARQUITECTURA	DESCRIPCIÓN
Cliente/Servidor	Divide el sistema en dos particiones, en que el cliente realiza las solicitudes al servidor. En muchos casos el servidor es una base de datos con la lógica de la aplicación representando como procedimientos almacenados.
Arquitectura basada en componentes	Descompone el diseño de la aplicación en funciones reusables o componentes lógicos, que establecen interfaces de comunicación bien definidos.
Arquitectura en capas	Particiones de la aplicación en grupos aislados (capas).
Bus de mensaje	Es un estilo de arquitectura que describe el uso de un sistema de software que puede recibir y enviar mensajes a través de uno o varios canales de comunicación, por lo que las aplicaciones pueden interactuar sin necesidad de conocer los detalles específicos entre sí.
N-niveles / 3-niveles	Divide funcionalidad en segmentos separados de la misma manera que el estilo de arquitectura en capas, pero con cada nivel siendo situado en un equipo físicamente independiente.
Orientada a objetos	Un paradigma, diseño basado en la división de responsabilidades para una aplicación o sistemas en objetos reutilizables y autosuficientes, cada uno con

	los datos y el comportamiento relevante para el objeto.
Arquitectura orientada a servicios (SOA)	Se refiere a las aplicaciones que exponen y consumen la funcionalidad como un servicio mediante el paso de mensajes o protocolos de comunicación.

Fuente: (Conery, Rob, Hanselman, Scott; Phil, Haack; Guthrie, Scott, 2009).
Elaborado por: Lloret Antonio.

Para el desarrollo del presente proyecto es importante tener en consideración la infraestructura de TI que dispone la Empresa, niveles de seguridad de redes de comunicación, sistema operativo instalados en sus servidores, tecnologías implementadas, hardware instalado, etc. De tal manera que permitan aprovechar en su totalidad un estilo de arquitectura de software que brinde confiabilidad en el funcionamiento y desempeño requerido de la solución software a integrar.

1.4.4.1. Estilo Cliente-Servidor.

El concepto de arquitectura cliente-servidor se refiere a la forma en que los componentes software interactúan. Según manifiesta los autores (Connolly, T. M.; Begg, C.E., 2007) existe un proceso cliente que requiere algún tipo de recurso y un servidor que proporciona el recurso. En un artículo de discusión en la página web de Wikipedia se manifiesta que “La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.”¹⁰. En el modelo en dos niveles Figura 9, el cliente gestiona la interfaz del usuario y la lógica de procesamiento del negocio y el servidor gestiona la funcionalidad de la base de datos.

¹⁰ Tomado de: <http://es.wikipedia.org/wiki/Cliente-servidor>

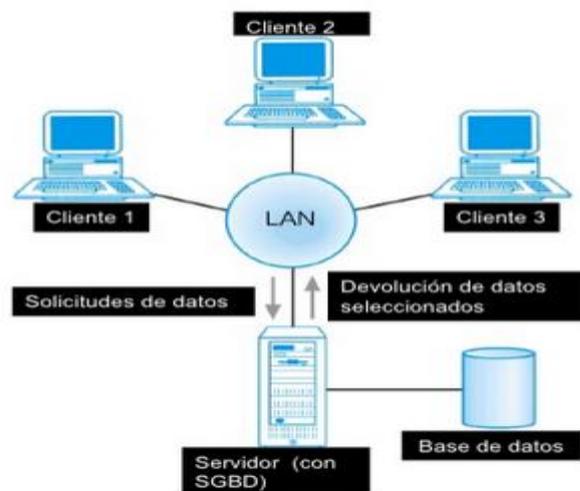


Figura 9. Arquitectura cliente-servidor.
Fuente: (Connolly, T. M.; Begg, C.E., 2007).
Elaborado por: Lloret Antonio.

No es obligatorio que el cliente y el servidor residan en la misma maquina; es muy común observar en la práctica ubicar al servidor en una ubicación específica dentro de la red de área local, así como a los clientes en las ubicaciones sobrantes. En el caso específico de la Empresa Imp. Com. El Hierro las aplicaciones comerciales con requisitos intensivos de datos están formadas por cuatro componentes principales: la base de datos, la lógica de las transacciones, la lógica de negocios y de la aplicación de gestión de datos y la interfaz de usuarios.

Según los autores (Connolly, T. M.; Begg, C.E., 2007) este tipo de arquitectura tiene muchas ventajas, como por ejemplo:

- Permite un acceso más universal a las bases de datos existentes.
- Mejores prestaciones: si los clientes y el servidor en computadoras distintas, las diferentes UCP podrán procesar distintas aplicaciones en paralelo. También resulta más fácil optimizar la maquina servidora si su única tarea consiste en realizar el procesamiento relacionado con la base de datos.
- Pueden reducirse los costos de hardware: solo el servidor requiere el suficiente espacio de almacenamiento y la suficiente capacidad de proceso como para almacenar y gestionar la base de datos.
- Se reducen los costos de comunicaciones: las aplicaciones llevan a cabo parte de las operaciones en el cliente y solo envían a través de la red las solicitudes de acceso a la base de datos, con lo que se envían menos datos a través de la red.

- Mayor coherencia: el servidor puede gestionar las comprobaciones de identidad, por lo que solo es necesario definir y validar las restricciones en un único lugar, en lugar de hacer que cada programa de aplicación lleve a cabo sus propias comprobaciones.
- Esta arquitectura puede ajustarse de forma bastante natural a la arquitectura de los sistemas abiertos.

1.4.4.2. Modelo de programación por capas.

En el diseño de sistemas informáticos actual se suelen usar las arquitecturas multinivel o programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).

En el caso de la Empresa Imp.Com. El Hierro en donde ya existe un sistema implementado por capas, se ha podido por medio de este proyecto de titulación generar un nuevo módulo de control de vacaciones que se incorpore y adapte con facilidad al sistema general de la Empresa.

La programación por capas es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, solo se ataca al nivel requerido sin tener que revisar entre código mezclado; un buen ejemplo de este método de programación sería el modelo de interconexión de sistemas abiertos; además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.

El más utilizado actualmente es el diseño en tres niveles (o en tres capas) y que en el caso específico de la solución software a aplicar en la Empresa Imp. Com. El Hierro es el que se describe a continuación (ver Figura 10).

1. **Capa de presentación:** la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay

errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.

2. **Capa de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.
3. **Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

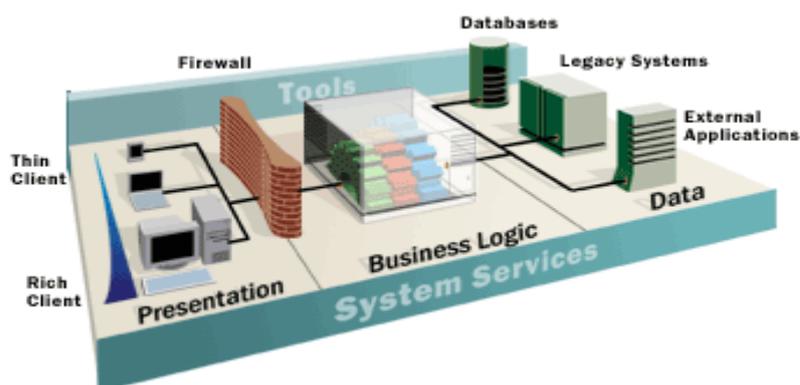


Figura 10. Arquitectura en tres capas.

Fuente: (<http://www.forosdelweb.com/f14/modelo-3-capas-arquitectura-n-capas-122898>).
Elaborado por: Lloret Antonio.

Todas estas capas pueden residir en un único ordenador, si bien lo más usual es que haya una multitud de ordenadores en donde reside la capa de presentación (son los clientes de la arquitectura cliente-servidor). Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio.

Si, por el contrario, fuese la complejidad en la capa de negocio lo que obligase a la separación, esta capa de negocio podría residir en uno o más ordenadores que realizarían solicitudes a una única base de datos. En sistemas muy complejos se llega a tener una serie de ordenadores sobre los cuales corre la capa de negocio, y otra serie de ordenadores sobre los cuales corre la base de datos. En el caso específico de la Empresa Imp. Com. El Hierro existen varios servidores en donde reside la capa de negocio; y de igual manera varios servidores donde se ejecuta la base de datos; esto con el fin de manejar un acceso más eficiente y seguro del gran volumen transaccional que soporta el sistema; además es importante considerar que la matriz se interconecta con los otras sucursales por medio de un procedimiento de replicación de datos, lo cual permite una independencia de procesamiento de datos con el propósito de agilizar sus recursos.

En una arquitectura de tres niveles, los términos "capas" y "niveles" no significan lo mismo ni son similares. El término "capa" hace referencia a la forma como una solución es segmentada desde el punto de vista lógico:

- **Presentación.** (Conocida como capa Web en aplicaciones Web o como capa de usuario en Aplicaciones Nativas)
- **Lógica de Negocio.** (Conocida como capa Aplicativa)
- **Datos.** (Conocida como capa de Base de Datos)

En cambio, el término "nivel" corresponde a la forma en que las capas lógicas se encuentran distribuidas de forma física. Por ejemplo:

- Una solución de tres capas (presentación, lógica del negocio, datos) que residen en un solo ordenador (Presentación+lógica+datos). Se dice que la arquitectura de la solución es de tres capas y *un nivel*¹¹

Una solución de tres capas (presentación, lógica del negocio, datos) que residen en dos ordenadores (presentación+lógica por un lado; lógica+datos por el otro lado). Se dice que la arquitectura de la solución es de tres capas y *dos niveles*¹²

¹¹ Programación por capas tomado de:http://es.wikipedia.org/w/index.php?title=Programaci%C3%B3n_por_capas&action=info

¹² Programación por capas tomado de http://es.wikipedia.org/w/index.php?title=Programaci%C3%B3n_por_capas&action=info

1.4.4.3. Arquitectura Cliente-Servidor en tres niveles.

Como lo comentan los autores (Connolly, T. M.; Begg, C.E., 2007) en su libro Sistemas de Bases de Datos, a mediados de la década de 1990, a medida que las aplicaciones fueron creciendo en complejidad y la necesidad de mejorar la escalabilidad de los sistemas empresariales, además de poder implementarse en centenares o miles de clientes, aparece en 1995 una nueva variación del modelo tradicional cliente-servidor en dos niveles, para intentar resolver los problemas de escalabilidad en las empresa. Esta nueva arquitectura proponía tres niveles, cada una de las cuales puede ejecutarse en una plataforma distinta:

1. El nivel de interfaz de usuario, que se ejecuta en la computadora del usuario final (el cliente).
2. El nivel de lógica de negocio y procesamiento de datos. Este nivel intermedio se ejecuta en un servidor que a menudo se denomina servidor de aplicaciones.
3. Un SGBD, que almacena los datos requeridos por el nivel intermedio. Este nivel puede ejecutarse en un servidor independiente, denominado servidor de base de datos.

Como se aprecia en la Figura 11, el cliente solo es ahora responsable de la interfaz de usuario de aplicación y, quizás, de realizar algún tipo de procesamiento lógico simple, como por ejemplo la validación de los datos de entrada, por lo que esta arquitectura se dispone de lo que se denomina clientes "simples". La lógica del negocio principal de la aplicación reside ahora en su propio nivel, que se conecta físicamente al cliente y el servidor de la base de datos a través una red de área local LAN. Un mismo servidor de aplicaciones puede dar servicio a múltiples clientes.

Ventajas:

- El hardware necesario es menos costoso.
- El mantenimiento de las aplicaciones está centralizado. Al transferirse la lógica del negocio desde las plataformas de los usuarios finales a un único servidor de aplicaciones.
- Al ser mayor la modularidad, resulta más sencillo modificar o sustituir uno de los niveles sin que los otros sean afectados.
- Resulta más fácil equilibrar la carga de procesamiento al separar la lógica principal de negocio de las funciones de base de datos.

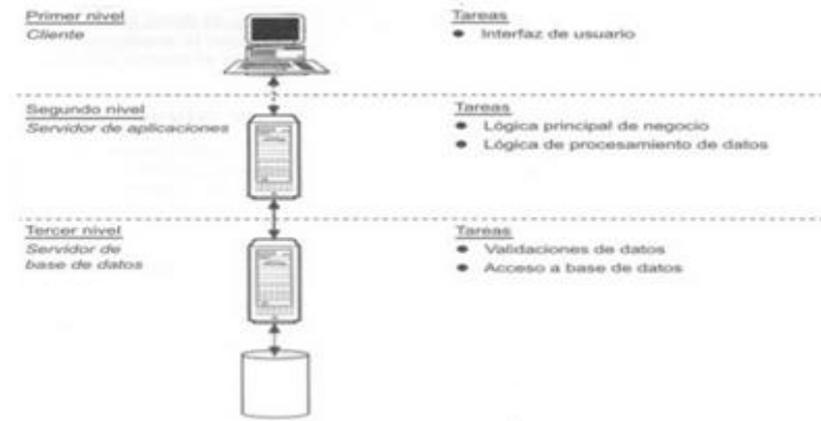


Figura 11. Arquitectura en tres niveles.
Fuente: (Connolly, T. M.; Begg, C.E., 2007).
Elaborado por: Lloret Antonio.

En lo que se refiere a nuestra solución de software, estamos hablando de una solución de tres capas y tres niveles, por cuanto la presentación, lógica del negocio y datos están ubicados en un solo ordenador; sin embargo vale la pena mencionar que la Empresa a través de su departamento de sistemas dispone de equipos adicionales para efectos de replicación de datos, seguridad y bloqueo de accesos no autorizados mediante la utilización de software anti robo y routers de control y filtrado.

Para efectos de la solución de software planteada, se considera todos los conceptos descritos anteriormente, y por ello se **consideró que la mejor opción que se adapta a las características del proyecto es el estilo arquitectónico Cliente-Servidor y Arquitectura en capas**; en función de sus ventajas en cuanto a replicación de datos, seguridad, acceso a los datos de forma centralizada, facilidades para su mantenimiento y soporte, funcionalidad, confiabilidad, rendimiento, escalabilidad, robustez y usabilidad.

Mayor detalle del desarrollo e implementación del presente proyecto aplicando el patrón arquitectónico cliente-servidor se especificara en el capítulo 3.

1.4.4.4. Cuadro resumen de los estilos arquitectónicos de software.

En el siguiente cuadro (Tabla. 3) se describe un resumen de los estilos de arquitectura que considero son los más apropiados para la estructura, desarrollo e implementación de la solución software planteada en este proyecto.

Tabla 3. Cuadro resumen de estilos de arquitectura de software.

CLIENTE/SERVIDOR	ARQUITECTURA EN CAPAS
Objetivo: El cliente solicita los recursos y servidor responde a la solicitud del cliente	Separación de la capa de presentación, capa de negocios y la capa de datos.
Componentes: <ul style="list-style-type: none"> ➤ Cliente ➤ Servidor 	<ul style="list-style-type: none"> ➤ Capa de presentación ➤ Capa de negocios ➤ Capa de datos
Característica Principal: El servidor se vuelve polivalente, al responder directamente a todas las solicitudes del cliente.	Permite distribuir la creación de una aplicación en niveles, en donde cada grupo de trabajo está totalmente abstraído del resto de niveles.
Ventajas: <ul style="list-style-type: none"> ➤ Mayor seguridad ➤ Acceso a datos centralizado ➤ Fácil mantenimiento ➤ Soporta muchos clientes 	<ul style="list-style-type: none"> ➤ Mantenibilidad ➤ Escalabilidad ➤ Flexibilidad ➤ Disponibilidad
Desventajas: <ul style="list-style-type: none"> ➤ Genera mayor tráfico en la red ➤ Carece de extensibilidad, escalabilidad y robustez. 	<ul style="list-style-type: none"> ➤ Dificultad en diseño de cada capa ➤ El procesamiento de una capa puede retardar el procesamiento de otras.

Fuente: (Conery, Rob, Hanselman, Scott; Phil, Haack; Guthrie, Scott, 2009).
Elaborado por: Lloret Antonio.

1.4.5. Selección del estilo arquitectónico de software.

Para la selección y valoración del estilo arquitectónico a adoptar para la solución de software se ha tomado en consideración en función de los atributos de calidad y en función como lo manifiesta Hewlett-Packard¹³ que desarrollo un conjunto de atributos de la calidad del software a los que se dio el acrónimo FURPS: funcionalidad, usabilidad, confiabilidad, rendimiento y mantenibilidad. Los atributos de calidad FURPS representan el objetivo de todo diseño de software. C.A.R. Hoare cita “Hay dos formas de construir un diseño de software. Una es

¹³ Grady. R. B. y D. L. Caswell, Software Metrics. Establishing a Company Wide Program, Prentice Hall, 1987.

hacerlo tan simple que sea obvio que no hay deficiencias y la otra es hacerla tan complicado que no haya deficiencias obvias. El primer método es mucho más difícil.”

- La **funcionabilidad** se califica de acuerdo con el conjunto de características y capacidades del programa, la generalidad de las funciones que se entregan y la seguridad general del sistema.
- La **consistencia** y la documentación.
- La **confiabilidad** se evalúa con la medición de la frecuencia y gravedad de las fallas de exactitud de los resultados que salen, el tiempo medio para que ocurra una falla, la capacidad de recuperación ante esta y lo predecible del programa.
- El **rendimiento** se mide con base en la velocidad de procesamiento, el tiempo de respuesta, el uso de recursos, el conjunto y la eficiencia.
- La **mantenibilidad** combina la capacidad del programa para ser ampliable (extensibilidad), adaptable y servicial, y además que pueda probarse, ser compatible y configurable (capacidad de organizar y controlar los elementos de la configuración de software y que cuente con la facilidad para instalarse en el sistema y para que se detecten los problemas.

Es importante observar que estos atributos de la calidad deben tomarse en cuenta cuando comienza el diseño, no cuando haya terminado este y la construcción se encuentre en marcha.

A continuación se presenta un cuadro valorativo (Tabla 4) de los estilos arquitectónicos basados en los atributos de calidad mencionados por Hewlett-Packard.

Tabla 4. Cuadro valorativo de los estilos arquitectónicos.

ESTILO DE ARQUITECTURAS	FUNCIONABILIDAD	USABILIDAD	CONFIABILIDAD	RENDIMIENTO	MANTENIBILIDAD
CLIENTE/SERVIDOR	OK	OK	OK	OK	OK
ARQUITECTURA EN CAPAS	OK	OK	OK	OK	OK

Fuente: (Lloret Orellana,2015).

Elaborado por: Lloret Antonio.

En función de la valoración de la Tabla 4, considero que para el caso del desarrollo del presente proyecto el estilo arquitectónico **cliente-servidor y el estilo arquitectura en capas** es el que mejor adaptabilidad tendría, en razón de que permite tener un alto grado de disponibilidad del sistema, basado en la distribución de capas que se realiza de forma efectiva la separación de funcionalidades, lo cual permite la disponibilidad en cualesquiera de sus capas. Otra ventaja de este estilo arquitectónico es su escalabilidad, que facilita una mejor distribución de las responsabilidades del sistema, lo cual implica mejor desempeño que se nota en el incremento de usuarios que acceden y ejecutan transacciones de manera simultánea. De igual manera este estilo de arquitectura maneja y distribuye cada capa en máquinas físicas o de ser el caso en una sola máquina, comunicándose con los componentes de otras capas por medio de interfaces bien logradas. Adicionalmente es flexible al aplicar cambios en cualquiera de sus capas, sin afectar el rendimiento del sistema al responder las solicitudes de los usuarios, esto permite ganar eficiencia en la distribución de recursos de hardware y software, aplicando seguridades en las capas que sean requeridas. Concluyendo podemos mencionar que en función de la alta disponibilidad, escalabilidad y flexibilidad de este estilo de arquitectura **cliente-servidor** y arquitectura en capas es la selección óptima para la solución software a implementar.

1.4.6. Lenguajes de cuarta generación (4GL).

Según manifiesta (Connolly T.M, Begg C.E, 2007) no existe consenso acerca de lo que constituye un lenguaje de cuarta generación, se trata, en esencia, de un lenguaje de programación optimizado. Una operación que requiera cientos de líneas de código en un lenguaje de tercera generación (3GL), como Cobol, requiere generalmente un número mucho menor de líneas en un 4GL.

Comparado con un 3GL, que es procedimental, un 4GL es no procedimental; el usuario define que hay que hacer, no el modo de hacerlo. Lo normal es que un 4GL dependa de modo fundamental de componentes de mucho mayor nivel, conocidos con el nombre de herramientas de cuarta generación. El usuario no define los pasos que un programa debe seguir para llevar a cabo una tarea, sino que define una serie de parámetros para las herramientas, las cuales los usan para generar un programa de aplicación. La opinión generalizada es que los 4GL pueden mejorar la productividad en un factor de 10, aunque a

cambio limiten los tipos de problemas que puedan abordarse. Los lenguajes de cuarta generación comprenden:

- Lenguajes de presentación
- Lenguajes especializados, como las hojas de cálculo y los lenguajes de base de datos;
- Generadores de aplicaciones que definen, insertan, actualizan y extraen datos e la base de datos para construir aplicaciones;
- Lenguajes de muy alto nivel que se emplean para generar códigos de aplicación.

Por ello es importante mencionar que en el caso de la Empresa Imp. Com. El Hierro todas las aplicaciones y programación de las mismas están desarrolladas con herramientas de cuarta generación, como es el caso de SQL y el lenguaje de programación de la base de datos Informix 4GL.

1.5. Informix.

1.5.1. Definiciones de Informix.

Informix es una familia de productos RDBMS de IBM, adquirida en 2001 a una compañía (también llamada Informix o Informix Software) cuyos orígenes se remontan a 1980.

El DBMS Informix fue concebido y diseñado por Roger Sippl a finales de los años 1970. La compañía Informix fue fundada en 1980, salió a bolsa en 1986 y durante parte de los años 1990 fue el segundo sistema de bases de datos más popular después de Oracle. Sin embargo, su éxito no duró mucho y para el año 2000 una serie de tropiezos en su gestión había debilitado seriamente a la compañía desde el punto de vista financiero. En 2001 IBM, impulsada por una sugerencia de Wal-Mart (el mayor cliente de Informix) compró Informix. IBM tenía planes a largo plazo tanto para Informix como para DB2, compartiendo ambas bases de datos tecnología de la otra. A principios de 2005, IBM lanzó la versión 10 del Informix Dynamic Server (IDS).

Volviendo su atención al mercado emergente de los [[Sistema administrador de bases de datos relacionales|RDBMS]], RDS lanzó su propio producto como "'Informix'" (de "'Inform'ation on Un'ix'", 'información en Unix') en 1981, incluyendo su propio lenguaje "'Informer'" y el editor de informes ACE, usado para extraer datos de la base de datos y

presentarlos a los usuarios de forma legible. También incluía la herramienta de formularios en pantalla PERFORM, que permitía a un usuario realizar consultas y editar los datos de la base de datos interactivamente.

En [[1985]] presentaron un nuevo motor de consultas basado en [[SQL]] como parte de su "INFORMIX-SQL" (o ISQL) versión 1.10 (la versión 1.00 no se llegó a lanzar). Este producto también incluía versiones SQL de ACE y PERFORM. La diferencia más importante entre ISQL y el anterior Informix era la separación del código de acceso a la base de datos en un motor independiente (sqlxec) en lugar de incrustarlo directamente en el cliente, preparando así el entorno a la llegada de la computación cliente-servidor con la base de datos corriendo en un máquina diferente a la del usuario final.

Durante principios de los años 1980 Informix siguió siendo una Empresa pequeña, pero a medida que Unix y SQL ganaban popularidad durante mediados de la década su suerte cambió. Para [[1986]] habían crecido lo suficiente como para salir a bolsa con éxito, y cambiaron el nombre de la compañía a "Informix Software". Sus productos eran INFORMIX-SQL versión 2.00 e INFORMIX-4GL 1.00, incluyendo ambos el motor de bases de datos y herramientas de desarrollo (I4GL para programadores, ISQL para no-programadores).

Siguieron una serie de lanzamientos, incluyendo un nuevo motor de consultas, inicialmente conocido como INFORMIX-Turbo. Turbo usaba el nuevo [[RSAM]], con grandes mejoras en el rendimiento con múltiples usuarios respecto a [[ISAM]]. Con el lanzamiento de la versión 4.00 de sus productos en 1989, Turbo fue rebautizado "INFORMIX-OnLine" (en parte porque permitía la realización de copias de respaldo coherentes de la base de datos mientras el servidor estaba en línea y los usuarios modificaban los datos) y el servidor original basado en C-ISAM fue separado de las herramientas (ISQL e I4GL) y llamado "INFORMIX-SE" ("Standard Engine", 'motor estándar'). La versión 5.00 de Informix OnLine fue lanzada a finales de 1990 e incluía soporte para transacciones completamente distribuidas con "commits" en dos fases y procedimientos almacenados. La versión 5.01 añadió soporte para [[Disparador (base de datos) |disparadores]], ([[Disparador (base de datos) |triggers]]).

Informix integró el [[mapeo objeto-relacional]] y los DaTablades de Illustra en la versión 7.x de su versión OnLine, lo que resultó en el "Informix Universal Server" (IUS), o más comúnmente, Versión 9.

Ambas versiones nuevas, V8 (XPS) y V9 (IUS), aparecieron en el mercado en [[1996]], haciendo a Informix la primera de las «tres grandes» compañías de bases de datos (siendo las otras dos Oracle y Sybase) en ofrecer soporte objeto-relacional integrado. Los expertos prestaron especial atención a los DaTablades, que pronto se hicieron muy populares:

aparecieron docenas en sólo un año, portados a la nueva arquitectura tras acuerdos de colaboración con Illustra.

A partir del año [[2000]], los principales hitos en la historia de Informix dejaron de centrarse en sus innovaciones técnicas. En mayo de ese año Informix compró [[Ardent Software]], un compañía que ya tenía su propia historia de fusiones y adquisiciones. Esta adquisición añadió los motores [[base de datos multidimensional|multidimensionales]] "UniVerse" y "UniData" (conocidos conjuntamente como "U2") a su ya por entonces extensa lista de motores de bases de datos, que incluía no sólo los productos históricos de Informix, sino también un motor SQL orientado a [[Almacén de datos|almacenes de datos]] de "Red Brick" y la versión 100% java de SQL, "Cloudscape" (que más tarde fue incorporado en la implementación de referencia de [[Java 2 Enterprise Edition|J2EE]]).

Informix es uno de los cuatro grandes de las bases de datos junto DB2 de IBM, SQL Server de Microsoft y Oracle (MarcadorDePosición1) como lo aseveran los autores (Torrealba Luis. Benites Jonny) ¹⁴.

SQL (Standar Query Lenguaje) es un lenguaje estandarizado de base de datos, el cual nos permite realizar Tablas y obtener datos de ella de manera muy sencilla. Para exponer más claramente los conceptos se realizaron ejemplo sobre relaciones que se crearan aquí para entender mejor como funciona SQL. También se puede decir, **SQL** es un lenguaje bastante sencillo, principalmente orientado a bases de datos y, sobre todo, al manejo de consultas. Visual Basic incorpora esta extensión junto a nuestras bases de datos, obteniendo potentes resultados. De hecho, las consultas que se realizan en Access, están desarrolladas o basadas en este lenguaje, por lo que su implementación en Visual Basic no es complicada.

El objetivo principal de SQL es la realización de consultas y cálculos con los datos de una o varias Tablas.

Desarrollador: IBM

Última versión: IBM Informix 12.10. xC2

Estable: 1 de octubre de 2013

Género: Sistema de gestión de bases de datos relacionales

Programado en C, C++

¹⁴ <http://sabd15n2.wikispaces.com/file/view/informix.pdf>

Sistema Operativo: Multiplataforma

Licencia: Propietaria

IBM Informix es un sistema de gestión de bases de datos relacionales disponible en plataformas Microsoft Windows, Linux, UNIX y Apple Mac OS X. El software Informix proporciona procesamiento de transacciones en línea y aplicaciones de soporte a la toma de decisiones para Empresas de todos los tamaños. Ofrece una plataforma de gestión de datos integrada, fiable y de bajo coste para proveedores de software independientes (ISV) y fabricantes de equipos originales (OEM).¹⁵

1.5.2. Informix 4GL.

Es el líder para manejar aplicaciones cliente que acceden a las bases de datos de Informix en el ambiente de Unix. Es una estructura basada en ventanas y caracteres orientados que permite a desarrolladores preparar o directamente incrustar en un código fuente de un lenguaje de cuarta generación determinado la estructura estándar de consultas SQL.

Desde el código fuente del 4GL el pre-procesador del Informix genera pseudo código (p-código) de máquina o código C dependiendo de cuál haya sido el 4GL elegido en tiempo de compilación.¹⁶

Para producir las aplicaciones se utilizan tres diferentes productos: el Compilador, el Sistema Rápido de Desarrollo y el Depurador Interactivo. El compilador genera código C y código ESQL/C que puede ser compilado en el código objeto, el sistema rápido de desarrollo reduce en gran manera el tiempo de compilación creando el pseudo código y el depurador ayuda a la detección y rápida corrección de errores dentro de la aplicación.

La gran utilidad del Informix lo han colocado como líder dentro del mercado para el desarrollo de aplicaciones para bases de datos, pero debido a que su lanzamiento fue en 1986 muchas aplicaciones creadas utilizándolo tal vez ya son muy viejas y necesitan adaptarse a las nuevas necesidades como la compatibilidad y uso de interfaces gráficas para el usuario.

Por lo anterior es que se ha desarrollado una nueva versión de Informix denominado Informix Dynamic Server 4GL que permite migrar hacia las nuevas necesidades del mercado con un simple recompilación del programa lo que facilita en gran medida el proceso de adaptación a

¹⁵ Tomado de http://www-07.ibm.com/software/kr/data/informix/product/pdf/d4gl_wp.pdf

¹⁶ <http://www.wikiteka.com/apuntes/semana-4/>

las nuevas tecnologías. Lo anterior permite que los costos se reduzcan y retiran la necesidad de un rediseño o reestructuración de un programa, ya que es tan fácil como presionar un botón y así migrar a diversas plataformas.

La nueva versión de Informix también combina otras ventajas como, un consumo menor de almacenamiento reduciendo el costo en hardware, también contempla una gran flexibilidad, una reducción en los requerimientos de memoria (en tipo de ejecución se cargan solo los módulos de p-código que se están usando actualmente) y una portabilidad muy grande ya que puede montarse sobre muchas plataformas. La evolución de los lenguajes tiende cada vez más a alejarnos de la maquina o hardware, creando una mayor abstracción de los problemas a resolver, esto es beneficioso pues genera un ahorro significativo de recursos como el tiempo que es tan valioso actualmente.

Los Lenguajes de Cuarta Generación tienden a ser muy compatibles entre sus mismas evoluciones lo que nos permite crear aplicaciones con la confianza de que el trabajo realizado no será desechado más adelante, tal es el caso de Informix.

CAPÍTULO 2.

DEFINICIÓN DEL PROBLEMA Y ANÁLISIS INICIAL

2.1. Introducción.

Con la finalidad de obtener una información oportuna, concreta y real del proceso de registro de información de los empleados que se incorporan y son parte de la nómina de la Empresa Importadora Comercial El Hierro, ha sido necesario recurrir a herramientas de captura de información como son la entrevista y la encuesta, las mismas que permitirán obtener de fuente directa información relevante sobre el registro y control de vacaciones del personal que labora en la Empresa, lo cual servirá para tener un diagnóstico inicial de la problemática del proceso de investigación planteado, y definir una solución software que resuelva el problema de control existente y lograr de esta manera los objetivos planteados con los mejores resultados posibles.

En el presente capítulo se describe el proceso de registro y control de vacaciones anuales que lleva a cabo el departamento de Talento Humano de la Empresa; de igual manera se detalla las deficiencias y falencias que existe en el proceso actual, y se plantean alternativas que pretenden a partir del diseño y análisis del proyecto entregar una solución software que mejoren significativamente el control y registro de vacaciones de los empleados de la Empresa.

2.2. Descripción del proceso actual.

Importadora Comercial El Hierro es una compañía limitada ubicada en la ciudad de Cuenca, fundada por un matrimonio emprendedor y visionario conformado por el Señor Rodrigo Matute Vázquez y la Señora Bertha Zamora Cabrera; constituida mediante escritura pública el 12 de julio de 1983, inscrita en el registro mercantil 152 junto con la resolución. 2590-10-83 de la Intendencia de Compañías de Cuenca, el 25 de agosto de 1983, en el año 1963 según consta en el Reglamento interno de la Empresa¹⁷. quienes ante la falta de servicios orientados al sector de la construcción y frente a la creciente demanda por parte de la comunidad cuencana establecen un negocio dedicado a la comercialización de materiales de construcción, ferretería y acabados de casa que permitan brindar un servicio especializado y de calidad a los consumidores de la provincia del Azuay. Actualmente la Empresa es un grupo corporativo familiar que se ha expandido a nivel nacional y cuenta con 14 sucursales distribuidas en las ciudades de Quito, Guayaquil, Cuenca, Machala, Ambato, Azogues y Loja con una

¹⁷ Reglamento interno de la Empresa Imp. Com. El Hierro Cía. Ltda.

infraestructura administrativa, tecnológica, logística y talento humano de primer orden que generan trabajo directo a trescientas personas y miles de empleo indirectos en todo el país.

La misión de la Empresa (Reglamento interno, 2013) tiene como objeto social el de importación, compra y venta de todo tipo de materiales y elementos que se requieren para la construcción, así como de maquinarias y herramientas que en esta actividad sean necesarias. Satisfacer las necesidades de la comunidad en la comercialización de materiales para la construcción, la industria y el hogar con calidad y garantía al menor precio, a través de la excelencia en el servicio al cliente.

La visión de la Empresa (Reglamento interno, 2013) es ser líderes en el mercado nacional en la comercialización de materiales para la construcción, la industria y el hogar, satisfaciendo oportunamente los requerimientos del cliente.

La Empresa registra en la actualidad una nómina promedio de trescientas personas que laboran a nivel nacional, que se encuentran registradas en la base de datos que dispone la Empresa; sin embargo el proceso de control y asignación de vacaciones anuales se lo viene haciendo por medio del Departamento de Talento Humano en forma manual.

Según (Gómez Nancy; 2014) jefe del departamento de talento humano y responsable de la nómina de la Empresa, manifiesta que el registro, asignación de días adicionales de vacaciones, diferimiento de vacaciones, control de asistencia y liquidación de vacaciones del personal de la Empresa, se lo realiza por medio de documentos elaborados en forma manual por el departamento a su cargo; el mismo que se basa a partir del cronograma anual de vacaciones elaborado en hoja electrónica en función de la fecha de ingreso del empleado (ver Anexo A); luego indica que para la autorización de vacaciones se entrega un formulario (ver Anexo B) elaborado por el departamento de talento humano al empleado, con el fin de que registre su petición de vacaciones, de igual forma el empleado puede solicitar el diferimiento de sus vacaciones; todas estas solicitudes deben ser autorizadas por el Jefe inmediato del empleado solicitante y validado posteriormente por el Jefe de Talento Humano.

Con esta información preliminar (cronograma anual y solicitud, ver anexo A y B), el asistente del departamento de talento humano, verifica que los datos registrados estén llenados correctamente y con las respectivas firmas de solicitud y autorización; se debe tomar en cuenta que las solicitudes de vacaciones pueden ser solicitadas por aquellos empleados que ya cumplieron su primer año de labores en la Empresa y reúnen los requisitos legales que indica el Ministerio de Trabajo y el Reglamento interno de la Empresa; artículos 28 y 29 (ver Anexo C).

El número de solicitudes de vacaciones varía en función de las fechas de ingreso de los empleados, cronograma anual y de las necesidades emergentes de los mismos. Una vez verificada la información, el Jefe de Talento Humano procede a notificar vía correo electrónico y/o documento escrito la autorización respectiva para goce de vacaciones del empleado, tanto al solicitante como a su Jefe inmediato y demás funcionarios que estén relacionados con la labor del beneficiario.

Adjunto se podrá observar en forma gráfica (Figura 12) el proceso de solicitud y registro de vacaciones.

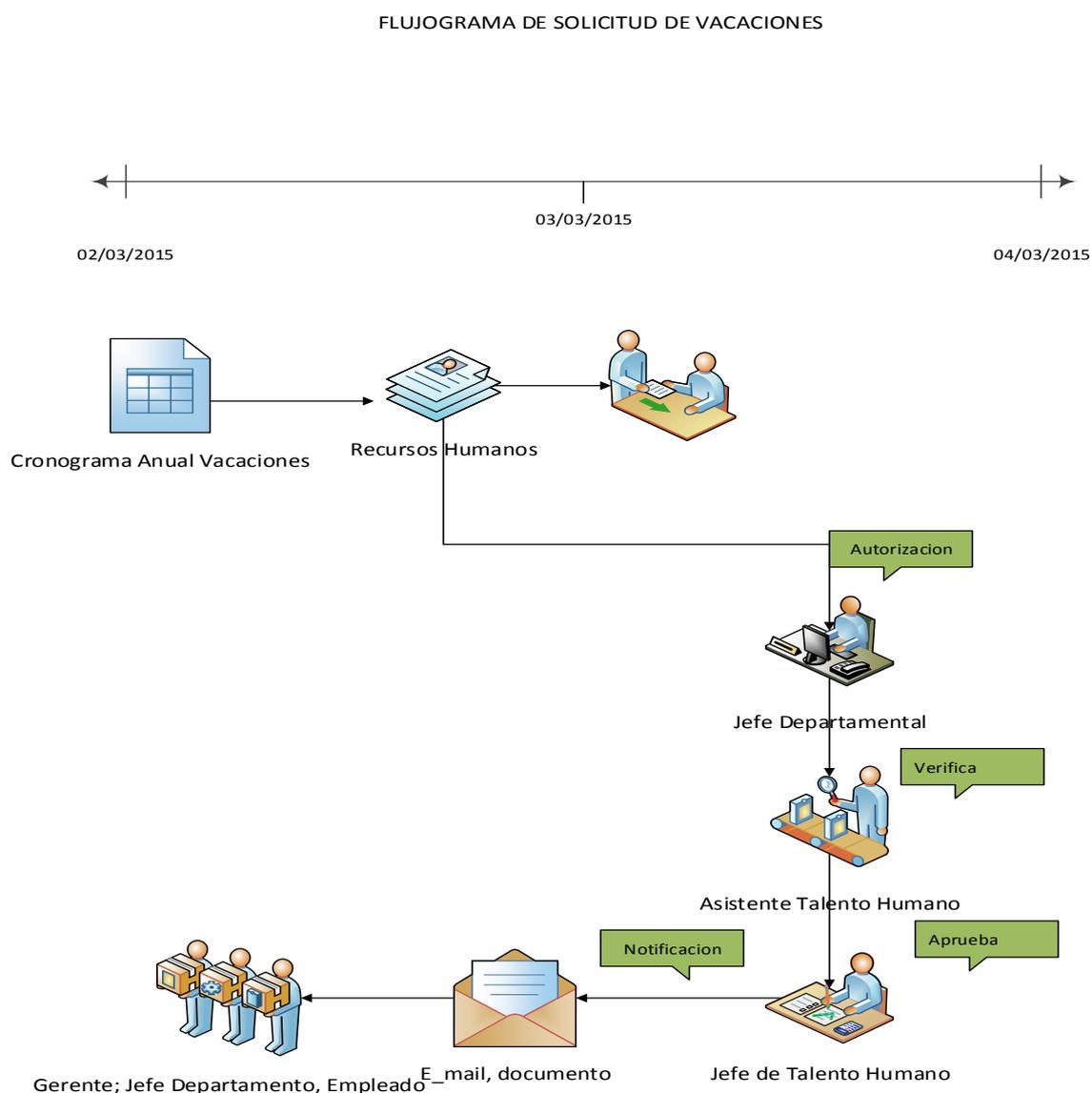


Figura 12 Proceso de solicitud de vacaciones.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

Como se pudo observar en la Figura 12, todo el proceso de registro y verificación de la información demora en autorizarse un promedio de 1 a 3 días, dependiendo de la disponibilidad de los funcionarios y autoridades de la Empresa; lo cual genera desperdicio de recursos y agilidad en la aprobación de dichos documentos, además de la confiabilidad e integridad de la información requerida.

2.3. Problemática.

Una vez que se ha descrito en forma somera el proceso vigente sobre la adjudicación de vacaciones anuales en la Empresa; en la siguiente Tabla. 5 se describe de manera simplificada los problemas que presenta llevar en forma manual el control de vacaciones de la Empresa Imp. Com. El Hierro, en especial la captura y posterior adjudicación y aprobación de vacaciones de los empleados de la Empresa.

Tabla 5. Descripción del problema

	Problema	Descripción del problema
1	Tiempo empleado en el proceso para solicitud de vacaciones	El asistente de talento humano de la Empresa, debe previamente consultar (ficha personal del empleado: archivos de la Empresa); verificar fecha de ingreso (ingresar al sistema), imprimir un formulario de solicitud de vacaciones (formato en Word). Todo este proceso requiere un tiempo considerable del asistente por cuanto debe trasladarse a la sección de archivos, verificar la información con la base de datos del sistema y finalmente entregar al jefe inmediato para su aprobación.
2	Trámite para aprobación de la solicitud de vacaciones	El jefe inmediato evalúa la solicitud y aprueba o niega en función de las necesidades de la Empresa; y remite al Jefe de Talento Humano para su notificación; este proceso puede tardar entre 1 a 2 días según disponibilidad de tareas de Talento Humano.
3	Registro y verificación de solicitudes entregadas	El departamento de Talento Humano procede con el registro de solicitudes aprobadas por los jefes departamentales, verifica que las mismas cumplan con las normativas de Ley; este proceso puede tardar en función del número de solicitudes entregadas.
4	Notificación de solicitudes aprobadas	El departamento de Talento Humano procede vía correo electrónico con la notificación respectiva a los empleados que han sido aprobadas las solicitudes, con copia a los jefes departamentales y Gerencia General.
5	Actualización y archivo de solicitudes de vacaciones	Una vez realizada la notificación, el personal de Talento Humano procede a actualizar la ficha personal del empleado, cronograma de vacaciones y sistema de rol de pagos de la Empresa.

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

2.4. Plan de solución.

Luego de realizar un diagnóstico inicial sobre el proceso actual de registro anual de vacaciones de la Empresa Imp.Com. El Hierro que se controla de manera manual y tradicional, el requerimiento de los directivos de integrar este control a los procesos automatizados que ya dispone la Empresa; y con el fin de mejorar y agilizar el registro, asignación, consultas y reportes de vacaciones de los empleados de la Empresa, se genera la necesidad de plantear una solución software que permita optimizar los recursos y actividades del Departamento de Talento Humano de la Empresa, dotándole de un módulo de programación que consiste en el desarrollo e implementación de una aplicación informática que permita el registro, control, verificación y validación de forma automática de los datos registrados de los empleados que laboran en Imp. Com. El Hierro de una manera segura y oportuna. La implementación y desarrollo de este módulo de programación permitirá integrar y completar el modulo principal de rol de pagos que dispone la Empresa; entregar información actualizada y confiable para el cálculo de vacaciones de cada uno de los empleados, así como optimizar tiempos y recursos del Departamento de Talento Humano.

La solución planteada para el control de vacaciones implica desarrollar una aplicación de escritorio que se sustenta en una programación en lenguaje de alto nivel de cuarta generación denominada 4gl que se ejecuta en un motor de la base de datos Informix, además contiene la generación de consultas y reportes basados en SQL (lenguaje de consulta estructurado); y se ejecuta en un servidor de Informix y reportes SQL que están configurados en un sistema operativo Red Hat que dispone la Empresa con las respectivas licencias de uso y desarrollo.

La solución planteada se basa en la utilización del patrón arquitectónico cliente-servidor; tomando en consideración la infraestructura técnica y de tratamiento de la información que dispone el Departamento de Sistemas de la Empresa, como se puede observar en la Figura 13.

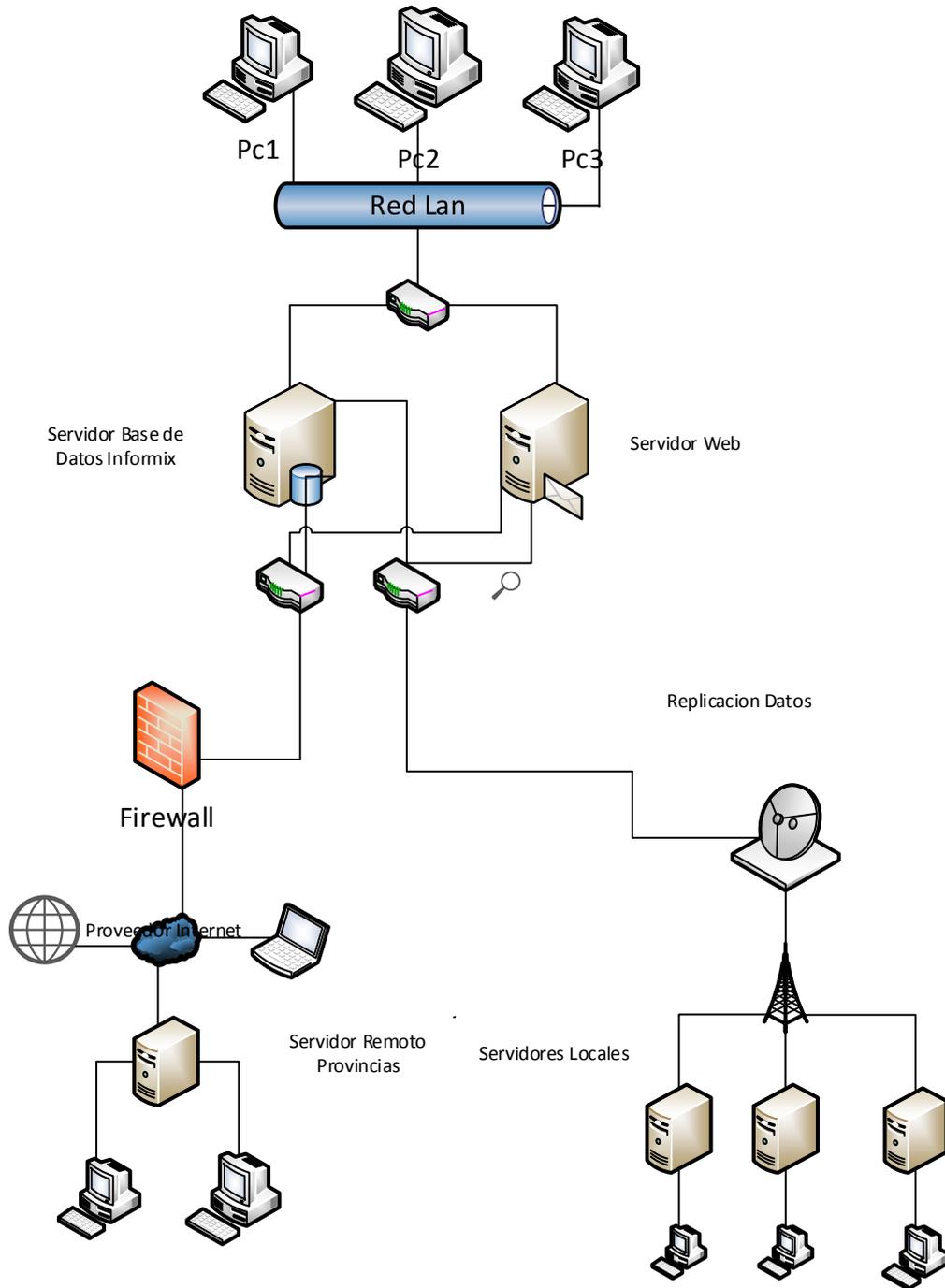


Figura 13. Implementación física de la solución software.
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

La implementación física de la solución software está conformada por una aplicación de escritorio, que se conecta desde la estación de trabajo que opera bajo el sistema operativo Windows mediante un software de emulación denominada NetTerm que es el encargado de

validar la identificación del usuario con el servidor de la base de datos Informix, este servidor esta igualmente configurado mediante reglas y políticas de acceso limitado determinadas por el Departamento de Sistemas de la Empresa.

El usuario de la aplicación mediante estos protocolos de acceso, puede ingresar y registrar la información desde cualquier estación de trabajo que dispone la Empresa a nivel nacional, ya sea directamente desde la Matriz Principal de la Empresa o mediante los servidores de otras ciudades o localizaciones que se conectan a través de la red interna soportada por equipos Cisco mediante VPN que brinda el proveedor de Internet contratado por la Empresa.

En la solución software planteada se aplica la metodología RUP, con la finalidad de cumplir con cada uno de los entregables que contempla cada fase de desarrollo y de esta manera garantizar el diseño, análisis, implementación y mantenimiento de la solución software. La aplicación del patrón arquitectónico cliente-servidor permite separar los datos de la aplicación de escritorio, la interfaz de usuario y la lógica de control de la aplicación en tres componentes principales y distintos como son cliente, servidor de aplicaciones, y servidor de base de datos; teniendo como resultado final una aplicación confiable, integrada, reutilizable y de fácil mantenimiento, lo cual permite detectar con facilidad cualquier error que pudiera presentarse; de esta manera se logra disminuir tiempos de desarrollo en futuras funcionalidades de la aplicación que se integren al módulo principal de la base de datos de la Empresa.

El patrón cliente-servidor ayuda a crear aplicaciones que separan los diferentes aspectos de la aplicación (lógica de entrada, la lógica de negocio, y la lógica de interfaz de usuario),

Esta separación le ayuda a gestionar la complejidad cuando se genera una aplicación, ya que le permite centrarse en un aspecto de la aplicación a la vez. Por ejemplo, usted puede centrarse en el cliente, sin depender de la lógica de negocio.

El diseño en tres capas trata sobre todo el no poner todo el código en las interfaces de usuario del sistema (IU). Para mejorar esto, la idea es tener 3 niveles de funcionalidad bien definidos:

- **Capa de presentación:** por medio de la interfaz de usuario IU (formularios Windows, páginas HTML,..) y sus controles visuales (textBox, comboBox, dataGrids) junto con sus eventos (los click y etc.)

- **Capa de negocio** (lógica del dominio), aquí ira todo el código que define las reglas de negocio (cálculos, validaciones). Surge de los procesos que hemos encontrado en el análisis.
- **Capa de acceso a datos**, el código que permite acceder a las fuentes de datos. Esencialmente trata sobre 4 operaciones básicas, (Ingreso, consulta, modificación y eliminación), que se realizan sobre la base de datos relacional Informix

Este diseño corresponde a las 3 capas lógicas (3 layers). Importante no confundir con las capas físicas (3 tiers) que corresponden al lugar donde se instalan los componentes de software¹⁸.

En 3 tiers la ejecución está distribuida:

- **Front-End**, donde se ejecutan las IU del cliente. Los Browsers.
- **MiddleWare**, recibe solicitudes de las IU a través de la red. Estos son mensajes (XML, SOAP,...) que se envían mediante protocolos de transporte HTTP, TCP, UDP,..). El Servidor Web.
- **BackEnd**, nuestra base de datos Informix o algún proceso externo a nuestro software.

En 3 tiers se instala en el BackEnd (la Base de datos) y parte en el MiddleWare (entidades de negocio y acceso a datos).

Los objetos del modelo son las partes de la aplicación que implementan la lógica de dominio de datos de la aplicación. En nuestro caso, un objeto de vacaciones podría recuperar información de la base de datos, actuar en él, y luego escribir la información actualizada en una Tabla vacaciones en la base de datos Informix SQL Server.

2.5. Descripción de la arquitectura.

Las aplicaciones Cliente-Servidor con Base de Datos se distinguen básicamente por las siguientes características:

¹⁸ <http://metodologiasdesistemas.blogspot.com/2007/05/diseo-en-3-capas-fisicas-lgicas-es.html>

- Toda la lógica de la aplicación (del negocio) físicamente está localizada dentro del servidor de aplicación, implementadas como transacciones, procedimientos almacenados, triggers, declaración de integridad referencial, restricciones en base de datos, etc.
- Al tener la lógica en el servidor, los servicios se construyen libres de contexto.
- El software cliente tiene el mínimo indispensable de lógica de aplicación, haciéndolo mucho más pequeño, liviano y fácil de distribuir. Es decir, en el cliente sólo hay lógica de navegación de pantallas, presentación de datos y validaciones físicas de datos.
- Dado que la lógica de la aplicación reside físicamente en el servidor, los clientes sólo ven datos y desconocen la configuración física de su almacenamiento permitiendo así liberar al servidor de datos de mantener el conocimiento de qué tareas está realizando el cliente (contexto) y se optimiza el uso de la red eliminando mensajes innecesarios.
- Independencia del software cliente de la estructura física y lógica de la base de datos, lo que en definitiva permite configurar la base de datos sin tener que reconstruir los clientes.
- Al tener toda la lógica de la aplicación en el servidor, ésta queda disponible para ser utilizada por cualquier otra componente de la aplicación aumentando la reutilización del código y bajando las probabilidades de inconsistencia.
- El tener el código centralizado y no duplicado, durante la fase de puesta en marcha, permite realizar las tareas de performance y tuning con mayor fluidez

En el caso de que la red LAN de PCs clientes se incremente (crecimiento de más sucursales en el país; proyección de más de 150 PCs) y tomando en consideración que la distribución geográfica es muy dispersa, tanto la distribución de la aplicación como su actualización, se convierte en un proceso muy complejo de administrar y con altos costos de inversión. Para solucionar esta limitación la arquitectura propuesta, utiliza la plataforma servidora de Informix, que funciona sin problemas con una configuración de clientes de esta magnitud. Para resolver la limitación del crecimiento en la cantidad de usuarios conectados a la aplicación, se dispone utilizar entre el cliente y el servidor de datos un servidor de aplicaciones para que realice las funciones de un monitor transaccional, llevando la arquitectura de 2 niveles a una de tres niveles físicos.

La conectividad con otras sucursales, se resuelve también con la arquitectura de 3 niveles, esto es habilitando los canales de comunicación en forma natural desde el nivel del monitor transaccional, permitiendo adicionalmente a la Empresa, definir los protocolos de comunicación con las entidades externas para intercambio electrónico de información; y poder incluir productos como correo y agenda electrónica (ver Figura 14).

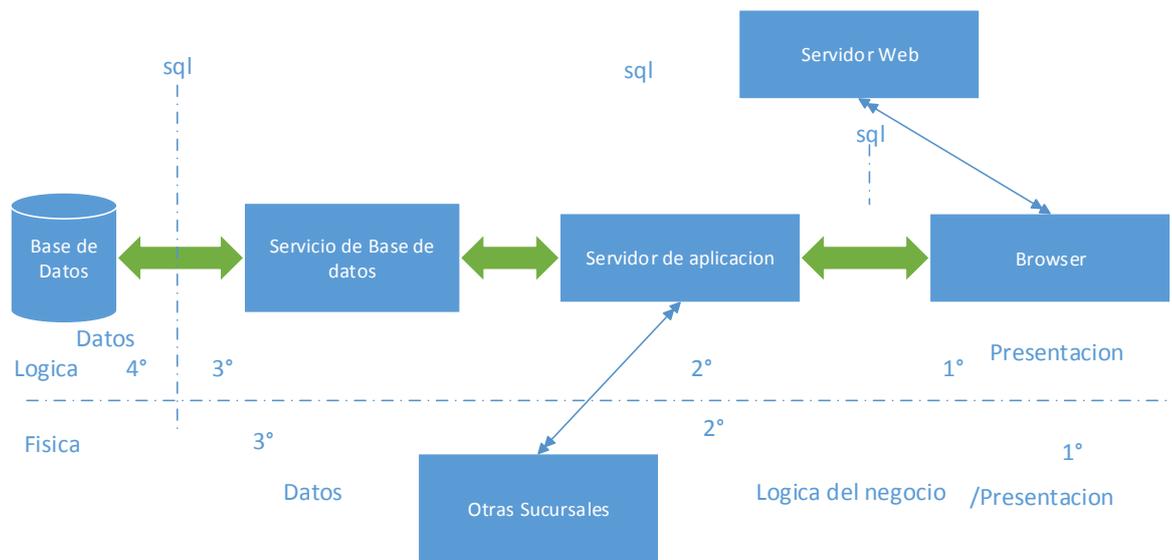


Figura 14. Implementación de cliente-servidor en la solución software.
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

2.6. Beneficios de la solución.

Con la ejecución de la solución software, se pretende beneficiar directamente el área de Sistemas y el Departamento de Talento Humano de la Empresa; adicionalmente se elevaría el grado de confiabilidad de los empleados en el manejo de su información referente al control de sus vacaciones; por otra parte el tiempo de respuesta para la verificación y aprobación de las solicitudes de vacaciones disminuirán considerablemente al tener automatizado el proceso; de igual forma significa un beneficio de carácter económico para la Empresa en virtud de ahorro de recursos de oficina, equipos utilizados e integración con el modulo principal de rol de pagos de la Empresa.

En forma general se detalla los beneficios que implica poner en marcha la solución software:

- Disminución de tiempo utilizado en la impresión y entrega de solicitudes para el registro y control de vacaciones de los empleados de la Empresa.
- Eliminación del registro individual de control de vacaciones sujeto a deterioro físico y pérdida de información.

- Integración de la solución software al Modulo general de rol de pagos que tiene implementado la Empresa.
- Disminución de costos de materiales de oficina y optimización de recursos.
- Información oportuna, confiable y rápida del registro automatizado de control de vacaciones de cada uno de los empleados de la Empresa.

2.7. Resultados esperados.

Una vez desarrollada e implementada la solución software se espera cumplir con los objetivos planteados en la propuesta planteada para mejorar el control de vacaciones de la Empresa Imp. Com. El Hierro de la ciudad de Cuenca.

- Desarrollar e implementar una aplicación de escritorio que permita llevar el control y registro de vacaciones de los empleados de la Empresa.
- Fácil manejo de la interfaz de tal manera que la misma sea una tarea agradable, confiable, sencilla y segura por parte del usuario.
- Que la aplicación esté sujeta a una parametrización, de tal manera que pueda ser fácilmente sostenida, escalable y actualizada en función de políticas y reglas del negocio.
- Implementación de seguridades adecuadas, basadas en normativas dadas por el Departamento de Sistemas de la Empresa.
- Disponibilidad de la información, que permita agilizar y asignar las vacaciones de los empleados de manera segura, rápida y oportuna.

CAPÍTULO 3.

DESARROLLO DE LA SOLUCIÓN

3.1. Introducción.

Al examinar la arquitectura en tres niveles, podemos darnos cuenta que el esquema conceptual viene a ser el “corazón” de la base de datos, que brinda el soporte a todas las vistas externas y se sustenta en el esquema interno. Sin embargo, el esquema interno no es otra cosa que la implementación física del esquema conceptual. Por lo tanto el esquema conceptual debe ser una imagen completa y precisa de los requisitos de datos de la Empresa, de lo contrario parte de la información de la Empresa no estará representada de forma correcta y se tendrá dificultades para implementar la solución software. Entendemos como diseño conceptual de la base de datos como el proceso de construir un modelo del uso de la información dentro de la Empresa, modelo que debe ser independiente de los detalles de la implementación.

La arquitectura de base de datos ANSI-SPARC utiliza tres niveles de abstracción: externo, conceptual e interno. El nivel externo está constituido por las vistas que los usuarios tienen de la base de datos. El nivel conceptual determina el contenido de información de la base de datos completa, es decir todas las entidades, sus atributos y sus relaciones, así como las restricciones aplicadas a los datos y la información de seguridad e integridad. El nivel interno se refiere a como se representan los datos, secuencia de los registros, índices y punteros, etc. En el presente proyecto se utilizara el patrón arquitectónico Cliente-Servidor que hace referencia a la manera en que los componentes software interactúan; es un modelo en tres niveles, compuesto por un modelo de interfaz de usuario (el cliente), un nivel de lógica de negocios y de procesamiento de datos (servidor de aplicaciones) y el servidor de base de datos distribuidos en máquinas distintas. Adicional a ello un software que controla la transferencia de datos entre clientes y servidores que permite en rutar las transacciones, la misma que garantiza fiabilidad e integridad.

3.2. Metodología de desarrollo de software.

El uso de la metodología de desarrollo de software RUP, apoyado por el lenguaje de modelado unificado UML, serán de suma importancia en el análisis, diseño e implementación de la solución software. Cada vez es mayor el número de empresas que están estandarizando la manera de modelar los datos, UML es un estándar del sector, es más claro y fácil de usar, además proporciona una mayor sinergia; todo esto ayudara a diseñar de mejor manera los

diferentes diagramas de cada fase de la metodología, que repercutirá en la construcción de la solución software.

3.3. Fases de la metodología de desarrollo de software.

Entre las características esenciales de RUP al ser una forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo) además de iterativo e incremental, está centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos como por ejemplo, el modelo de casos de uso, el código fuente, etc. y roles tales como el papel que desempeña una persona en un determinado momento; su estructura dinámica de RUP permite que detalle a continuación la tabla que corresponde a las fases, tareas y artefactos empleados en el desarrollo de la solución software.

Tabla 6. Fases y artefactos de la metodología de desarrollo de software.

FASES	TAREAS	ARTEFACTOS
Fase de Inicio	Determinar el alcance del proyecto y establecer los requisitos funcionales y no funcionales de la solución software.	<ul style="list-style-type: none"> ➤ Documento de Visión ➤ Documento de Especificación de requerimientos.
Fase de Elaboración	Detallar y seleccionar los casos de uso que permitirán seleccionar la arquitectura para la solución software, además depurar las funcionalidades y requisitos pendientes y generar mejoras para el desarrollo de la solución software.	<ul style="list-style-type: none"> ➤ Documento de Especificación de Casos de Uso. ➤ Arquitectura de Software conformada por: <ul style="list-style-type: none"> Vista Lógica: <ul style="list-style-type: none"> ➤ Diagrama de Clases ➤ Modelo E-R Vista de Implementación: <ul style="list-style-type: none"> ➤ Diagrama de Secuencia ➤ Diagrama de estados ➤ Diagrama de Colaboración Vista de Procesos: <ul style="list-style-type: none"> ➤ Diagrama de Actividad Vista de Despliegue: <ul style="list-style-type: none"> ➤ Diagrama de Despliegue

Fase de Construcción	Construir la solución y ejecutar las pruebas suficientes que garanticen la implementación de las funcionalidades, con el objetivo de satisfacer los requerimientos del cliente.	➤ Desarrollo de la solución software
Fase de Transición	Avalar la disponibilidad de la solución software para los usuarios, arreglar los errores y defectos encontrados en las pruebas, capacitar a los usuarios y ofrecer el soporte técnico necesario.	➤ Plan de Pruebas ➤ Manual de Usuario

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

3.4. Fase de inicio.

En esta fase se configura el documento de Visión que pretende recoger cada una de las necesidades que manifiestan los involucrados en el proyecto, evaluar las necesidades y presentar una visión global de la solución software; de igual manera se define el documento relacionado a la Especificación de requerimientos cuyo objetivo es detallar la información necesaria que permita comprender los requerimientos funcionales y no funcionales que se implementen en la solución software. Las técnicas que se han utilizado para este propósito son entrevistas a los involucrados, observación directa y estudio del proceso actual del registro de vacaciones que tiene la Empresa.

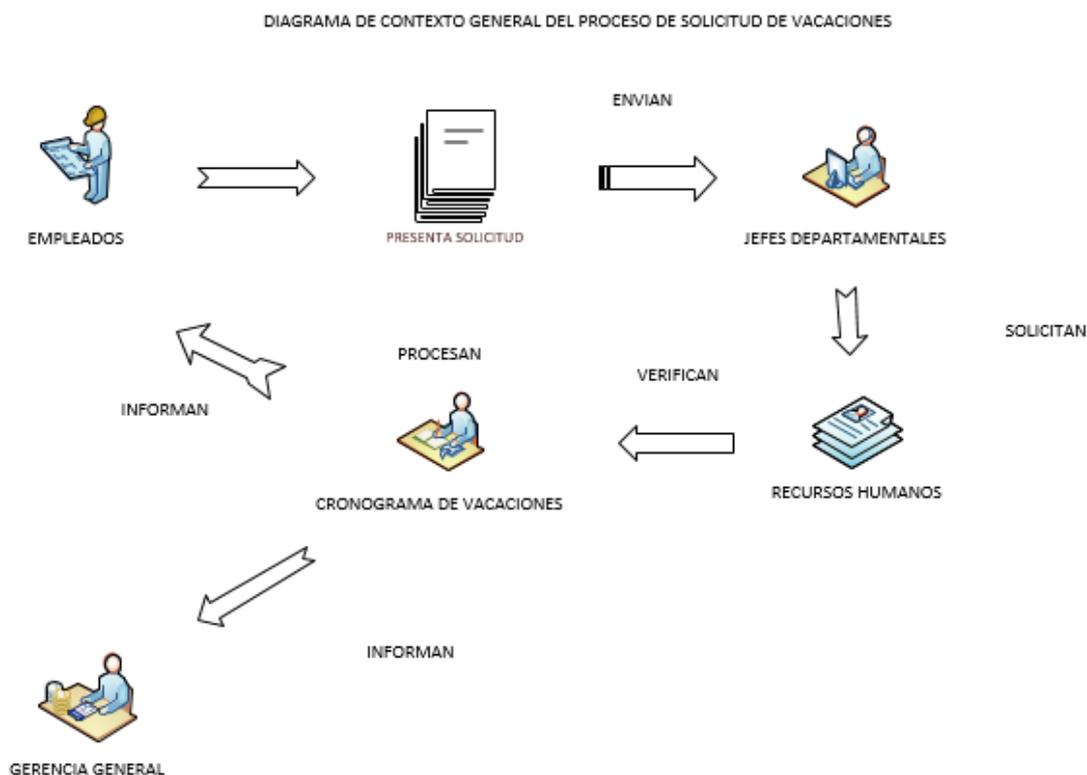
Las técnicas descritas permitirán analizar, conocer y comprender las necesidades de desarrollar una solución software y plasmar mediante los documentos de Visión y especificación de requerimientos, así como las técnicas de captura de información que se describen a continuación.

3.4.1. Estudio del proceso actual.

El responsable de Talento Humano de la Empresa proporciono la información necesaria y detallada sobre el proceso de control y registro de vacaciones del personal de Imp. Com. El Hierro, (Ver anexos A, B, C), lo cual permitió comprender y observar cómo se realiza actualmente el registro de vacaciones, el personal involucrado, materiales que se utilizan, tiempos de ejecución, etc. De la documentación recolectada y analizada se puede destacar lo siguiente:

- El cronograma anual de vacaciones que está elaborado en hoja electrónica Excel, se basa en función de la fecha de ingreso del trabajador, sin embargo su asignación puede variar en función de las necesidades de la Empresa.
- Las solicitudes de cargo a vacaciones son emitidas por el Departamento de Talento Humano y entregadas al Jefe de Almacén y/o departamento para su estudio y aprobación.
- Una vez que las solicitudes cumplen con los requerimientos legales y formales del reglamento interno, son revisadas y aprobadas por el Jefe de Talento Humano y se remiten vía correo electrónico interno al Gerente General para su visto bueno y aprobación definitiva.
- Por último se procede con la comunicación respectiva al beneficiario y al personal que pueda ser afectado por la ausencia temporal del trabajador para su conocimiento y programación respectiva de actividades.

A continuación se puede observar en el diagrama adjunto la secuencia del estudio del proceso actual.



Tomando en consideración el proceso descrito, se procedió a realizar las entrevistas, en especial al personal del departamento de Talento Humano, con la finalidad de auscultar y observar las actividades que ejecutan para llevar a cabo el registro de vacaciones.

3.4.2. Entrevistas.

Las personas seleccionadas para este proceso, provienen del departamento de Talento Humano de la Empresa, que son los actores principales en razón de que conocen la normativa legal vigente y el reglamento interno para su correcta aplicación. Las preguntas se basaron en aspectos claves del proceso como son las actividades que realizan, como lo hacen, que herramientas utilizan, que personas revisan, controlan y aprueban las solicitudes, documentos habilitantes, normativas vigentes, etc. Con la evaluación de estas entrevistas se ha podido realizar el levantamiento de información que permite continuar con la elaboración de los documentos visión y especificación de requerimientos.

3.4.3. Observación directa.

Este mecanismo permite observar de manera directa las diferentes actividades que genera el personal involucrado en el registro y control de vacaciones de la Empresa. Con este mecanismo se facilita determinar los flujos y secuencias del proceso y se logra observar las fortalezas y debilidades del proceso actual, lo cual es primordial para dar solución mediante la implementación de la solución software propuesta.

3.4.4. Documento Visión.

El documento Visión permite establecer la relación existente entre los actores (personal de la Empresa) y el proyecto como tal; de igual forma configura toda la información relevante y necesaria para el proyecto como es: perfil de los actores, definición del problema actual, análisis de factibilidad, presupuesto y restricciones para la implementación de la solución software.

3.4.4.1. Definición del problema actual.

A través de la siguiente Tabla 7 se describe varias preguntas que determinan el problema actual.

Tabla 7. Definición del problema actual.

Definición	Descripción
El problema de...	Asignación de tiempo para el asistente de Talento Humano en configurar, generar e imprimir desde el computador todas las solicitudes de vacaciones presentadas por los trabajadores, lo cual conlleva demasiado tiempo verificar en el archivo manual el registro de vacaciones para su posterior aprobación y comunicación. Todo este proceso manual de convalidación toma entre 1 a 3 días dependiendo del número de solicitudes presentadas.
Afecta a...	<ul style="list-style-type: none"> ➤ Personal de la Empresa ➤ Departamento de Sistemas ➤ Departamento de Talento Humano ➤ Emisión de rol de pagos
El impacto asociado es ...	Demora en el registro y control de vacaciones en las planillas de control del IESS, que puede perjudicar al trabajador en su contabilización de aportes.
Una solución apropiada sería...	Realizar una aplicación automatizada que permita llevar el registro de vacaciones de cada trabajador de manera oportuna, rápida y confiable e integrada al sistema de rol de pagos de la Empresa.
Para...	Beneficio de los trabajadores, que sabrán con claridad cuantos días disponen de vacaciones anuales y poder programar sus actividades con tiempo y seguridad.
Quienes...	Trabajadores de la Empresa en general, Jefes de Almacén y Jefes Departamentales.
Nombre de la solución software	Control de vacaciones que permita automatizar anualmente el registro de días disponibles de cada trabajador.
Que...	Permite el registro de vacaciones anuales de manera ágil y eficiente mediante la asignación automática en función de su fecha de ingreso a la Empresa.
No...	Como archivar en carpetas manuales que pueden extraviarse, averiarse o perderse y causar perjuicio a los trabajadores y reclamos a la Empresa.
Nuestra solución	Permitirá registrar en el sistema integrado de la Empresa toda la información relativa al control de vacaciones, de tal manera que pueda ser consultada y procesada en la base de datos disponible de la Empresa.

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

3.4.4.2. Funciones del personal involucrado en el proyecto.

Dentro del documento de visión se destaca el papel que desempeña funcionarios de la Empresa en la definición y alcance del proyecto, sus responsabilidades, toma de decisión e

información que aportan para el proyecto. En la siguiente Tabla (ver Tabla 8) se describen en forma resumida sus actividades y responsabilidades.

Tabla 8. Descripción y responsabilidades de los involucrados en el proyecto.

FUNCIONARIO	DESCRIPCION	RESPONSABILIDAD
Gerente General	Representante legal de Imp.Com. El Hierro Cía. Ltda.	Asignar los recursos económicos y tecnológicos para la ejecución del proyecto y vigilar su cumplimiento en los plazos establecidos.
Jefe de Sistemas	Configurar la base de datos, integración del módulo de vacaciones al sistema principal del SGBD de la Empresa	Aprobar los requisitos y funcionalidades del proyecto, así como coordinar con el departamento de Talento Humano la definición de parámetros y necesidades del módulo a integrar.
Jefe de Talento Humano	Controlar que el proceso de registro de vacaciones cumpla con las normativas legales vigentes y reglamento interno.	Aprobar los requisitos y funcionalidades del proyecto y facilitar toda la información necesaria para el éxito del proyecto.
Asistente de Talento Humano 1	Coordinar el registro de vacaciones en base al cronograma anual de adjudicación de vacaciones.	Registrar y validar el ingreso, modificación y eliminación de los registros de vacaciones de los trabajadores en el sistema de base de datos de la Empresa.
Asistente de Talento Humano 2	Imprimir y comunicar a los trabajadores y directivos de la Empresa la aprobación de solicitudes de vacación presentadas.	Archivar las solicitudes de vacaciones registradas en cada carpeta de los trabajadores para consulta y revisión de los entes competentes.

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Mayor detalle del proceso de la fase de inicio referirse al anexo D.

3.4.5. Especificación de requerimientos.

El propósito de la documentación de visión sobre la especificación de requerimientos es obtener toda la información que sea necesaria de tal manera que permita analizar y comprender los requerimientos que la Empresa desea y pueda ser evaluada de forma precisa

los requisitos funcionales y no funcionales del proyecto. Los requerimientos funcionales definen el comportamiento interno del sistema, entre los que se destaca los detalles técnicos, cálculos, manipulación de la información y otras funcionalidades en las que se demuestra como los casos de uso son aplicados en la práctica; en cambio los requerimientos no funcionales explican criterios que pueden ser utilizados para determinar la calidad y robustez de la operación del sistema.

En la Tabla 9 se describe los requisitos funcionales en base a un nivel de prioridad que se logra durante la implantación y desarrollo del proyecto.

Tabla 9. Requisitos funcionales de la solución software.

CODIGO RF.	REQUISITO	PRIORIDAD
RF01	Verificar cronograma anual de vacaciones	BAJA
RF02	Autenticación de usuarios	MEDIA
RF03	Solicitud de vacaciones	MEDIA
RF04	Registro de vacaciones	ALTA
RF05	Anulación de Vacaciones	ALTA

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

Para mayor información sobre los niveles de prioridad y requisitos funcionales y no funcionales referirse al anexo E.

3.4.5.1. Visión de la solución software.

La puesta en marcha e implementación de la solución software tiene como finalidad responder las necesidades de los usuarios involucrados y resolver la problemática; para lo cual la implementación se sustenta en el cumplimiento de todos los entregables inmersos en cada una de las fases de la metodología de desarrollo de software, realizando las iteraciones necesarias con el objetivo de desarrollar un producto software confiable que satisfaga los requerimientos de los usuarios.

3.4.5.2. Características de la solución software.

Las características de la solución software representan los módulos que conforman la solución software y que se detallan en la Tabla 10.

Tabla 10. Descripción de los módulos de la solución software.

CÓDIGO M.	MÓDULO	DESCRIPCIÓN DEL MÓDULO
M01	Cronograma anual de vacaciones.	Mediante la lectura del cronograma anual de vacaciones registrado en la hoja de cálculo Excel, visualizará la información del personal de trabajadores con derecho a vacaciones, previo al ingreso al sistema.
M02	Autenticación de usuarios	En este módulo se valida a los usuarios que tienen previamente registrados sus permisos de acceso al sistema y deben ser empleados de la Empresa. En función de los permisos asignados por el Dpto. Sistemas podrán acceder a la aplicación de Control de vacaciones.
M03	Registro de Vacaciones	Módulo principal integrado al sistema, el cual permite registrar, consultar, modificar, imprimir y eliminar el registro de vacaciones anuales del empleado.

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

3.5. Fase de elaboración.

En esta fase se configura el documento relacionado a Especificación de casos de uso que tiene como finalidad detallar en forma concreta cada una de las funcionalidades que el usuario de la solución software podrá realizar; también se define el documento sobre Arquitectura del Software que tiene como propósito asegurar que la solución software cumpla con todas las funcionalidades que aspira el usuario, tomando en consideración los atributos de calidad que son indispensables y las resoluciones que se plantearon al momento de especificar la construcción de la arquitectura.

3.5.1. Especificaciones de casos de uso.

Una vez establecido los actores que participan en la solución software (Ver Anexo D) y de haber conseguido los requerimientos funcionales (Ver Anexo E), se presenta en forma detallada la interacción de los usuarios con las funcionalidades de la solución software, que se muestra en la Tabla 11.

Tabla 11. Descripción de los casos de uso.

CÓDIGO CU.	ACTIVIDAD	ACTORES	DESCRIPCIÓN DE CASOS DE USO
CU01	Consultar Cronograma Anual de Vacaciones	Empleados de Imp.Com. El Hierro	Los usuarios deberán abrir la hoja de Excel con el cronograma de vacaciones.
CU02	Control de Acceso	Empleados de Imp.Com. El Hierro	Los usuarios realizan su registro de ingreso al sistema digitando su login y clave para poder operar el aplicativo.
CU03	Registro de Datos	Empleados de Imp.Com. El Hierro	Los usuarios admitidos por el sistema, pueden procesar la información relacionada al Control de Vacaciones.
CU04	Anulación de solicitudes	Empleados de Imp.Com. El Hierro	El administrador del sistema, puede anular una transacción relacionada al Control de Vacaciones

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Solamente los empleados de Imp. Com. ElHierro se les considera como usuarios o actores que interactúan con la aplicación, los asistentes de Talento Humano y Sistemas se les considera administradores del aplicativo.

Los casos de uso sirven para visualizar en forma transparente las funcionalidades que desenvuelve el usuario frente al aplicativo. Seguidamente en la Figura 15, se detalla el diagrama general de casos de uso que realiza el asistente de talento humano.

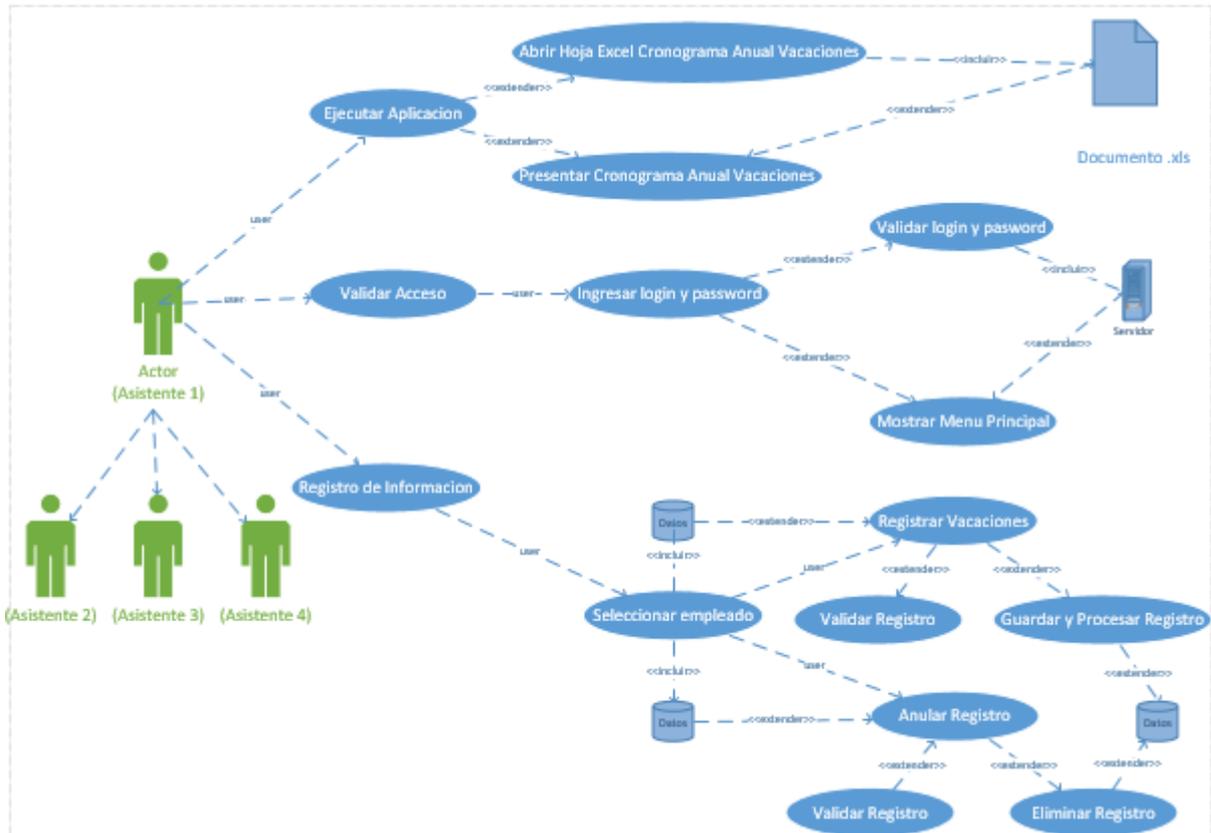


Figura 15 Diagrama general de caso de uso.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

3.5.2. Arquitectura de software propuesta.

La arquitectura de software implica considerar aspectos relevantes sobre la organización, planificación y comunicación que tiene el sistema a implantar, conceptos como la reusabilidad y facilidad en la evolución de la solución software, su acoplamiento a la metodología de desarrollo de software, con el propósito de obtener la documentación necesaria que permita plantear la arquitectura y que la misma cumpla con todas las funcionalidades que espera el usuario y los atributos de calidad asociados que deben regirse.

La arquitectura de software propuesta en la solución planteada, es el patrón arquitectónico Cliente-Servidor; En el mundo de TCP/IP las comunicaciones entre computadoras se rigen básicamente por lo que se llama modelo Cliente-Servidor, éste es un modelo que intenta proveer usabilidad, flexibilidad, interoperabilidad y escalabilidad en las comunicaciones.

Desde el punto de vista funcional, se puede definir la computación Cliente-Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma. En el modelo Cliente- Servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio). La forma más estándar de aplicación y uso de sistemas Cliente-Servidor es mediante la explotación de las PC's a través de interfaces gráficas de usuario; mientras que la administración de datos y su seguridad e integridad se deja a cargo de computadoras centrales tipo servidor. Usualmente la mayoría del trabajo pesado se hace en el proceso llamado servidor y el o los procesos cliente sólo se ocupan de la interacción con el usuario (aunque esto puede variar). Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

En la Figura 16 se presenta la arquitectura propuesta para el desarrollo de la solución software.

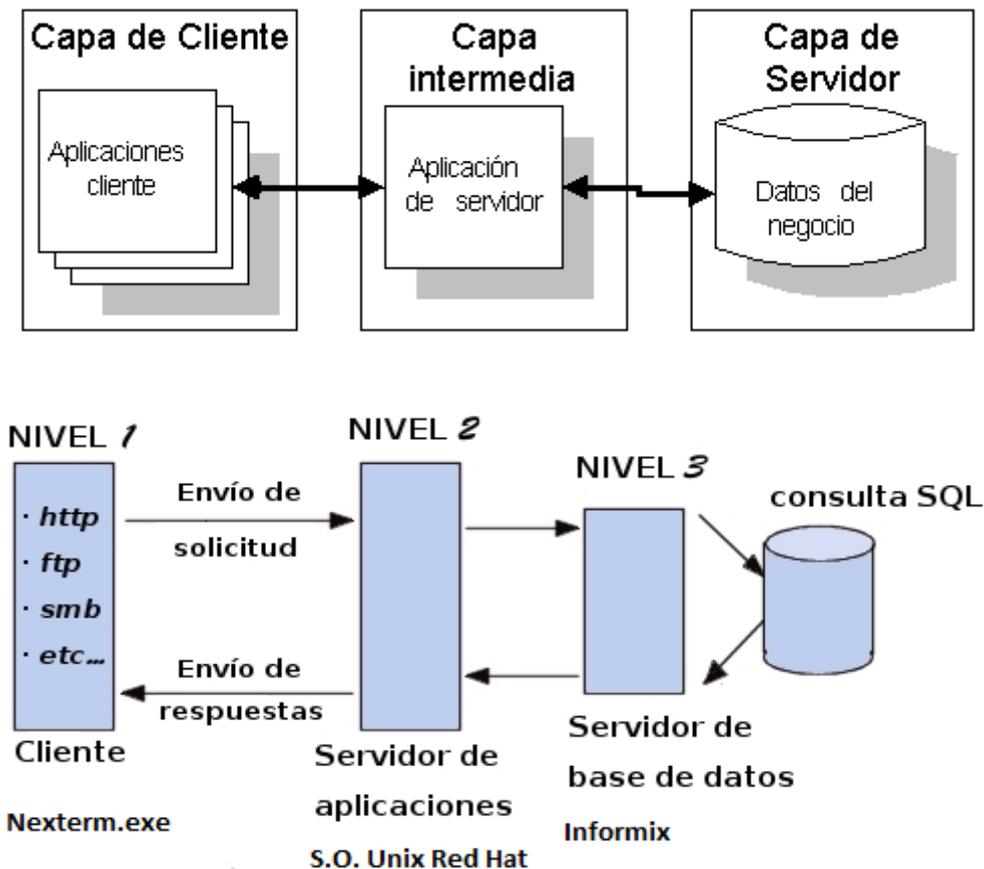


Figura16. Arquitectura propuesta.
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

3.5.3. Descripción del patrón arquitectónico Cliente–Servidor.

La arquitectura Cliente-Servidor es una extensión de programación modular en la que la base fundamental es separar el conjunto principal de software en módulos con el fin de hacer más fácil el desarrollo y mejorar su mantenimiento. Esta arquitectura permite distribuir físicamente los procesos y los datos en forma más eficiente para evitar el tráfico de la red que puede volverse no manejable en un momento determinado.

En esta arquitectura la capacidad de proceso está dividida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debido a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores de sistema operativo, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

El Cliente normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de la red de la Empresa. Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.
- Recibir resultados del servidor.
- Formatear resultados.

Servidor es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se le conoce con el término **back-end**. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Las funciones que lleva a cabo el proceso servidor se resumen en los siguientes puntos:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de bases de datos.
- Formatear datos para transmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

Front/end: Es la parte de la aplicación que interactúa con el usuario. Basados en una interfaz gráfica con el usuario (GUI). El Cliente corre la aplicación que ofrece la interfaz con el usuario.

Back/end: Es la parte no-interactiva de la aplicación. La mayor parte reside en las Bases de Datos (relacionales o no). No requieren una gran Base de Datos compartida, pueden ser elaboradas solamente en el Cliente. Exigen dos capas, una para la aplicación del usuario (Cliente) y otra para la base de datos (Servidor).

La Arquitectura de tres niveles es lógica y no física. Se preocupa con las funciones y no con la implantación. La Arquitectura es utilizada para desarrollar sistemas Centralizados o Distribuidos. La Arquitectura facilita la distribución de los componentes del sistema.

- Modelo presentación-negocio-datos
- Distribución flexible.
- Sistema abierto. No dependiente.

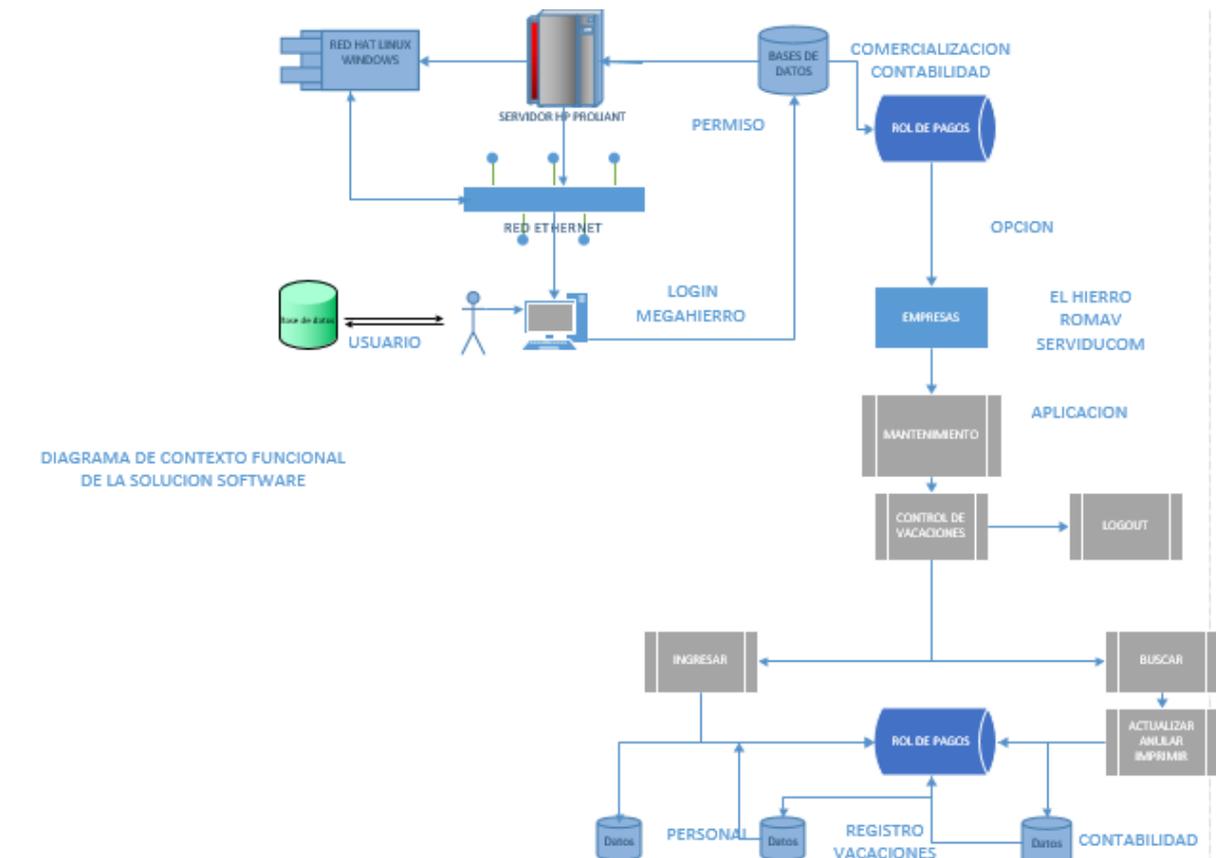
En la arquitectura Cliente-Servidor los tres niveles de aplicación se relacionan. Focaliza sobre la estructura y la adaptación. Y determina qué entra en cada nivel y cómo la aplicación se relaciona con otras aplicaciones.

- Distribución de Datos
- Distribución de Procesamiento
- Interfaz gráfica con el usuario

Lógica de la aplicación: se refiere a las funcionalidades que ofrece la solución software como son: consulta de usuarios, ingreso, modificación, eliminación, reportes de los registros de información de control de vacaciones, etc.; que son transacciones generadas por procedimientos de programación en 4GL y SQL ubicados en el motor de la base de datos Informix.

Interfaz de usuario (UI): software de emulación de terminal netterm.exe que sirve de entrada para la interacción de los usuarios con la solución software.

Diagrama de contexto: En la grafica siguiente se podrá visualizar en forma general como la solución software planteada se integra dentro del sistema automatizado que tiene actualmente implementado la Empresa, y como se ubican los procesos que desarrollan las funcionalidades de tal manera que demuestren la integración de la solución software al entorno global de la Empresa.



3.6. Fase de construcción.

En esta fase se detalla el desarrollo de la solución software, la cual contiene la codificación, en donde se procede con la implementación de los requerimientos funcionales de los usuarios involucrados y también los requerimientos no funcionales que la solución software mantiene consigo.

3.6.1. Cliente.

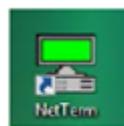
El Cliente representa la interfaz de usuario de la aplicación de escritorio, en ella se habilita un proceso de comunicación directa con el usuario, donde se produce una interacción dinámica que tiene como propósito facilitar el trabajo del usuario y evitar cualquier dificultad al acceder a la aplicación. Para la selección de la interfaz del usuario se ha considerado que el mismo se adapte a un entorno gráfico del sistema operativo Windows y el modo texto que caracteriza a la aplicación que se ejecuta en la plataforma del sistema operativo Red Hat Linux, tomando en cuenta criterios de usabilidad y accesibilidad que se manejan en el entorno de Informix.

3.6.2. Diseño de la interfaz del usuario.

Para el desarrollo de la interfaz del usuario se consideró el software de aplicación netterm.exe que consiste en un emulador de terminal tipo texto; dicha interfaz es muy amigable con el usuario y bastante liviana y rápida.

En la gráfica adjunta se presenta la pantalla de presentación al momento de ejecutar la aplicación desde el escritorio de Windows.

- Crear acceso directo en Inicio -> Todos los programas -> InterSoft International, Inc -> NetTerm.



Acceso Directo a NetTerm

- Presionar doble click sobre el icono de NetTerm lo cual permite observar la pantalla de presentación del emulador. (ver Figura 17).

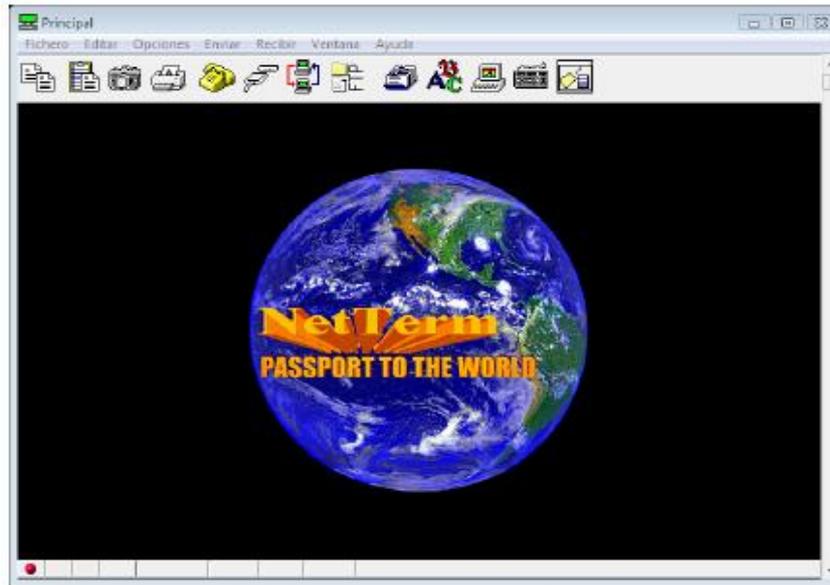


Figura 17. Pantalla de presentación.
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Para ingresar al sistema, el usuario debe seleccionar o escoger mediante el puntero del mouse si desea manualmente a través del tabulador el botón Conectar , seguidamente el sistema solicitara ingresar el login y password como se muestra en la Figura 18.

```
Red Hat Enterprise Linux Server release 5.1 (Tikanga)
Kernel 2.6.18-53.el5xen on an x86_64
login: egc
Password:
Login incorrect
login:
```

Figura 18 Ingreso al sistema.
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Si los datos ingresados no son los que se encuentran previamente registrados en la base de datos de Informix, el sistema visualizara un mensaje de error “login incorrect” tal como se aprecia en la figura anterior.

En caso de que el login y password estén correctos, aparecerá una pantalla de bienvenida con un listado de los empleados que cumplen años en el mes corriente; para omitir dicho detalle se puede presionar la combinación de teclas ctrl-C. Ver Figura 19.

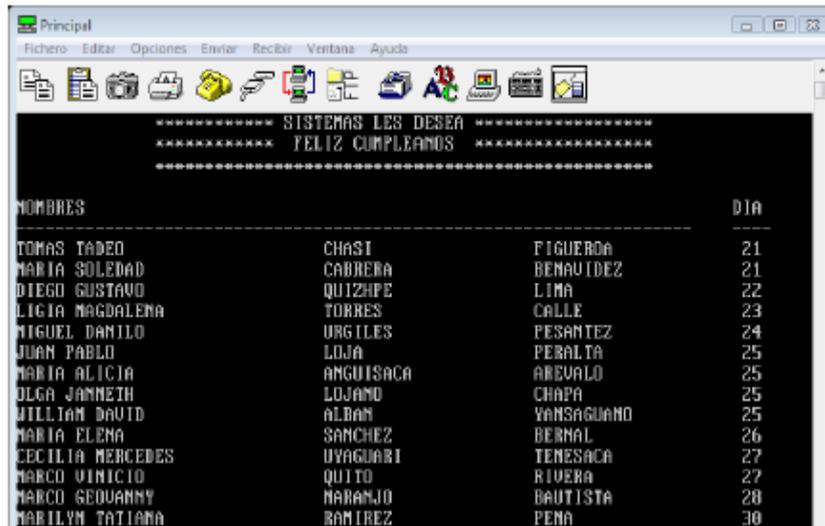


Figura 19. Bienvenida al sistema.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

Seguidamente se presenta las aplicaciones disponibles que contienen el modulo principal de la Empresa.



Figura 20. Aplicaciones disponibles.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

La aplicación desarrollada se encuentra localizada luego de seleccionar el módulo Rolpago, seguidamente seleccionamos la Empresa en la cual deseamos procesar la información de Control de Vacaciones; cabe anotar que el sistema es multi-empresarial, es decir permite trabajar en varias empresas que pertenecen al grupo corporativo de la familia Matute Zamora dueña de la Empresa Imp. Com. El Hierro. (Ver Figura 21).

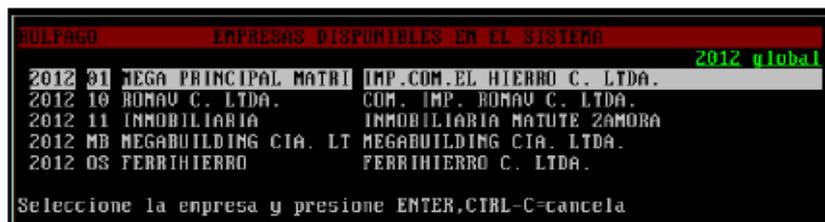


Figura 21. Empresas del grupo corporativo.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

Una vez seleccionada la Empresa; el usuario de la aplicación debe tener los permisos y accesos correspondientes, de no disponer estas autorizaciones se visualiza un mensaje de alerta.

```
...Usuario no autorizado CONSULTE CON EL ADMINISTRADOR
```

A continuación se presenta el menú principal de la solución software.

```
ROL DE PAGOS:  Mantenimiento Transaccion Consulta Reporte Salir
MANTENIMIENTO:  ... control Vacaciones personal MR Salir
REGISTRO Y CONTROL DE VACACIONES
```

Una vez ingresado a dicho programa, se manejan las siguientes opciones:

```
CONTROL VACACIONES:  Ingresar Buscar Modificar imprimir aNular ...
BUSCAR UN REGISTRO DE VACACIONES
```

Ingresar: Permite registrar una transacción de vacaciones, para un empleado que aún no tiene registros de vacaciones.

Buscar: Permite consultar historial de vacaciones, para empleados que tienen ya registrados transacciones de vacaciones.

Modificar: Agrega registros de vacaciones para un empleado, y adicionalmente permite visualizar los detalles de transacciones anteriores.

Imprimir: Imprime un listado de las transacciones de vacaciones de un empleado, mostrando ingresos, egresos y saldo.

Anular : Permite anular las transacciones de un empleado.

3.6.3. Servidor.

En el patrón arquitectónico Cliente Servidor, la parte que corresponde a las tareas que se realizan en el Servidor podemos mencionar que para su desarrollo se utiliza el lenguaje de programación de cuarta generación 4GL que permite generar el código fuente del aplicativo, en el cual se desarrolla la lógica del negocio y permite garantizar las funcionalidades del aplicativo. Para la codificación del aplicativo y que permita interactuar con la base de datos se consideró las siguientes características:

- Programación de la lógica del negocio aplicando lenguaje 4GL.
- Programación de instrucciones usando SQL
- Generación de reportes y formularios por medio FORMS y REPORTS
- Herramientas de administración de la base de datos DBACCESS
- Transferencia de archivos mediante FTP
- Configuración del servidor de la Base de Datos Informix versión 11.6

También contiene menús en modo texto para hacer las tareas de administración, compilación y ejecución.

Ejemplo de una serie de funciones de texto que aceptan de forma automatizada los mensajes.

```

FUNCTION message_box()
DEFINE caution_msg CHAR(50),
confirm_msg CHAR(50),
dummy SMALLINT
LET confirm_msg = "Presione una tecla para continuar."
CALL yesno_box(caution_msg, confirm_msg) RETURNING dummy
END FUNCTION -- message_box --
FUNCTION yesno_box(caution_msg, confirm_msg)
DEFINE caution_msg CHAR(50),
confirm_msg CHAR(50),
yes_no CHAR(1)
OPEN WINDOW w_yesno AT 10, 10
WITH 4 ROWS, 56 COLUMNS
ATTRIBUTE (BORDER, MESSAGE LINE FIRST+1,
PROMPT LINE FIRST+2)
MESSAGE caution_msg
PROMPT confirm_msg CLIPPED FOR CHAR yes_no
CLOSE WINDOW w_yesno
RETURN (DOWNSHIFT(yes_no) = "y")
END FUNCTION -- yesno_box --

```

La sintaxis que utiliza es muy similar a la de SQL de Oracle 8, pero presenta algunas pequeñas variaciones, además de funciones que no están presentes en SQL. Como ejemplo, Informix no utiliza left o right en los Joins, solo outer. Otra de las diferencias que se encuentra es la orden return, que en el caso de SQL solo acepta un valor y debe ser de tipo INT ,en SQL se utiliza generalmente para el retorno de errores, mientras que en Informix se puede utilizar para devolver datos de tipo: char, int etc.

Informix incorpora una serie de herramientas con el fin de facilitar y agilizar el trabajo al usuario, a continuación se muestran algunas de ellas:

ONCHECK: Comprueba los discos e índices y los repara.

ONMONITOR: Muestra un menú de administración.

ONSTAT [-A]: Saca estadísticas de la memoria (shared-memory) en el momento que el servidor está on-line.

ISQL: Muestra un menú para consultas, formularios y SQL.

DBACCESS: Menú con herramientas de administración.

3.7. Fase de transición.

En esta fase se tiene como propósito colocar la solución software a disposición de los usuarios finales; capacitar al usuario sobre la aplicación de escritorio y finalmente configurar el aplicativo de tal manera que esté de acuerdo con los requerimientos de hardware y software del entorno empresarial. Para lo cual se ha desarrollado el Manual del Usuario (ver Anexo I), que lo que pretende es brindar una ayuda adicional para el correcto manejo y funcionamiento de la solución software.

CAPÍTULO 4.

PRUEBAS DE VALIDACIÓN

4.1. Introducción.

El propósito de este capítulo está sustentado en un documento que contiene información relacionada con los objetivos, tanto generales como específicos, así como de la estrategia y los recursos que le serán destinados. Para ello es necesario definir que se va a probar, como se va a hacer, de tal forma que vayamos obteniendo resultados que nos permitan refinar el proyecto. La metodología del plan de pruebas está basado en RUP que ofrece su enfoque de pruebas para valorar la calidad del software por medio de los siguientes objetivos¹⁹:

- Encontrar y documentar los defectos en la calidad del software
- Aconsejando acerca de la calidad percibida en el software
- Proveyendo la validación de los supuestos hechos en las especificaciones de diseño y los requerimientos a través de demostraciones concretas
- Validando las funciones del producto de software según fueron diseñadas
- Validando que los requerimientos hayan sido implementados apropiadamente
- Validación del diseño.

4.2. Tipos de pruebas.

Para efectos de desarrollo de software, los tipos de prueba se basaran en la metodología RUP, teniendo en cuenta que esta metodología está orientada a la documentación y formalización de todas las actividades realizadas. Cabe resaltar que las pruebas se pueden realizar en cualquier momento del desarrollo de la solución software; a continuación se presenta un resumen de los tipos de prueba:

¹⁹ Fragmentos tomados del libro electrónico de RUP Versión 2003.06.12.01

Tabla 12. Tipos de prueba aplicadas a la solución software.

TIPO DE PRUEBA	DEFINICIÓN	FASE DE RUP
Unitarias	Consisten en la ejecución de actividades que le permitan verificar al desarrollador que los componentes unitarios están codificados bajo condiciones de robustez.	Elaboración
Integración	Consisten en la comprobación de que elementos del software que interactúan entre sí, funcionan de manera correcta.	
Sistema	Consiste en la ejecución de actividades de prueba en donde se debe verificar que la funcionalidad total de un sistema fue implementada de acuerdo a los documentos de especificación definidos en el proyecto; deben cubrir los aspectos funcionales y no funcionales del sistema, debe utilizar como bases de prueba entregables tales como: requerimientos iniciales, casos de uso, historias de usuario, diseños, manuales técnicos y de usuario final, etc. Por último, es importante que los tipos de pruebas ejecutados en este nivel se desplieguen en un ambiente de pruebas.	Construcción
Funcionales: Usabilidad Aceptación	Usabilidad permite encontrar problemas relacionados con el factor humano o denominado de usabilidad. La prueba de aceptación es considerada la prueba final basada en las especificaciones de los usuarios	Transición

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

4.3. Estrategia de pruebas.

Una estrategia de prueba del software debe incluir pruebas de bajo nivel que verifiquen que todos los segmentos de código fuente se han implementado correctamente, así como pruebas de alto nivel que validen las principales funciones del sistema frente a los requisitos del cliente.

En la Figura 21 se puede observar algunos roles y actividades presentes en el proceso de pruebas:

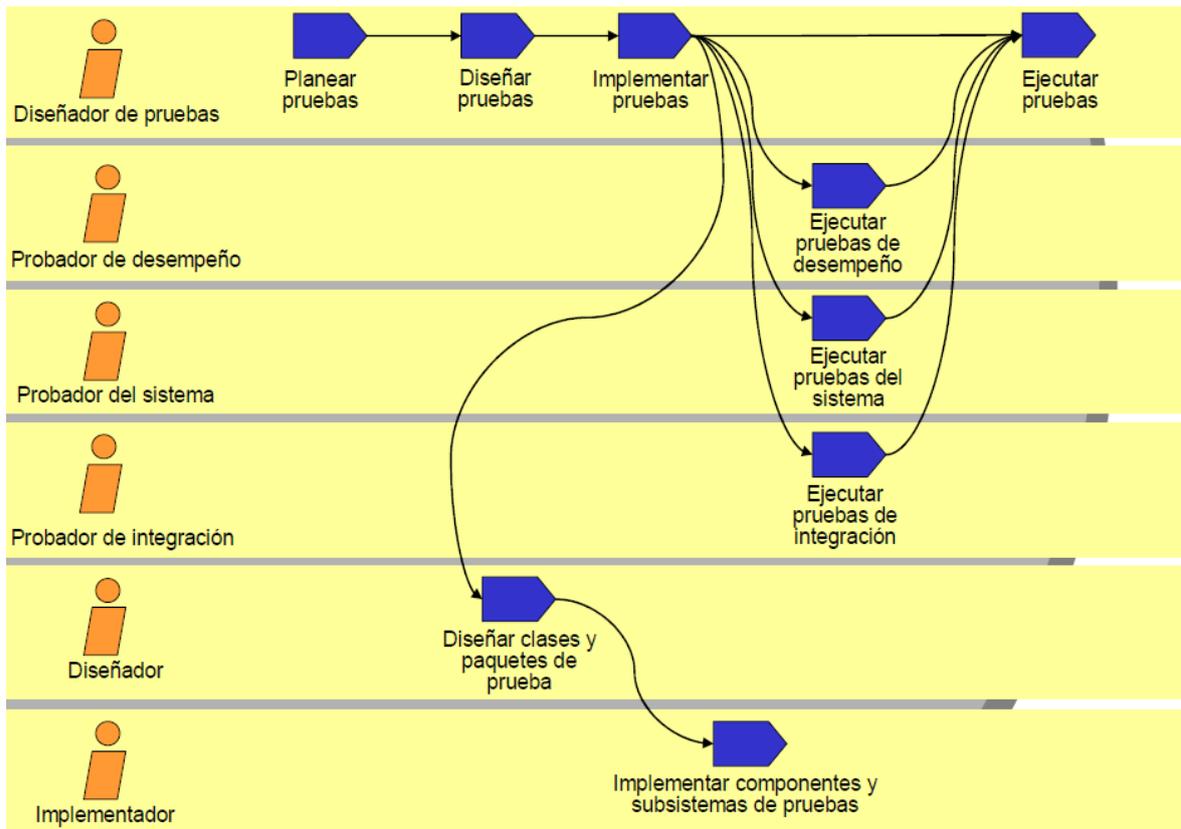


Figura 21. Roles y actividades del proceso de pruebas
 Fuente: libro electrónico de RUP Versión 2003.06.12.01.
 Elaborado por: Lloret Antonio.

4.4. Herramientas para pruebas.

En función del tipo de prueba, se usaran herramientas de software vinculadas con el motor de la base de datos Informix;

Informix incorpora una serie de herramientas con el fin de facilitar y agilizar el trabajo al usuario; así como permitir que el desarrollador pueda realizar pruebas de verificación e integridad de la base de datos y probar que la solución software cumpla con los requerimientos del usuario. En la siguiente Tabla 13 se enumera algunas de las herramientas a utilizar.

Tabla 13. Herramientas para la ejecución de pruebas.

TIPO DE HERRAMIENTA	DESCRIPCIÓN	TIPO DE USO
ONCHECK	Comprueba los discos e índices de la base de datos y los repara.	Reparación de índices en caso de fallos del servidor

ONMONITOR	Visualiza un menú de administración	Administra en forma directa funciones de recuperación y registro de datos
ISQL	Muestra un menú para consultas, formularios y SQL	Ejecución de scripts y procedimientos de consulta y verificación de datos.
DBACCESS	Menú con herramientas de administración de la base de datos.	Creación, Modificación, Eliminación de registros de la base de datos

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

4.5. Configuración del ambiente de pruebas.

El ambiente de pruebas para la solución software se desarrollara aprovechando las instalaciones de infraestructura de software y hardware que dispone la Empresa Imp.Com. El Hierro; y la computadora portátil del desarrollador en la cual se configurara una máquina virtual con el uso de la herramienta de código abierto VirtualBox que simulara el entorno de trabajo de la Empresa. En la siguiente Tabla 14 se describe los dispositivos que disponen la Empresa y el desarrollador:

Tabla 14. Configuración de equipos para el ambiente de pruebas.

COMPONENTE	CONFIGURACIÓN	SOFTWARE
SERVIDOR PRINCIPAL	HP ProLiant ML310e Gen8 v2 - Servidor – torre Procesador:Xeon E3- 1240V3 / 3.4 GHz Memoria: 16GB ram Disco duro: HDD 1X2 TB SATA 6Gb 7200rpm Procesador gráfico: Matrox 6200 PCI express 2xGigabit Ethernet	Red Hat Enterprice Linux Informix Dynamic Server
COMPUTADOR PORTATIL	Windows 8 64 bit ESD Memoria : 8GB ram	Windows 8 64 bit NetTerm

	Procesador: Intel Core I5 1.6ghz	Ms-Office 2013
	Disco Duro: 750 MB	VirtualBox

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Es importante destacar que en el servidor principal que representa la ubicación física del modelo en el cual se halla configurado el patrón arquitectónico cliente-servidor; que interactúa con la base de datos Informix; de igual manera se ha configurado en el equipo portátil un ambiente de prueba mediante una máquina virtual que simula el entorno de trabajo del departamento de sistemas de la Empresa.

4.6. Ejecución y resultados de las pruebas.

La fase de pruebas que se ejecutaran está descritas en la Tabla. 15 y que de acuerdo al libro electrónico RATIONAL UNIFIED PROCESS, "la disciplina de pruebas se centra principalmente en la evaluación de la calidad del producto y se lleva acabo de acuerdo a las siguientes prácticas:

- Buscar y documentar defectos en la calidad del software.
- Servir como guía para el logro de la calidad del software.
- Validar que el software funciona de acuerdo a las especificaciones que se hicieron en el diseño".²⁰

A continuación se describen las principales tipos pruebas que se pueden realizar a cualquier tipo de software. Cada prueba contendrá como mínimo la siguiente información:²¹

- Objetivo de la prueba
- Descripción de la prueba
- Técnica
- Criterio de Completitud
- Consideraciones Especiales

²⁰ <http://clases3gingsof.wikifoundry.com/page/Proceso+de+Pruebas+de+RUP>

²¹ <http://ing-sw.blogspot.com/2005/04/tipos-de-pruebas-de-software.html>

4.6.1. Pruebas unitarias.

Objetivo de la Prueba:	Se focaliza en ejecutar cada módulo (o unidad mínima a ser probada, ej. = una clase) lo que provee un mejor modo de manejar la integración de las unidades en componentes mayores. Busca asegurar que el código funciona de acuerdo con las especificaciones y que el módulo lógico es válido.
Descripción de la Prueba:	<ul style="list-style-type: none">➤ Particionar los módulos en pruebas en unidades lógicas fáciles de probar.➤ Por cada unidad hay que definir los casos de prueba (pruebas de caja blanca).➤ Para esto los casos de prueba deben diseñarse de forma tal que se recorran todos los caminos de ejecución posibles dentro del código bajo prueba; por lo tanto el diseñador debe construirlos con acceso al código fuente de la unidad a probar.➤ Los aspectos a considerar son los siguientes: Rutinas de excepción, Rutinas de error, Manejo de parámetros, Validaciones, Valores válidos, Valores límites, Rangos, Mensajes posibles.
Técnica:	<ul style="list-style-type: none">➤ Comparar el resultado esperado con el resultado obtenido.➤ Si existen errores, reportarlos.
Criterio de Completitud:	<ul style="list-style-type: none">➤ Todas las pruebas planeadas han sido ejecutadas.➤ Todos los defectos que se identificaron han sido tenidos en cuenta.
Consideraciones Especiales:	Para la elaboración de pruebas unitarias en Informix se puede utilizar 4GL; ISQL.

4.6.2. Pruebas de integración.

Objetivo de la Prueba:	Identificar errores introducidos por la combinación de programas probados unitariamente.
	Determina cómo la base de datos de prueba será cargada.

Verificar que las interfaces entre las entidades externas (usuarios) y las aplicaciones funcionan correctamente.

Verificar que las especificaciones de diseño sean alcanzadas.

Determina el enfoque para avanzar desde un nivel de integración de las componentes al siguiente.

Descripción de la Prueba:

- Describe cómo verificar que las interfaces entre las componentes de software funcionan correctamente.
- Determina cómo la base de datos de prueba será cargada.
- Determina el enfoque para avanzar desde un nivel de integración de las componentes al siguiente.
- Decide qué acciones tomar cuando se descubren problemas.

Por cada Caso de Prueba ejecutado:

- Comparar el resultado esperado con el resultado obtenido.

Técnica:

- Utilizar la técnica top-down. Se empieza con los módulos de nivel superior, y se verifica que los módulos de nivel superior llaman a los de nivel inferior de manera correcta, con los parámetros correctos.
- Utilizar la técnica down-top. Se empieza con los módulos de nivel inferior, y se verifica que los módulos de nivel inferior llaman a los de nivel superior de manera correcta, con los parámetros correctos.

Criterio de Completitud:

- Todas las pruebas planeadas han sido ejecutadas.
- Todos los defectos que se identificaron han sido tenidos en cuenta.

Consideraciones Especiales: Ninguna.

4.6.3. Pruebas de sistema.

Objetivo de la Prueba: Asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación.

Descripción de la Prueba: Las pruebas del sistema deben enfocarse en requisitos que puedan ser tomados directamente de casos de uso y reglas y funciones de negocios. El objetivo de estas pruebas es verificar el ingreso, procesamiento y recuperación apropiado de datos, y la implementación apropiada de las reglas de negocios. Este tipo de pruebas se basan en técnicas de caja negra, esto es, verificar el sistema (y sus procesos internos), la interacción con las aplicaciones que lo usan vía GUI y analizar las salidas o resultados.

En esta prueba se determina qué pruebas de Sistema (usabilidad, volumen, desempeño, etc.) asegurarán que la aplicación alcanzará sus objetivos de negocio.

La prueba de Sistema incluye:

- Prueba funcionalidad
- Prueba de Usabilidad
- Prueba de Performance
- Prueba de Documentación y Procedimientos
- Prueba de Seguridad y Controles
- Prueba de Volumen
- Prueba de Esfuerzo (Stress)
- Prueba de recuperación
- Prueba de múltiples sitios

Para capitalizar el trabajo hasta ahora completado, los casos de prueba de las pruebas previas realizadas pueden frecuentemente ser reorganizados y rehusados durante la prueba de sistema. No obstante, deben ser desarrollados casos de prueba adicionales para aquellos aspectos del sistema, tales como documentación, procedimientos y desempeño que no han sido probados durante la prueba unitaria y de integración.

La prueba de sistema es compleja porque intenta validar un número de características al mismo tiempo, a diferencia de otras pruebas que sólo se centran en uno o dos aspectos del sistema al mismo tiempo.

Técnica:	<p>Ejecute cada caso de uso, flujo básico o función utilizando datos válidos e inválidos, para verificar que:</p> <p>Los resultados esperados ocurren cuando se utiliza un dato válido.</p> <p>Los mensajes de error o de advertencia aparecen en el momento adecuado, cuando se utiliza un dato inválido.</p> <p>Cada regla de negocios es aplicada adecuadamente.</p>
Criterio de Completitud:	<ul style="list-style-type: none"> ➤ Todas las pruebas planeadas han sido ejecutadas. ➤ Todos los defectos que se identificaron han sido tenidos en cuenta.
Consideraciones Especiales:	<ul style="list-style-type: none"> ➤ Identifique o describa aquellos aspectos (internos o externos) que impactan la implementación y ejecución de las pruebas del Sistema.

4.6.4. Pruebas funcionales.

Objetivo de la Prueba:	Se asegura el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.
Descripción de la Prueba:	<p>Las pruebas Funcionales deben enfocarse en los requisitos funcionales, las pruebas pueden estar basadas directamente en los Casos de Uso (o funciones de negocio), y las reglas del negocio. Las metas de estas pruebas son:</p> <ul style="list-style-type: none"> ➤ Verificar la apropiada aceptación de datos, ➤ Verificar el procesamiento y recuperación y la implementación adecuada de las reglas del negocio.

Este tipo de pruebas están basadas en técnicas de caja negra, que es, verificar la aplicación (y sus procesos internos) mediante la interacción con la aplicación vía GUI y analizar la salida (resultados). Lo que se identifica a continuación es un diseño preliminar de las pruebas recomendadas para cada aplicación.

Técnica:

Se ejecuta cada caso de uso, flujo de caso de uso, o función, usando datos válidos e inválidos, para verificar lo siguiente:

- Que los resultados esperados ocurran cuando se usen datos válidos.
- Que sean desplegados los mensajes apropiados de error y precaución cuando se usan datos inválidos.
- Que se aplique apropiadamente cada regla de negocio.

- Todas las pruebas planeadas han sido ejecutadas.
- Todos los defectos que se identificaron han sido tenidos en cuenta.

Criterio de Completitud:

Consideraciones Especiales:

Identifique o describa aquellos aspectos (internos o externos) que impactan la implementación y ejecución de las pruebas de funcionalidad

Para mayor detalle de las pruebas descritas referirse al Anexo H.

4.7. Métricas del producto software.

Las métricas de calidad del software, como las métricas de productividad, se enfocan en el proceso, el proyecto y el producto. De tal manera que la Empresa al desarrollar y analizar puede corregir aquellas áreas del proceso de software que sean la causa de los defectos del software. El proceso de recopilación de datos, cálculo de métricas y análisis de métricas son pasos importantes que la Empresa debe implementar para iniciar un programa de métricas; el desarrollador de la solución software al generar una línea de referencia de métricas, una

base de datos que contenga mediciones de proceso y producto podrá obtener mejor comprensión del trabajo que realiza y de la solución software que desarrolla.

Lord Kelvin manifiesta que “La medición permite ganar comprensión acerca del proceso y del proyecto, al proporcionar un mecanismo de evaluación objetiva”.

4.7.1. Métricas en la fase de análisis.

Para esta fase nos apoyaremos en las métricas de software orientadas a la función cuyo propósito es obtener el punto de función que se basa en las características del dominio y de la complejidad de información del software, y que es más atractivo cuando se obtiene tempranamente en la evolución del proyecto. Para determinar el cálculo del punto de función se recurrió a los siguientes parámetros²²:

- **Número de entradas de usuario.**
Se cuenta cada entrada de usuario que proporciona diferentes datos a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada. Puntos de función se derivan de medidas directas del dominio de la información.
- **Número de salidas de usuario.**
Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, etc. Los elementos de datos particulares dentro de un informe no se cuentan de forma separada.
- **Número de peticiones de usuario.**
Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.
- **Número de archivos.**
Se cuenta cada archivo maestro lógico (esto es, un grupo lógico de datos que es parte de la base de datos o un archivo independiente).
- **Número de interfaces externas.**
Se cuentan todas las interfaces legibles por la máquina (por ejemplo: archivos de datos de cinta o disco) que se utilizan para transmitir información a otro sistema. Una vez que se han recopilado los datos anteriores, a la cuenta se asocia un valor de complejidad.

²²<http://ingsoft01-eam.blogspot.com/2011/10/metricas.html>

En función de los parámetros mencionados en la Tabla 15 se muestra dicha identificación en sujeción a cada requisito funcional del producto software:

Tabla 15. Identificación de los parámetros de la solución software.

CÓDIGO RF.	REQUISITO	ENTRADA USUARIO	SALIDA USUARIO	PETICIÓN USUARIO	ARCHIVOS	INTERFAZ EXTERNA
RF01	Verificar Cronograma anual de Vacaciones	0	1	0	0	1
RF02	Autenticación de usuarios	1	1	1	0	0
RF03	Solicitud de vacaciones	0	2	1	0	1
RF04	Registro de vacaciones	1	2	2	1	0
RF05	Anulación de vacaciones	1	1	1	0	0
TOTAL		3	7	5	1	1

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Por medio de la suma de los valores registrados en cada parametro de la Tabla . 15, se procede a colocar este valor en la Tabla 16 en donde se aplica la formula que consiste en multiplicar el valor de cuenta por el factor de ponderacion adecuado, en funcion del grado de complejidad del producto software, hasta obtener el valor de cuenta total.

Tabla 16. Obtención de valores del dominio de la información.

Parámetros	CUENTA	FACTOR DE PONDERACIÓN			SUBTOTAL
		Simple	Medio	Complejo	
Entradas de usuario	3	3	*4	6	12
Salidas de usuario	7	4	*5	7	35
Peticion de usuarios	5	3	*4	6	20
Archivos	1	7	*10	15	10
Interfaces externas	1	5	*7	10	7
CUENTA TOTAL					84

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Establecido el valor de **CUENTA TOTAL**, se procede con el cálculo del valor de complejidad **TOTAL GI**, que se sustenta en base a la asignación de los siguientes valores que se detallan en la Tabla 17.

- **Sin influencia (0):** El sistema no contempla este atributo.
- **Influencia Mínima (1):** La influencia de este atributo es muy insignificativa.
- **Influencia Moderada (2):** El sistema contempla este atributo, y su influencia a pesar de ser pequeña debe ser tomada en cuenta.
- **Influencia Apreciable (3):** La influencia de este atributo debe ser considerada, aunque no es relevante.
- **Influencia Significativa (4):** Este atributo es de gran importancia para el Sistema.

Tabla 17. Factores de ajuste de valor en la dificultad del sistema.

Nro.	FACTORES DE AJUSTE	VALOR
1	Comunicación de datos	4
2	Procesamiento distribuido	1
3	Objetivos de rendimiento	3
4	Configuración de uso intensivo	2
5	Tasas de transacción rápida	1
6	Entrada de datos en línea	3
7	Amigabilidad en el diseño	4
8	Actualización de datos on line	4
9	Procesamiento complejo	2
10	Reusabilidad	2
11	Facilidad de instalación	3
12	Facilidad operacional	4
13	Adaptabilidad	4
14	Versatilidad	3
	TOTAL GI	40

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

El punto de función se calcula utilizando la siguiente relación:

$$PF = \text{Conteo total} * (0.65 + 0.01 * \text{TOTAL}_{GI})$$

Donde Conteo total es la suma de todas las entradas PF obtenidas en la Tabla. 16, y Total GI se refiere a los factores de ajuste de valor (FAV).

$$PF = 84 * (0.65 + 0.01 * 40)$$

$$PF = 2217.60 \text{ LDC}$$

Este valor obtenido representa una estimación del tamaño global de la implementación de las funciones de la aplicación de control de vacaciones de la solución software. Los valores constantes en la ecuación y los factores ponderados que se aplican a los conteos de dominio de información se determinan de manera empírica.

Vale mencionar la importancia de la cita que realiza Michael Mah y Larry Putnam en el libro Ingeniería del software que dice “En lugar de solo meditar acerca de cuál “nueva métrica” aplicar [...] también debemos plantearnos la pregunta más básica: ¿Que haremos con las métricas?”.

4.7.2. Métricas en la fase de diseño e implementación.

“El producto, la tecnología y las personas tienen una fuerte influencia en el desarrollo y la calidad del software. El proceso se encuentra dentro de unas condiciones de entorno que incluyen: entornos de desarrollo, condiciones del negocio, y características del cliente. Estas condiciones, son de gran importancia puesto que permiten definir las reglas del proceso y poder contribuir a la calidad del software.”²³



Figura tomada de Roger Presuman. Ingeniería de Software

Figura 22. Desarrollo y calidad de software.
Fuente: (Roger S. Pressman, 2010).
Elaborado por: Lloret Antonio.

La eficacia de un proceso de software se mide a través de un juego de métricas según los resultados que provienen del proceso.

Dentro de éstos resultados se debe incluir:

²³<http://dateca.unad.edu.co/> Métricas del proceso y mejoras en el proceso del software

- Medida de errores detectados antes de la entrega del software
- Defectos detectados
- Productos de trabajo entregados
- Esfuerzo humano y tiempo consumido
- Ajuste con la planificación

4.7.3. Eficacia de la eliminación de defectos.

La eficacia de la eliminación de defectos (EED), es una métrica que permite medir la habilidad de filtrar las actividades de la garantía de calidad y de control, ya que es aplicable a todas las actividades del marco de trabajo del proceso.²⁴

Se define de la siguiente forma: $EED = E / (E + D)$

E	Número de errores encontrados antes de la entrega del software
D	Número de defectos encontrados después de la entrega

El valor ideal de EED es 1. No se han encontrado defectos en el software.

En la Tabla 18 se detalla el valor obtenido antes y después de depurar la solución software.

Tabla 18. Errores y defectos encontrados en la solución software.

ACCIONES REALIZADAS	ERRORES	DEFECTOS
Ejecutar interfaz del sistema	0	0
Abrir cronograma anual	0	0
Autenticación de usuarios	1	0
Registro de solicitudes	1	1
Búsqueda de solicitudes	1	0
Anulación de solicitudes	1	1
TOTAL	4	2

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

²⁴

$$EED = (4 / (4+2))$$

Eficacia de Eliminación de defectos = 0.67

En decir el valor de EED tiende a 1, lo cual implica que no se han concretado errores luego de la entrega del software. Por lo tanto queda garantizado la calidad del producto y el control de pruebas del aplicativo en búsqueda de posibles errores antes de la entrega final.

Para efectos de suficiencia del aplicativo se procedió en el mes de julio del 2015 generar solicitudes de ingreso y anulación de vacaciones del personal activo de la empresa, y no se registraron novedades en el funcionamiento de la solución software.

CAPÍTULO 5.

CONCLUSIONES, RECOMENDACIONES Y ACCIONES FUTURAS

Al culminar este trabajo de fin de titulación es necesario resaltar que la factibilidad del proyecto fue importante, pero considero que la necesidad empresarial lo es incluso mucho más; no siempre es aconsejable construir un sistema o solución software de alta tecnología que muchas veces nadie lo utiliza o aplica. En este proyecto de investigación e implementación se pudo obtener las siguientes conclusiones que aspiro cumplan con las expectativas y alcance del proyecto planteado.

Conclusiones

- Todo proyecto de software inicia por alguna necesidad de mejora; de corregir un defecto, o ampliar las funciones y características de una aplicación ya existente, o la necesidad de crear un producto, servicio o sistema nuevo; todos estos argumentos permitieron proponer una solución software que se adapte a las necesidades de la Empresa.
- La selección de la metodología de desarrollo de software RUP, su correcta y disciplinada aplicación en las diferentes fases de implementación permitieron consolidar y disponer de un producto software escalable, adaptable y funcional.
- La selección del patrón arquitectónico Cliente-Servidor permitió tener una visión clara e idónea para el diseño y construcción de la solución software basado en la infraestructura disponible de la Empresa, configurando de esta manera una aplicación confiable, estable y manejable en cualquiera de los componentes Cliente-Servidor.
- La implementación del software de emulación Netterm en el lado del Cliente permitió una comunicación liviana y rápida con el servidor de la base de datos que permite interactuar de forma amigable entre un ambiente Windows y Linux como sistemas operativos para la ejecución de la solución software.
- La utilización de los productos y herramientas de desarrollo de Informix conformados por el motor de la base de datos, lenguaje de programación 4GL, lenguaje de consulta estructurado SQL, generador automático de formularios y reportes, permitieron codificar y desarrollar la solución software con niveles altos de rendimiento, disponibilidad, replicación, seguridad, que permiten un procesamiento óptimo de transacciones en línea, al mismo tiempo reducen tiempos, grado de complejidad y conocimientos de administración de la base de datos.
- La última versión de IDS Informix Dynamic Server que se encuentra instalada en el servidor permitió simplificar y automatizar muchas de las tareas de mantenimiento de una base de datos empresarial, compatibilidad con múltiples plataformas y normas estándar abiertas, e incluye funciones de cliente-servidor.

- La realización del plan de pruebas permitió validar los requerimientos funcionales y no funcionales de la solución software; al mismo tiempo depurar el código fuente generado en 4GL, implementar scripts mediante SQL y hacer uso de DBACCESS directamente desde la base de datos Informix para probar la integridad de los datos y demostrar que la solución software es un producto de muy buena calidad.
- La puesta en marcha de la solución software permitió obtener muchos beneficios al departamento de Talento Humano de la Empresa, por cuanto se agilitaron los tiempos de respuesta en las solicitudes de vacaciones de los empleados, precisión y control en los cálculos de asignación de días de vacación, generando confianza y seguridad en la integración con el módulo de rol de pagos que mantiene la Empresa.
- Mediante la integración del módulo control de vacaciones en el menú principal de rol de pagos que tiene implementado la Empresa; se pudo constatar por parte del personal de Talento Humano y Sistemas, la adaptabilidad, facilidad y performance de la solución software.

Recomendaciones

Luego de observar y analizar los resultados obtenidos en la aplicación de la solución software vale la pena dejar planteado las siguientes recomendaciones:

- El cronograma anual de vacaciones que la Empresa mantiene actualmente en formato de hoja de cálculo debería ser automatizado en base a la información disponible en la solución software implementada.
- El formulario de solicitud de vacaciones que actualmente entrega el departamento de Talento Humano debería ser estandarizado y generado automáticamente desde la base de datos con la información personal de cada uno de los empleados que cumplen con los requerimientos de orden legal.
- El cambio de modo texto a modo gráfico de toda la aplicación existente en la Empresa, ayudaría muchísimo para mejorar la presentación de la información e identificación de los empleados de la Empresa, así como los productos que comercializan.
- Incorporar la herramienta OpenAdmind para identificación de las administraciones remotas.
- Seguir manteniendo el motor de la base de datos Informix, porque con las nuevas versiones del producto se pretende ofrecer mayor rendimiento, disponibilidad y capacidad de ampliación para manejar volúmenes masivos de datos complejos a muy alta velocidad.

Acciones futuras

Considerando que gracias a la eficiente gestión de parte de Directivos y trabajadores de la Empresa Imp. Com. El Hierro, en el que se augura un crecimiento constante se podría seguir con la implementación de nuevos módulos de control relacionados con el Departamento de Talento Humano que complementen un sistema totalmente integrado con las otras áreas de la Empresa con una interfaz gráfica y que pueda estar disponible en un ambiente web.

Connolly, T. (s.f.). *www.es.wikipedia.com*. Obtenido de <http://es.wikipedia.org/wiki/Cliente-servidor>

BIBLIOGRAFÍA

- Arencibia Cobas, J., & Palma, T. (2012). *Guía práctica de Arquitectura de Información para aplicaciones multimedia educativas*. Recuperado el 11 de 3 de 2015, de nosolousabilidad: <http://www.nosolousabilidad.com/articulos>
- Bass L., C. P. (2005). *Software Architecture in Practice*. Addison-Wesley.
- Conery, R., Hanselman, S., Phil, H., & Guthrie, S. (9 de 10 de 2009). *Application Architecture Guide*. Recuperado el 15 de 3 de 2015, de microsoft.com: <http://msdn.microsoft.com>
- Corporation, I. (2007). *Rational Unified Process*.
- Hassan Montero, Y. M. (2004). *www.hipertext.net*. Recuperado el 18 de 3 de 2015, de <http://www.hipertext.net/web/pag206.htm>
- I. Jacobson, G. B. (2000). *El proceso unificado de desarrollo de software*. Madrid, España: Pearson Education. S.A.
- James-O-Coplien. (s.f.). *www.amazon.com*. Obtenido de <http://www.amazon.com>
- Kruchten, P. (1996). *www.rational.com*. Recuperado el 30 de 10 de 2014, de <http://www.rational.com>
- Lloret Orellana, A. (26 de 09 de 2015). Implementacion fisica de la solucion software. Cuenca, Azuay, Ecuador.
- Milagro, A. P. (10 de 08 de 2011). *Arquitectura en Capas*. Recuperado el 3 de marzo de 2015, de blogstop.com: <http://arquitecturaencapas>
- Montes de Oca Sanchez de Bustamante, A. (2004). *Acimed*. Recuperado el 12 de 11 de 2014, de <http://scieloprueba.sld.cu>
- Pressman, R. S. (2010). Ingeniería del Software: un enfoque practico. En R. S. Pressman, *Ingeniería del Software* (pág. 50). Mexico: McGraw Hill.
- Ronda Leon, R. (2007). *nosolousabilidad*. Recuperado el 11 de 1 de 2015, de <http://nosolousabilidad.com/articulos/diagramacion.htm>
- Rosenfield, L. y. (1999). Information Architecture for the World Wide Web. En L. Rosenfield, *Information Architecture* (2da. ed.).
- Rueda Chacon, J. (3 de 2006). *usac.edu*. Recuperado el 18 de 12 de 2014, de http://biblioteca.usac.edu.gt/tesis/08/08_0308
- S., C. (2004). *Software Project Management*. Recuperado el 14 de 3 de 2015, de www.ibm.com: <http://www.ibm.com>
- Sablon Fernandez, Y., & Hernandez Aballe, D. (2013). *nosolousabilidad*. Recuperado el 12 de 12 de 2014, de <http://nosolousabilidad.com>
- Scott_Ambler. (s.f.). *es.wikipedia.org*. Obtenido de <http://en.wikipedia.org/wiki/>
- Sistemas, I. I. (s.f.). *www.iiis.org*. Obtenido de <http://www.iiis.org>
- Thomas M. Connolly, C. E. (2007). Sistemas de base de datos. En B. C. Connolly T.M., *Sistema de Bases de Datos* (págs. 51-59). Madrid: Pearson.
- Tim_Berners-Lee. (s.f.). *es.wikipedia.org*. Obtenido de <http://es.wikipedia.org>
- Valladolid Zhingre, R. (29 de 10 de 2014). Tesis. Loja, Loja, Ecuador.
- Vargas Cano, A. (12 de 03 de 2014). *wordpress.com*. Recuperado el 18 de 12 de 2014, de <http://aalbertovargasc.files.wordpress.com>
- Articulo sobre Base de Datos Informix. (s.f.). *www.wikispaces.com*. Obtenido de <http://sab15n2.wikispaces.com/files/view/informix.pdf>
- Articulo sobre el proceso del software. (s.f.). *datateca.unad.edu.co*. Obtenido de <http://datateca.unad.edu.co/contenidos>
- Articulo sobre experiencia del usuario. (s.f.). *nosolousabilidad.com*. Obtenido de <http://nosolousabilidad.com>
- Articulo sobre metodologias de sistemas. (05 de 2007). *www.blogspot.com*. Obtenido de <http://metodologiasdesistemas.blogspot.com>

Artículo sobre UML . (s.f.). *www.utdallas.edu*. Obtenido de https://www.utdallas.edu/.../UML.../Rumbaugh--UML_2.0_Reference_C

Hierro, I. C. (25 de 4 de 2014). Reglamento Interno. Cuenca, Azuay, Ecuador.

BIBLIOGRAFÍA

- Arencibia Cobas, J., & Palma, T. (2012). *Guía práctica de Arquitectura de Información para aplicaciones multimedia educativas*. Recuperado el 11 de 3 de 2015, de nosolousabilidad: <http://www.nosolousabilidad.com/articulos>
- Bass L., C. P. (2005). *Software Architecture in Practice*. Addison-Wesley.
- Conery, R., Hanselman, S., Phil, H., & Guthrie, S. (9 de 10 de 2009). *Application Architecture Guide*. Recuperado el 15 de 3 de 2015, de microsoft.com: <http://msdn.microsoft.com>
- Hassan Montero, Y. M. (2004). *www.hipertext.net*. Recuperado el 18 de 3 de 2015, de <http://www.hipertext.net/web/pag206.htm>
- I. Jacobson, G. B. (2000). *El proceso unificado de desarrollo de software*. Madrid, España: Pearson Education. S.A.
- Kruchten, P. (1996). *www.rational.com*. Recuperado el 30 de 10 de 2014, de <http://www.rational.com>
- Milagro, A. P. (10 de 08 de 2011). *Arquitectura en Capas*. Recuperado el 3 de marzo de 2015, de blogstop.com: <http://arquitecturaencapas>
- Montes de Oca Sanchez de Bustamante, A. (2004). *Acimed*. Recuperado el 12 de 11 de 2014, de <http://scieloprueba.sld.cu>
- Pressman, R. S. (2010). Ingeniería del Software: un enfoque practico. En R. S. Pressman, *Ingeniería del Software* (pág. 50). Mexico: McGraw Hill.
- Ronda Leon, R. (2007). *nosolousabilidad*. Recuperado el 11 de 1 de 2015, de <http://nosolousabilidad.com/articulos/diagramacion.htm>
- Rosenfield, L. y. (1999). Information Architecture for the World Wide Web. En L. Rosenfield, *Information Architecture* (2da. ed.).
- Rueda Chacon, J. (3 de 2006). *usac.edu*. Recuperado el 18 de 12 de 2014, de http://biblioteca.usac.edu.gt/tesis/08/08_0308
- S., C. (2004). *Software Proyect Management*. Recuperado el 14 de 3 de 2015, de www.ibm.com: <http://www.ibm.com>
- Sablon Fernandez, Y., & Hernandez Aballe, D. (2013). *nosolousabilidad*. Recuperado el 12 de 12 de 2014, de <http://nosolousabilidad.com>
- Scott_Ambler. (s.f.). *es.wikipedia.org*. Obtenido de <http://en.wikipedia.org/wiki/>
- Sistemas, I. I. (s.f.). *www.iis.org*. Obtenido de <http://www.iis.org>
- Thomas M. Connolly, C. E. (2007). Sistemas de base de datos. En B. C. Connolly T.M., *Sistema de Bases de Datos* (págs. 51-59). Madrid: Pearson.
- Tim_Berners-Lee. (s.f.). *es.wikipedia.org*. Obtenido de <http://es.wikipedia.org>
- Valladolid Zhingre, R. (29 de 10 de 2014). Tesis. Loja, Loja, Ecuador.
- Vargas Cano, A. (12 de 03 de 2014). *wordpress.com*. Recuperado el 18 de 12 de 2014, de <http://aalbertovargasc.files.wordpress.com>
- Artículo sobre Base de Datos Informix. (s.f.). *www.wikispaces.com*. Obtenido de <http://sabd15n2.wikispaces.com/files/view/informix.pdf>
- Artículo sobre el proceso del software. (s.f.). *datateca.unad.edu.co*. Obtenido de <http://datateca.unad.edu.co/contenidos>

- Artículo sobre experiencia del usuario. (s.f.). *nosolousabilidad.com*. Obtenido de <http://nosolousabilidad.com>
- Artículo sobre metodologías de sistemas. (05 de 2007). *www.blogspot.com*. Obtenido de <http://metodologiasdesistemas.blogspot.com>
- Artículo sobre UML . (s.f.). *www.utdallas.edu*. Obtenido de https://www.utdallas.edu/.../UML.../Rumbaugh--UML_2.0_Reference_C
- Corporation, I. (2007). *Rational Unified Process*.
- James-O-Coplien. (s.f.). *www.amazon.com*. Obtenido de <http://www.amazon.com>
- Lloret Orellana, A. (26 de 09 de 2015). Implementación física de la solución software. Cuenca, Azuay, Ecuador.
- Hierro, I. C. (25 de 4 de 2014). Reglamento Interno. Cuenca, Azuay, Ecuador.

GLOSARIO

Vista:	Información y lógica del negocio.
Modelo:	Representación de la información.
Controlador:	Accesos del usuario.
Cliente:	Tipos de recursos disponibles por el usuario.
Servidor:	Dispositivos que proporcionan los recursos.
Reglamento Interno:	Manual de normativas del personal de la Empresa.
Informix:	Producto software del motor de la base de datos.
Unix:	Sistema operativo del entorno software.
4GL:	Lenguaje de programación de cuarta generación.
SQL:	Lenguaje de consulta estructurado.
AI:	Arquitectura de Información.
Middle Ware:	Capa intermedia que ofrece bloques de construcción reutilizables.
SGBD	Sistema de gestión de base de datos
NetTerm	Software de emulación de terminal

ANEXOS

Anexo A: Cronograma anual de vacaciones

CRONOGRAMA DE VACACIONES AÑO 2015 imp. Com. El Hierro Cia. Ltda.																	
Nº	EM	SER	NOMBRES	F.ING.	CARGOS	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBR	OCTUBRE	NOVIEMBR	DICIEMBRE
GERENCIA ADMINISTRAT. - FINANCIERA																	
01	91		Caldas Calle Dora Cecilia	01/07/1990	Gerente Adm. Financ.												
01	992		Cajas Fuela Sandra Maribel	11/08/2006	Asist. Contable												
01	2034		Valencia Guzman Adriana Elizabeth	02/01/2014	Asist. Contable												
01	1897		Marin Morán Verónica Julissa	06/09/2012	Asist. Contab. Contr. Cajas												
01	653		Vintimilla Tenezaca Eva Marina	01/03/2005	Jefe de Contabilidad												
01	2104		Lema Urzúa Hilda Maribel	29/10/2014	Asist. Contabilidad												
01	2118		Sanchez Pesantez Monica Alexandra	23/12/2014	Asist. Contabilidad												
CRONOGRAMA DE VACACIONES AÑO 2015																	
EQ. SUPERVISION DE CAJAS																	
01	212		Alvarez Calle Rosa Hilda	04/04/2005	Supervisora de Cajas												
01	1986		Marin Calle Celia Maria	20/08/2013	Asist. Contab. Contr. Cajas												
01	931		Mestas Lazo Mercy Elizabeth	01/06/2006	Asist. Contab. Contr. Cajas												
01	2073		Tepan Cochancela Daniela Angelica	23/04/2014	Asis. Contab. Contr. Cajas												
01	2101		Sangumma Guaman Esteban Fabricio	03/10/2014	Asis. Contab. Contr. Cajas												
ARCHIVO																	
01	1741		Mora Domanguez Eleana Dolores	09/06/2011	Aux. Archivos												
01	1815		Seminario Rodriguez Mayra Lucia	14/12/2011	Aux. Archivos												
TALENTO HUMANO																	
01	604		Gomez Laine Nancy Magdalena	01/07/2010	Jefa Recursos Humano												
01	49		Lloret Orellana Antonio	01/11/1988	Jefe Seguridad y Salud												
01	1852		Quezada Quezada Sara Beatriz	16/04/2012	Asist. Recursos Humanos												
01	1850		Parra Paredes Victoria Elizabeth	09/04/2012	Médico Laboral												
01	2095		Cedillo Guaman Juan Pablo	08/09/2014	Trabajador Social												
ASESORIA LEGAL																	
01	1964		Carron Reyes Benigno Patricio	04/06/2013	Abogado												
01	2024		Pazmiño Chamorro Juan Andres	19/11/2013	Abogado												
01	957		Rodas Narea Rolando Patricio	10/01/2013	Abogado												
EQ. AUDITORIA INTERNA																	
01	2120		Garcia Monroy Jhonny Leandro	12/01/2013	Asist. Aud Interna												
01	266		Avayaca Flores Cecilia del Camen	04/10/2006	Asist. Aud. Interna												
01	1966		Cardenas Chacha Erick Rusbel	06/06/2013	Asist. Aud. Interna												
01	846		Quito Rivera Marco Vinicio	30/10/2006	Asist. Aud. Interna												
01	951		Rodas Ulloa Juan Rodrigo	26/10/2006	Jefe Auditoria Interna												
EQ. SISTEMAS																	
01	1544		Araujo Ochoa Cayo Fernando	18/01/2010	Asist. Tecnologia												
01	2028		Guachichalca Romero Amibal Manuel	05/12/2013	Asist. Tecnologia												
01	822		Ortega Rivera Alex Xavier	18/11/2009	Asist. Tecnologia												
01	246		Sanchez Bernal Maria Elena	01/07/2009	Jefe de Tecnologia												
01	2123		Tones Tapia Diana Martina	21/01/2013	Asist. Tecnologia												
01	823		Quezada Tecto Edgar Geovany	01/07/2005	Asist. Tecnologia												
CRONOGRAMA DE VACACIONES AÑO 2015 imp. Com. El Hierro Cia. Ltda.																	
Nº	EM	SER	NOMBRES	F.ING.	CARGOS	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBR	OCTUBRE	NOVIEMBR	DICIEMBRE
05	2075		Escobar Velasco Maria de los Angeles	05/05/2014	Cajero Vendedor												
05	1696		Maza Lalao Monica Elizabeth	22/02/2011	Superv. Cajas												
05	1942		Haro Otruno Patricio Bolivar	25/03/2013	Agente Ven. Ferreteria												
05	2106		Llerena Paredes Segundo Oswaldo	04/11/2014	Chofer												
10	55	2002	Munillo Zambrano Eulogio Manuel	12/11/2012	Empacador-Cargador												
11	55	2056	Nieto Fiallos Mario Oswaldo	03/03/2014	Empacador-Cargador												
15	55	1975	Portero Naranjo Klever Eduardo	03/03/2014	Empacador-Cargador												
CRONOGRAMA DE VACACIONES AÑO 2015																	
EM	SER	NOMBRES	F.ING.	CARGOS	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBR	OCTUBRE	NOVIEMBR	DICIEMBRE	
MEGAHIERRO YUNGULLA																	
58	1894		Amijos Valdivieso Luz Vicenta	17/08/2012	Ventas. Mostrador												
58	1764		Cajamarca Morales Jhonny Eduardo	01/08/2011	Subserv. Almacén												
58	958		Chimbo Saquilanda Silvia Ester	11/09/2014	Jefe Almacén												
58	1730		Merchan - Luis Guillermo	21/11/2011	Empacador-Cargador												
58	1779		Monge Chanboga Denis Jacqueline	10/10/2011	Ventas Mostrador												
58	1762		Ortega Barros Jorge Eduardo	25/07/2011	Bodeguero												
58	1798		Suarez Morales Eddy Darwin	14/11/2011	Chofer												
CRONOGRAMA DE VACACIONES AÑO 2015																	
EM	SER	NOMBRES	F.ING.	CARGOS	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBR	OCTUBRE	NOVIEMBR	DICIEMBRE	
MEGAHIERRO GIRON																	
40	1914		Clavijo Araujo Edwin Fabian	23/10/2012	Bodeguero												
40	1910		Pando Pinos Juan Fernando	18/10/2012	Ventas Mostrador												
40	1913		Sanchez Cabrera Jose Victoriano	19/10/2012	Chofer												
40	1918		Vivar Marquez Santiago Agustin	14/11/2012	Empacador-Cargador												
50	2129		Castro Intriago Carlos Lenin	27/01/2015	Jefe Almacén												
CRONOGRAMA DE VACACIONES AÑO 2015																	
EM	SER	NOMBRES	F.ING.	CARGOS	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBR	OCTUBRE	NOVIEMBR	DICIEMBRE	
GUARDIAS DE SEGURIDAD																	
02	893		Amay Juncay Juan Pedro	12/03/2010	Guardia Mega Politécnica												
04	892		Barba Gomez Luis Gonzalo	12/05/2010	Guardia Mega Paute												
02	811		Carron Edison Patricio	23/06/2010	Guardia Mega Politécnica												
02	2037		Cajon Saldaña Luis Enrique	16/01/2014	Guardia Mega Politécnica												
02	2040		Garcia Carabao Cristhian Jovanny	03/02/2014	Guardia Mega Politécnica												
02	1872		Giron Rosales Richard Francisco	18/06/2012	Guardia Local Principal												
03	1637		Pacheco Pinto Eduardo Fabian	02/09/2011	Guardia Mega Arenal												
02	502		Palacios Bravo Manuel Jesus	24/07/2002	Guardia Mega Principal												
01	490		Valdez Bustamante Mauro Antonio	01/07/2002	Guardia Mega Machala												
10	05	1766	Vazquez Cedillo Kleber Teodoro	28/02/2009	Guardia local el Arenal												
11	01	1462	Vazquez Guarachea Juan Carlos	18/04/2009	Guardia Local Principal												
Elaborado por: Departamento de Talento Humano																	
IMP. COM. EL HIERRO CIA. LTDA. EXT. 7241-7242-7243 Fecha: 5-enero-2015																	

Figura 23. Cuadro anual de vacaciones

Fuente: (Imp.Com.El Hierro, 2015).

Elaborado por: Lloret Antonio.

Para mayor detalle del cronograma abrir hoja de cálculo Anexo A.xls

Anexo B: Formulario de solicitud de vacaciones

01
A.

SOLICITUD Y TRÁMITE DE PERMISO CON CARGO A VACACIONES 1544		
LUGAR Y FECHA:	<u>Cuenca, 04 de marzo de 2015</u>	
EMPLEADO:	<u>Ing. Cayo Fernando Araujo Ochoa</u> CARGO: <u>Asis. de Tecnología</u>	
LOCAL:	<u>Principal</u>	DEPARTAMENTO: <u>Sistemas</u>
GRUPO DE TRABAJO:	DÍAS LIBRES:	
SOLICITO SE ME CONCEDA HACER USO DE... <u>2</u> ...DÍAS, CON CARGO A VACACIONES DESDE EL... <u>05</u> ...DE... <u>marzo</u> ...DE... <u>2015</u> ...HASTA EL... <u>06</u> ...DE... <u>marzo</u> ...DE... <u>2015</u> ...		
		n)  SOLICITANTE
ANALIZADAS LAS NECESIDADES DEL PERSONAL, PARA EL NORMAL DESENVOLVIMIENTO DE LAS FUNCIONES DEL ÁREA, SE DETERMINA QUE LA PRESENTE SOLICITUD DE PERMISO CON CARGO A VACACIONES ES:		
FECHA DE INGRESO EN LA EMPRESA: ... <u>18 de Enero de 2010</u> ...		
Aceptado del <u>05</u> de <u>marzo</u> del <u>2015</u> hasta el <u>06</u> de <u>marzo</u> del <u>2015</u>		
EN SU AUSENCIA SERÁ REEMPLAZADO POR: <u>Ing. Disma Torres</u>		
LUGAR Y FECHA:	<u>2 días</u>	
		n)  JEFE DEPARTAMENTAL
DÍAS USADOS:	DÍAS POR USAR:	
	<u>1</u>	 <u>5/ Mar/ 15</u>

Figura 24. Solicitud de vacaciones.
Fuente: (Imp.Com. El Hierro, 2015).
Elaborado por: Lloret Antonio.

Anexo C: Reglamento interno

bajo igual en horas distintas de las del turno ordinario.

CAPITULO III. DE LOS REGISTROS INDIVIDUALES.

Artículo 24.- Registros Individuales.- El empleador llevará un Registro individual de todos los trabajadores, en el que se hará constar obligatoriamente, el nombre, número de cédula de identidad, edad, procedencia, estado civil, cargas familiares y demás información necesaria, realizando actualizaciones periódicas a dicho registro, en el cual se hará constar también, las novedades sobre el desempeño de las obligaciones que deben cumplir los trabajadores, el mismo que servirá como referencia para posteriores ascensos o estímulos.

Artículo 25.- Novedades laborales.- Cada vez que se requiera imponer una sanción de la naturaleza que fuere,

será comunicada por escrito al trabajador y esta constancia insertada en el correspondiente Registro Individual, el cual servirá como sustento para las consecuencias que deriven del incumplimiento del contrato laboral, y se procederá de conformidad a lo que establece en este reglamento interno

Artículo 26.- Custodia de los Registros.- Este Registro y toda la documentación relacionada con los asuntos contractuales, estarán a cargo del Gerente, el jefe del Departamento de Talento Humano o su delegado.

Artículo 27.- Función del Registro.- La Empresa establecerá sistemas de registro individual para controlar la asistencia y puntualidad. El trabajador se someterá a dicho control, y esta obligado a realizar las marcaciones correspondientes en las que se registraran horas de entrada y de salida del trabajador a sus labores en la empresa. Estas marcaciones serán la única constancia de que el trabajador se presentó a laborar. Los datos del registro serán

prueba para las acciones administrativas, civiles y de otra índole.

CAPITULO IV. DE LAS VACACIONES.

Artículo 28.- Cronograma anual de Vacaciones.- Según lo establecido en la Constitución del Ecuador y Código de Trabajo, los trabajadores tienen derecho a gozar de vacaciones anuales, las mismas que serán establecidas en un cronograma elaborado por la Compañía, el mismo que será dado a conocer a todos los trabajadores por parte del Departamento de Talento Humano.

Artículo 29.- Diferimiento de Vacaciones por parte del Trabajador.- Cuando el trabajador por cualquier causa desee diferir su periodo de vacaciones, deberá presentar una solicitud por escrito dirigido al jefe inmediato, por triplicado, en donde la una copia se dirigirá a la jefatura del departamento de Talento Humano y la otra a Gerencia

Figura 24. Extracto del reglamento interno.
Fuente:(Reglamento Interno Imp.Com. El Hierro, 2013).
Elaborado por: Lloret Antonio.

Para mayor detalle del Reglamento Interno se adjunta documento.

Anexo D: Documento de visión

**Desarrollo de una solución software empresarial para control de vacaciones anuales
en la Empresa Importadora Comercial El Hierro Cía. Ltda.**

Documento de Visión

Versión 1.1

Historial de Revisiones

Fecha	Versión	Descripción	Autor
15-ABRIL-2015	1.0	Versión Inicial planteada como propuesta de implementación.	Antonio Lloret Orellana
15-JUNIO-2015	1.1	Correcciones realizadas al documento	Antonio Lloret Orellana
6-AGOSTO-2015	1.2	Correcciones realizadas al documento	Antonio Lloret Orellana
21-SEPT-2015	1.3	Correcciones realizadas al documento	Antonio Lloret Orellana

1.1. Introducción del documento visión.

El propósito del documento visión es recolectar, analizar y definir los requerimientos de alto nivel y presentar las características principales del proyecto “Desarrollo de una solución software empresarial para control de vacaciones anuales en la Empresa Importadora Comercial El Hierro Cía. Ltda.” ; ubicada en la ciudad de Cuenca.

El alcance de este proyecto se circunscribe al desarrollo de una aplicación de escritorio que permita el registro y control de vacaciones anuales de los empleados y trabajadores de Imp. Com. El Hierro, de manera oportuna, confiable, ágil, mejorando los tiempos de respuesta a las solicitudes de los empleados de la Empresa.

1.2. Conceptos y abreviaturas.

Tabla 19. Conceptos y abreviatura del documento de visión.

Abreviatura	Concepto
Imp.Com. El Hierro	Compañía Limitada Importadora Comercial El Hierro
UML	Lenguaje Unificado de Modelado
RUP	Rational Unified Process
SGBD	Sistema de Gestión de Base de Datos.

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

1.3. Posicionamiento y oportunidades de negocio.

Este proyecto de tesis permitirá a la Empresa Imp. Com. El Hierro Cía. Ltda.; optimizar su sistema general de rol de pagos, especialmente en lo que se refiere a tener un mejor control sobre la asignación y consulta de vacaciones anuales de su nómina, reduciendo en forma considerable el tiempo del personal de talento humano en recoger la información de cada empleado que solicita hacer uso de sus vacaciones, liquidación y/o pago de las mismas, además de facilitar en forma oportuna las fechas y periodos de vacaciones de sus empleados y trabajadores.

1.4. Sentencia que define el problema.

En la Tabla 20 se define la problemática actual.

Tabla 20. Definiciones del problema.

Definición	Descripción
El problema de	Tiempo que ocupa al asistente de talento humano elaborar, asignar, regular, verificar el cronograma anual de vacaciones desde la hoja de cálculo de Excel. Tiempo “muerto” en llenar las solicitudes a ser entregadas a los empleados para la aprobación de las mismas por parte de sus jefes inmediatos. El proceso que tarda en llenar la solicitud, verificar los datos, comunicar el pedido, aprobar y notificar la solicitud, puede alcanzar entre 1 a 3 días dependiendo del número de solicitudes a tramitar.
Afecta a	Departamento de Talento Humano, departamento de sistemas que genera la nómina de pagos, departamento de contabilidad que registra el asiento contable en caso de pago y a los empleados en general que les impide programar sus vacaciones con anterioridad.
El impacto asociados	Demora en el registro y control de vacaciones en el departamento de sistemas y talento humano, lo cual causa retraso en la programación de actividades de la Empresa.
Una solución ideal sería	Implementar una solución automatizada que permita registrar y convalidar la información del empleado respecto a sus vacaciones acumuladas, de manera oportuna y confiable, lo cual generaría seguridad e integridad de la información, además que se integra al módulo principal de rol de pagos de la empresa.

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

1.5. Sentencia que define la posición del producto

Tabla 21. Posicionamiento del producto.

Definición	Descripción
Para	Los asistentes del Dpto. Talento Humano que dispondrán de un mejor servicio para consultar y registrar las vacaciones.
Quienes	Los empleados y trabajadores que laboran en la Empresa.
Nombre del producto	Solución software que permita automatizar el registro y consulta de vacaciones por parte del departamento de talento humano.
Que	Permita el registro y consulta de vacaciones anuales de los empleados de la empresa de manera ágil, oportuna y segura.
No como	Registrar en hojas de papel de forma manual y no confiable, desperdiciando recursos y tiempo valioso del personal de la Empresa.
El producto	Permitirá registrar y consultar de manera oportuna, rápida y eficiente, guardando toda la información en la base de datos de la Empresa.

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

1.6. Descripción de stakeholders (actores y usuarios del proyecto)

Uno de los puntos importantes dentro del análisis de requerimientos del proyecto es involucrar, identificar y ubicar a los participantes y usuarios del mismo; para ello se presenta un perfil de cada uno de ellos, de igual forma se detalla los problemas que ellos perciben con el objetivo de propender una mejor solución del proyecto y justificar el porqué de implementar los requisitos que validen la funcionabilidad del proyecto.

1.7. Resumen de stakeholders

En la siguiente Tabla 22 se presenta la información referente a los stakeholders en función de sus responsabilidades en el registro y control de vacaciones.

Tabla 22. Resumen de Stakeholders.

Nombre	Descripción	Responsabilidades
Ing. Patricio Matute Zamora.	Gerente General de Imp. Com. El Hierro Cía. Ltda.	Dar seguimiento del cumplimiento del proyecto, autorizar y asignar los recursos económicos y tecnológicos; aprobar requerimientos y funcionalidades del proyecto.
Ing. María Elena Sánchez B.	Jefe de Sistemas	Proveer información del sistema actual; asesorar y aprobar los requisitos y funcionalidades del proyecto, supervisar el cumplimiento del proyecto.

Antonio Lloret O.	Jefe de Seguridad	Coordinar autorización y uso de documentación y accesos al sistema de la Empresa.
Nancy Gómez L.	Jefe de Talento Humano	Responsable de la supervisión, ejecución y asignación de la información requerida para la culminación del proyecto.
Ing. Sara Quezada Q.	Asistente de Talento Humano	Recolectar, ingresar, validar, consultar y generar el control y registro de vacaciones de los empleados de la Empresa.

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

1.8. Perfil de los stakeholders.

1.8.1. Representante de Gerencia y Operación.

Tabla 23. Perfil del Gerente General.

Representante:	Jhonny Patricio Matute Zamora Ing.Com
Descripción:	Gerente General de Imp.Com. El Hierro Cía. Ltda.
Tipo:	Gerente General
Responsabilidades:	<ul style="list-style-type: none"> ➤ Nombrar a los responsables de las Gerencias de Talento Humano, Financiero, Contabilidad, Sistemas, Crédito, Legal y otras dependencias. ➤ Coordinar y supervisar el cumplimiento del Plan Estratégicos de la Empresa. ➤ Revisar que se cumplan con los principios de administración y contabilidad de aceptación general. ➤ Velar por la aplicación de la norma y el cumplimiento de las normas de la organización. ➤ Cuidar el resguardo y mantenimiento de la confidencialidad de la información suministrada por la organización y por el cliente. ➤ Responsable en cumplimiento de las funciones y objetivos del área Administrativa, Financiera, Contable, Legal y Sistemas de la Empresa. ➤ Aprobar y asignar los requisitos funcionales y no funcionales de los proyectos en ejecución.
Criterio de éxito:	Permanente búsqueda de herramientas que permitan mejorar y optimizar el servicio al cliente y el bienestar de sus colaboradores.
Grado de participación:	<ul style="list-style-type: none"> ➤ Asignar los recursos económicos y tecnológicos que sean necesarios para la implementación efectiva del presente trabajo de titulación. ➤ Efectuar el seguimiento respectivo para el cumplimiento del proyecto. ➤ Aprobar requisitos y funcionalidades del proyecto.

Comentarios: El gerente general se constituye en el stakeholder principal para la aprobación del proyecto.

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Tabla 24. Perfil del Jefe de Sistemas.

Representante:	Maria Elena Sanchez Bernal Ing.Sistemas.
Descripción:	Jefe de Sistemas de Imp.Com. El Hierro Cía. Ltda.
Tipo:	Jefe de Sistemas.
Responsabilidades:	<ul style="list-style-type: none">➤ Aprobar y asignar los requisitos funcionales y no funcionales de los proyectos en ejecución.➤ Coordinar y supervisar el cumplimiento del Proyecto presentado.➤ Cuidar el resguardo y mantenimiento de la confidencialidad de la información suministrada por la Empresa y por el cliente.➤ Planifica, organiza y controla la implementación de los sistemas informáticos y evalúa el desarrollo de los mismos.➤ Brinda apoyo y asesoría al desarrollador del proyecto.➤ Verifica que se cumplan las asignaciones en las fechas establecidas cumpliendo con las normas y procedimientos.
Criterio de éxito:	Garantiza que la base de datos cuente con los controles y seguridad adecuada.
Grado de participación:	<ul style="list-style-type: none">➤ Evalúa la factibilidad del proyecto, sus alcances y objetivos.➤ Asigna tareas y recursos para su desarrollo.➤ Revisa el proyecto en sus diferentes niveles de desarrollo.➤ Recibe la solicitud y organiza reuniones de trabajo con el autor del proyecto para determinar los requerimientos del sistema.
Comentarios:	El Jefe de Sistemas es el stakeholder clave para la implementación del Proyecto.

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Tabla 25. Perfil del Jefe de Seguridad.

Representante:	Antonio Lloret Orellana Profesional en formacion.
Descripción:	Jefe de Seguridad de Imp.Com. El Hierro Cía. Ltda.
Tipo:	Jefe de Seguridad.
Responsabilidades:	<ul style="list-style-type: none">➤ Verifica que se cumplan las asignaciones en las fechas establecidas cumpliendo con las normas y procedimientos.➤ Supervisa periódicamente los recursos disponibles de los equipos y su uso adecuado.

- Informa a los directivos de la Empresa sobre el status de las actividades que se realizan.

Criterio de éxito: Garantiza que las labores de desarrollo cuente con los controles y seguridad adecuada.

Grado de participación:

- Vigila que el personal cumpla con su horario, normas de permisos, etc.
- Registra la asistencia diaria del personal.

Comentarios: Ninguno.

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

Tabla 26. Perfil del Jefe de Talento Humano.

Representante: **Nancy Gomez Laime Profesional en formacion.**

Descripción: Jefe de Talento Humano de Imp.Com. El Hierro Cía. Ltda.

Tipo: Jefe de Talento Humano.

Responsabilidades:

- Conducir el proceso de selección de personal, inducción, capacitación y bienestar laboral y económico de los empleados de la Empresa.
- Asesorar y participar en la formulación de la política de personal de la Empresa.
- Controlar el cumplimiento del rol vacacional de los empleados de la Empresa.
- Preparar y mantener los registros de vacaciones del personal de la Empresa.

Criterio de éxito: Asegurar el cumplimiento de las normas legalmente establecidas.

Grado de participación:

- Planifica, coordina y supervisa el desarrollo del proyecto.
- Designa el personal que estará involucrado en el desarrollo del proyecto.
- Gestionar con el departamento de Talento Humano y Sistemas que se entregue la información necesaria del proceso actual de registro de vacaciones de la Empresa.

Comentarios: Ninguno.

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

Tabla 27. Perfil del Asistente de Talento Humano.

Representante: **Sara Quezada Q. Ingeniera Comercial.**

Descripción: Asistente de Talento Humano de Imp.Com. El Hierro Cía. Ltda.

Tipo:	Asistente de Talento Humano.
Responsabilidades:	<ul style="list-style-type: none"> ➤ Dar seguimiento diario de las solicitudes de vacaciones de los empleados de la Empresa. ➤ Registrar las solicitudes y novedades del proceso de registro de vacaciones del personal de la Empresa. ➤ Comunicar a los jefes departamentales y directivos de la Empresa sobre la aprobación de vacaciones asignadas al personal de la Empresa. ➤ Elaborar los informes de personal que se encuentran en periodo de vacaciones y los empleados que se encuentran en proceso de aprobación.
Criterio de éxito:	Asegurar y optimizar que el proceso de control de vacaciones se cumpla dentro del cronograma establecido y de acuerdo a los requerimientos y necesidades de la Empresa.
Grado de participación:	<ul style="list-style-type: none"> ➤ Elaborar el cronograma anual de vacaciones, entregar los formularios de solicitud de vacaciones. ➤ Ingresar diariamente las solicitudes de vacaciones. ➤ Ingresar en el sistema informático de la Empresa las novedades respecto al ingreso, salida del personal de la Empresa. ➤ Imprimir y comunicar el resultado de las vacaciones a los empleados que realizaron la solicitud respectiva.
Comentarios:	Ninguno.

Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

2. Descripción global del producto.

2.1. Perspectiva del producto.

La solución software a implementar se basa en un producto desarrollado en lenguaje de programación 4gl que tiene como propósito mejorar el control de vacaciones de los empleados y trabajadores de Imp. Com. El Hierro, agilizando la asignación de vacaciones en función de un cronograma previamente establecido, reduciendo los tiempos de aprobación de los mismos, así como de disponer de una información oportuna y confiable, y finalmente eliminar la posibilidad de pérdida del registro de datos, en virtud de que se elimina el registro manual.

2.2. Resumen de características.

Seguidamente se presenta algunos de los beneficios que la solución software aspira a cumplir en respuesta a las necesidades de los empleados y directivos de la Empresa.

Tabla 28. Beneficios de la solución software a implementar:

Beneficios del empleado	Características que lo apoyan
-------------------------	-------------------------------

Optimizar el tiempo de respuesta en la recepción de solicitudes de vacaciones de los empleados de la empresa	Normalización del formato de solicitud de vacaciones de forma estándar y que permita su fácil interpretación y registro.
Consulta rápida del estado actual de disponibilidad de vacaciones al momento de asignar y aprobar la solicitud.	Aplicación automatizada que verifica los días disponibles que tiene derecho el empleado.
Atención oportuna al momento de brindar información sobre los días disponibles de vacaciones que tiene el empleado.	Aplicación automatizada que asigna con seguridad y confiabilidad los días que tiene derecho el empleado en función de sus años de servicio.
Conocimiento del historial de vacaciones que registra el empleado.	Aplicación automatizada que muestra el registro de vacaciones que tiene utilizado y por utilizar el empleado.

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

2.3. Presupuesto y costo del producto.

En la Tabla 29 se presenta los costos referenciales para el desarrollo e implementación de la solución software; los mismos que han sido financiados por la Gerencia de la Empresa y el autor del proyecto.

Tabla 29. Costos referenciales de la implementación de la solución software:

Descripción	Costo del proyecto
Horas programadas para desarrollo de la aplicación.	USD. 2.500,00
Consumo por utilización de la infraestructura tecnológica	USD. 1.000,00
Gastos de útiles de oficina e impresión.	USD. 500,00
Otros gastos	USD. 1.000,00
TOTAL	USD. 5.000,00

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

La infraestructura de tecnología y herramientas de software son activos de la Empresa Imp. Com. El Hierro, de igual manera las horas de programación y asesoramiento del personal de sistemas de la Empresa son asumidos directamente por la Gerencia General de la Empresa.

3. Restricciones.

- La hoja electrónica de cronograma anual de vacaciones se elabora en base a un reporte independiente del sistema de nómina de rol de pagos que dispone la Empresa.
- Los formularios de solicitud de vacaciones son elaborados por el personal de Talento Humano con formatos pre establecidos del utilitario Microsoft Office.

4. Precedencia y prioridad.

- Para la obtención de la información primaria referente a los días disponibles que tiene los empleados de la Empresa, se deberá recurrir a los registros manuales individuales que reposan en el archivo de documentación física de la Empresa.
- Investigación previa para la selección de la metodología y arquitectura de software a utilizar en la implementación y desarrollo del proyecto.
- Selección del tipo de tecnologías a aplicar en el desarrollo de la solución software.
- Integración del módulo de control y registro de vacaciones anual al sistema integrado de base de datos que dispone la Empresa.
- Ejecución de la aplicación de escritorio que permita consultar y posteriormente asignar de manera automática el periodo de vacaciones a que tienen derecho dentro de la normativa legal vigente los empleados de la Empresa Imp. Com. El Hierro.

5. Requisitos del producto.

A continuación se presenta requisitos esenciales para el desarrollo de la solución software:

- El desarrollo e implementación del proyecto deberá cumplir con todos los requerimientos funcionales previamente establecidos por los involucrados en el proyecto.
- Los programas y módulos que conforman el presente proyecto deberán adaptarse al cronograma y presupuesto establecido.
- La implementación del proyecto se realizara utilizando las licencias de uso legal de software que dispone la Empresa como son Sistema Operativo Windows y Red Hat Linux, motor de la base de datos Informix, utilitarios Microsoft Office, emulador Netterm, VirtualBox, etc.
- La acta previa de entrega y recepción del producto verificara que el proyecto se encuentre implementado, integrado y funcionando a satisfacción del cliente.

5.1. Estándares aplicables.

Se sujetaran de acuerdo a las siguientes regulaciones:

- El control de calidad de software del proyecto se determinara en función de la satisfacción de los requerimientos y necesidades de los usuarios implicados en el proyecto.
- Se aplicara la metodología de desarrollo de software RUP, con el propósito de que se cumpla todas las fases y entregables de la solución software.

- Se utilizara la arquitectura Cliente-Servidor con el fin de brindar el soporte adecuado y eficiente para el óptimo desempeño de la solución software.
- Se aplicaran estándares probados en la industria de desarrollo de software para garantizar el funcionamiento de la solución software.

5.2. Requisitos del sistema.

Los requisitos de desarrollo de la solución software se complementaran en base a lo siguiente:

- El aplicativo se construirá en lenguaje de alto nivel, sobre el motor de la base de datos Informix.
- El acceso, registro, y envío de datos entre el servidor de la base de datos y la estación de trabajo se realizara por medio del emulador NetTerm que es una interface de comunicación entre la base de datos y el sistema operativo.

6. Requisitos de documentación.

6.1. Manual de usuario.

La documentación relacionada con el manual de usuario se define dentro de la etapa de transición y se podrá revisar en el Anexo I.

6.2. Guía de instalación y configuración.

La guía de instalación y configuración de la solución software será definida en el Manual de usuario descrito en el Anexo I.

7. Atributos y características del producto software.

Los atributos y características que configuran el producto software se basan en:

- Interfaz amigable con el usuario
- Facilidad de uso
- Características de seguridad e integridad
- Aplicación de escritorio que cumple con las normativas de calidad de la arquitectura de software seleccionada.

Anexo E: Documento de especificación de requerimientos

Desarrollo de una solución software empresarial para control de vacaciones anuales en la Empresa Importadora Comercial El Hierro Cía. Ltda.

Documento de Especificación de Requerimientos

Versión 1.1

Historial de Revisiones

Fecha	Versión	Descripción	Autor
15-ABRIL-2015	1.0	Versión Inicial planteada como propuesta de implementación.	Antonio Lloret Orellana
15-JUNIO-2015	1.1	Correcciones realizadas al documento	Antonio Lloret Orellana
6-AGOSTO-2015	1.2	Correcciones realizadas al documento	Antonio Lloret Orellana
21-SEPT-2015	1.3	Correcciones realizadas al documento	Antonio Lloret Orellana

1.1. Introducción del documento especificación de requerimientos.

El propósito del documento Especificación de Requerimientos es analizar y entender aquellos requerimientos tanto del cliente como los del usuario manifiesten; de tal manera que los mismos se logren concretar en la solución software; así mismo este documento se constituye en una información de gran utilidad para conseguir definir en forma concreta y precisa los requerimientos funcionales y no funcionales que el cliente pretende obtener.

1.2. Personal que interviene.

El personal involucrado que interviene en el diseño, desarrollo y ejecución de la solución software se detalla a continuación:

Tabla 30. Rol del autor del presente proyecto.

Nombre: Antonio Lloret Orellana

Rol:

- Gestor del proyecto
- Analista de requerimientos
- Programador
- Diseñador de la Base de Datos

Categoría profesional: Analista

Responsabilidades:

- Análisis y especificación de requerimientos
- Diseño de la arquitectura de la solución software
- Programador de los módulos de la solución software
- Diseño de la Base de datos

Información de contacto: aloret@megahierro.com

Fuente: (Lloret Antonio, 2015).
Elaborado por: Lloret Antonio.

1.3. Definiciones, acrónimos y abreviaturas

Tabla 31. Definiciones, acrónimos y abreviaturas del documento de ER.

Abreviatura	Definición
SGBD	Sistema de Gestión de la Base de Datos
DBA	Administrador de la Base de Datos
RUP	Rational Unified Process
TCP IP	Protocolo de Internet Protocolo de Control de Trasmisión
Informix	Motor de la Base de datos
FTP	Protocolo de transferencia de archivos
NetTerm	Emulador de texto

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio. (Artículo sobre Eficiencia de la Base de Datos Informix)

2. Descripción General.

2.1. Funcionalidades de la solución software.

Los requerimientos funcionales hacen referencia al comportamiento interno del sistema; es decir se refieren a los parámetros de configuración, cálculos de tipo numérico, validaciones de campos claves de la base de datos, detalles técnicos y otras especificaciones; adicional a ello se describe características de aplicación de los casos de uso en la práctica, y se registra el nivel de prioridad que desempeñan en la solución software.

A continuación en la Tabla 32 se presenta un resumen de prioridades.

Tabla 32. Requisitos funcionales de la solución software.

CÓDIGO RF.	REQUISITO	PRIORIDAD
RF01	Consultar Cronograma Anual	BAJA
RF02	Recepción de solicitudes	MEDIA
RF03	Autenticación de usuarios	MEDIA
RF04	Control de vacaciones	MEDIA
RF05	Registro de Vacaciones	ALTA
RF06	Anulación de registros	ALTA

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

Inicialmente partimos con RF01 que consiste en leer el cronograma anual de vacaciones como referencia para el siguiente proceso que es RF02, el cual consiste en recibir todas las solicitudes de vacaciones para proceder con su validación y aprobación, luego se desarrollara el RF03 que consiste en permitir el acceso a la aplicación de la asistente de talento humano

para el manejo del aplicativo, seguidamente se procede con el control de vacaciones que permite visualizar información específica del empleado respecto a su historial de vacaciones, posteriormente se sigue con RF05 que es propiamente el registro de vacaciones asignado al empleado y finalmente RF06 que permite anular un registro de vacaciones previamente registrada y generada.

2.2. Evolución predecible del proyecto.

El avance de la solución software se medirá conforme las iteraciones que se realice en cada una de las fases de la metodología de desarrollo de software, hasta conseguir una solución software que este de conformidad con los requerimientos del cliente, la misma que deberá estar sustentada en la correcta aplicación de la metodología de desarrollo RUP y la implementación de la arquitectura de software seleccionada.

2.3. Requisitos comunes de las interfaces.

2.3.1. Interfaz de usuario.

La interfaz del usuario deberá estar relacionada con las pantallas, ventanas y formularios que debe manipular el usuario para realizar una acción determinada, lo cual permitirá al usuario desenvolverse con facilidad en su entorno de trabajo, estas características de las interfaces incluyen lo siguiente:

PF: Se refieren en un ambiente de texto a las teclas de función que se encuentran incorporadas en el teclado del computador y que permiten al usuario ejecutar una acción determinada tipo comando, también se utiliza como cuadros informativos relacionados a una ayuda sobre la acción a ejecutarse.

MENU: Es una caja de texto de la interfaz que permite escoger de una lista de opciones que puede desplazarse en forma horizontal para visualizar más opciones y de esta manera acceder al aplicativo para registrar una operación.

SUBMENU: Parte de un menú principal, manteniendo un orden jerárquico en función de su prioridad o uso.

2.3.2. Interfaz de hardware.

TECLADO: Permite interactuar a través del computador con las bondades del aplicativo, las mismas que se adaptan de acuerdo a la programación del aplicativo, pues permite el uso de teclas especiales y funciones programadas.

2.3.3. Interfaz de software y comunicación.

La aplicación de escritorio dispone de un software de emulación denominada NetTerm que permite realizar el enlace entre el sistema operativo del computador y el sistema operativo del servidor. El proceso de comunicación entre los componentes de la solución software se genera mediante la aplicación del patrón arquitectónico Cliente-Servidor.

2.4. Descripción de requisitos funcionales.

2.4.1. Requisito funcional 1 (RF01).

Introducción:

Antes de ingresar al aplicativo; el Asistente de Talento Humano mediante la apertura del archivo generado en la hoja electrónica Excel procederá a leer la información relativa al cronograma anual de vacaciones.

Entradas:

Ejecución de la aplicación de Microsoft Office Excel

Procesos:

Verificar inicialmente que el archivo de Excel corresponda al cronograma anual de vacaciones del año actual y se encuentre actualizada según el último reporte entregado por el departamento de sistemas del módulo principal de rol de pagos.

Salidas:

Mensaje: Fecha de actualización último mes vigente [mes actual].

2.4.2. Requisito funcional 2 (RF02).

Introducción:

Antes de ingresar al aplicativo; el Asistente de Talento Humano procederá con la emisión y recepción del formulario de solicitud de vacaciones para que sea llenado por el empleado y aprobado por el Jefe departamental o directivo autorizado.

Entradas:

Ejecución de la aplicación de Microsoft Office Word.

Procesos:

Verificar que el documento este autorizado y aprobado; además cumpla con los requerimientos de orden legal.

Salidas:

Mensaje: Firmas de conformidad.

2.4.3. Requisito funcional 3 (RF03).

Introducción:

La aplicación de escritorio permitirá la autenticación del usuario previamente registrado en la base de datos del sistema principal para acceder al módulo de control de vacaciones.

Entradas:

Ingreso de la siguiente información:

Login: Username del usuario.

Password: clave secreta del usuario.

Procesos:

El aplicativo verificara el login y password del usuario que le permita acceder al módulo de control de vacaciones.

Salidas:

Si el login y password son correctos, el usuario podrá ingresar al aplicativo de lo contrario presentara el siguiente mensaje:

- Login o password incorrecto.

2.4.4. Requisito funcional 4 (RF04).

Introducción:

Una vez que el usuario ha ingresado en la aplicación de escritorio, podrá visualizar un Menú principal de la aplicación, donde podrá navegar por las diferentes opciones del aplicativo que le permitirá consultar y registrar información del empleado referente a sus vacaciones.

Entradas:

Información que se obtiene del Menú Principal del Sistema Central:

Procesos:

El aplicativo permite al usuario en primer lugar seleccionar la Empresa; seguidamente escoger la opción de rolpago y la Empresa que pertenece el empleado; luego se presenta la opción de Mantenimiento, una vez escogida esta opción le permite seleccionar el modulo Control Vacaciones en donde puede finalmente consultar o ingresar la información referente a vacaciones.

Salidas:

El usuario puede seleccionar la opción de Salir para retornar al Menú anterior.

Mensaje: Salir.

2.4.5. Requisito funcional 5 (RF05).

Introducción:

Una vez que el usuario ha seleccionado la opción Ingresar, el usuario podrá registrar las solicitudes de vacaciones previamente autorizadas y suministradas por el departamento de Talento Humano.

Entradas:

Información que se obtiene de la Base de Datos:

- Nombres del Empleado
- Grupo de Trabajo
- Estado del Empleado
- Cargo que ocupa el empleado
- Fecha de ingreso del empleado
- Fecha de salida del Empleado
- Serial del empleado
- Tipo de transacción
- Fecha d
- Fecha Fin
- Periodo vacaciones
- Ingreso
- Egreso
- Saldo
- Estado

Información que debe ingresar el usuario:

- Serial del empleado
- Tipo de transacción

Procesos:

El aplicativo permite al usuario registrar las solicitudes ingresando el serial del empleado e ingresando el número de días autorizados a tomar vacaciones; el aplicativo calculara y validara automáticamente en función del tipo de transacción, los días asignados, los días ya tomados y el saldo disponible.

Salidas:

El usuario puede seleccionar la opción de Salir para retornar al Menú anterior.

Mensaje: Registrar una nueva transacción.

2.4.6. Requisito funcional 6 (RF06).

Introducción:

Una vez que el usuario ha seleccionado la opción Buscar, el usuario podrá anular las solicitudes de vacaciones y el histórico del empleado registrado en la base de datos de la Empresa.

Entradas:

Información que se obtiene de la Base de Datos:

- Nombres del Empleado
- Grupo de Trabajo
- Estado del Empleado
- Cargo que ocupa el empleado
- Fecha de ingreso del empleado
- Fecha de salida del Empleado
- Serial del empleado
- Tipo de transacción
- Fecha d
- Fecha Fin
- Periodo vacaciones
- Ingreso
- Egreso
- Saldo
- Estado

Información que debe ingresar el usuario:

- Serial del empleado

Procesos:

El aplicativo permite al usuario revisar las solicitudes ingresando el serial del empleado, o también puede ingresar el nombre del empleado; el aplicativo mostrara toda la información relacionada del empleado en cuanto a su disponibilidad de vacaciones y tendrá la oportunidad de actualizar la información.

Salidas:

El usuario puede seleccionar la opción de Salir para retornar al Menú anterior.

Mensaje: transacción actualizada.

Una vez explicado los requisitos funcionales, se procede a presentar los requisitos no funcionales, los cuales nos proporcionan propiedades tales como fiabilidad, integridad, seguridad, tiempos de respuesta y capacidad de procesamiento que dispone el aplicativo.

2.5. Requisitos no funcionales.

2.5.1. Requisitos de seguridad.

- El acceso a la base de datos será mediante el uso del software de emulación NetTerm, lo que supondrá mantener una única instancia de acceso a la base de datos.
- La conexión a la base de datos será persistente, en función de las características propias del motor de la base de datos, el número de usuarios que pueden interactuar al mismo tiempo es totalmente manejable.
- Se dispondrá de dispositivos de hardware que controlen y supervisen mediante políticas de acceso el ingreso de usuarios autorizados por medio de Firewall y protocolos de comunicación.

2.5.2. Requisitos de rendimiento y escalabilidad.

- **Mayor flexibilidad:** Informix ofrece más opciones y oportunidades de escalar las capacidades y mejorar la gestión y utilización de los recursos. Más aún, ofrece nueva interoperabilidad con diversas herramientas de código abierto, ofreciendo numerosas oportunidades para el desarrollo de aplicaciones.
- **Mayor fiabilidad:** Informix ofrece una mejor autonomía y mayor automatización, incluso durante la recuperación en caso de errores.
- **Mayor rendimiento:** Con Informix, las transacciones empresariales se ejecutan con mayor rapidez y las consultas analíticas se realizan hasta un 50% más rápido que antes.
- **Mayor eficiencia:** Con la capacidad de utilizar mejor las infraestructuras heterogéneas existentes, un programador de tareas automatizado, mayor rendimiento y requisitos de administración aún menores, Informix ofrece un nivel de eficiencia mayor.²⁵

2.5.3. Requisitos de disponibilidad.

- Compatibilidad con múltiples plataformas y normas standard abiertas como por ejemplo los servicios web, Linux, el procesamiento autónomo y el comercio electrónico.
- La aplicación de escritorio Control de Vacaciones estará disponible en cualquier equipo Pc con sistema operativo Windows, Linux, Unix. Otros Unix, AIX, HP Unix; se integra fácilmente con herramientas de desarrollo de aplicaciones Informix y de terceros en conformidad con los estándares.

²⁵http://www-01.ibm.com/common/ssi/rep_ca/2/838/ESESZP10-0472/ESESZP10-0472.PDF

- Cumple los estándares de coherencia de datos e incluye funciones del patrón arquitectónico cliente-servidor.

2.5.4. Requisitos de usabilidad.

- El diseño de la aplicación de escritorio al ser desarrollada dentro de Informix Dynamic Server versión 10.0 ofrece un mayor rendimiento, disponibilidad y capacidad de ampliación para manejar volúmenes masivos de datos complejos a muy alta velocidad.
- Cuenta con una interfaz amigable e intuitiva para el usuario.

2.5.5. Requisitos de software.

Para obtener un óptimo desempeño es necesario contar con un software actualizado y con las respectivas licencias de uso.

Cliente:

- Sistema operativo Windows, Unix, Linux, AIX, HP_unix.
- Software de emulación NetTerm

Servidor:

- Informix Dynamic Server Versión 11.6
- Informix SQL
- Informix Standard Engine

2.5.6. Requisitos de hardware.

Para logra un óptimo funcionamiento del aplicativo debemos contar con el siguiente tipo de hardware:

Cliente:

- Equipo de escritorio o laptop
- Memoria mínima de 512MB de RAM
- Disco de almacenamiento mínimo de 500 Megabytes

- Procesador Pentium o superior.
- Puertos USB
- WLAN : 802.11b/g/n
- Monitor 15" o superior.

Servidor:

- Memoria RAM de 4GB o superior
- Procesador 1 Intel 3.1 GHz Xeon E3-1220V2
- Discos 2x 1TB 7.2K RPM SATA 3Gbps 3.
- RAID 0 , RAID 1
- Gigabit Ethernet

2.6. Otros requisitos.

Las licencias de uso del motor de la base de datos deben ser renovadas anualmente.

Anexo F: Documento de especificación de casos de uso

Desarrollo de una solución software empresarial para control de vacaciones anuales en la Empresa Importadora Comercial El Hierro Cía. Ltda.

Documento de Especificación de Casos de Uso

Versión 1.1

Historial de Revisiones

Fecha	Versión	Descripción	Autor
15-ABRIL-2015	1.0	Versión Inicial planteada como propuesta de implementación.	Antonio Lloret Orellana
15-JUNIO-2015	1.1	Correcciones realizadas al documento	Antonio Lloret Orellana
6-AGOSTO-2015	1.2	Correcciones realizadas al documento	Antonio Lloret Orellana
21-SEPT-2015	1.3	Correcciones realizadas al documento	Antonio Lloret Orellana

1.1. Diagramas de casos de uso.

El diagrama de casos de uso pretende ayudar a expresar el uso de la aplicación de escritorio que ejecutara el asistente o usuario que registre la información relacionada al Control de Vacaciones, en la siguiente Tabla 33 de manera global se describe los casos de uso que interviene el usuario.

Tabla 33. Descripción de los casos de uso que realiza el usuario.

CODIGO CU.	ACTIVIDAD	ACTORES	DESCRIPCION DE CASO DE USO
CU01	Consultar Cronograma Anual de Vacaciones	Empleados de Imp.Com. El Hierro	Los usuarios deberán abrir la hoja de Excel con el cronograma de vacaciones.
CU02	Control de Acceso	Empleados de Imp.Com. El Hierro	Los usuarios realizan su registro de ingreso al sistema digitando su login y clave para poder operar el aplicativo.
CU03	Registro de Datos	Empleados de Imp.Com. El Hierro	Los usuarios admitidos por el sistema, pueden procesar la información relacionada al Control de Vacaciones.
CU04	Anulación de registro	Empleados de Imp.Com. El Hierro	El administrador del sistema, puede proceder a anular un registro de vacaciones.

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Seguidamente en la Figura 25, se detalla el diagrama general de casos de uso que realiza el asistente de talento humano.

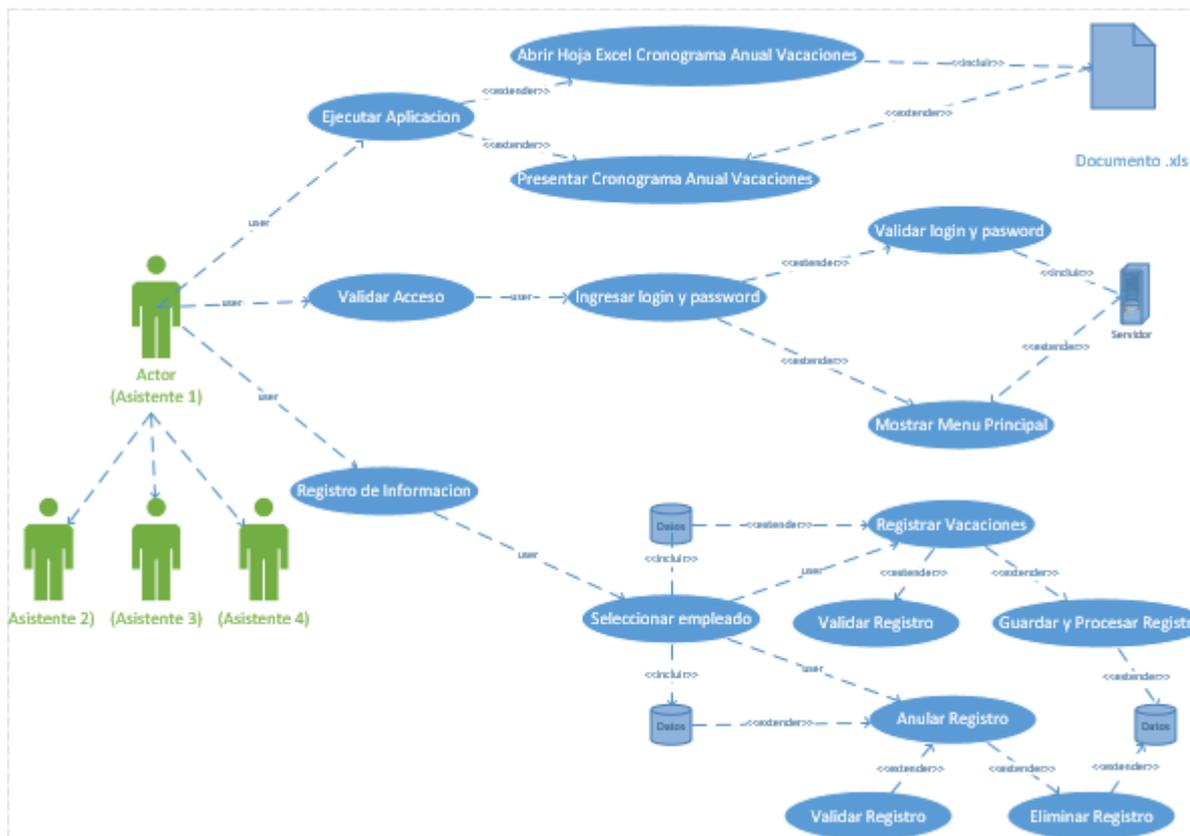


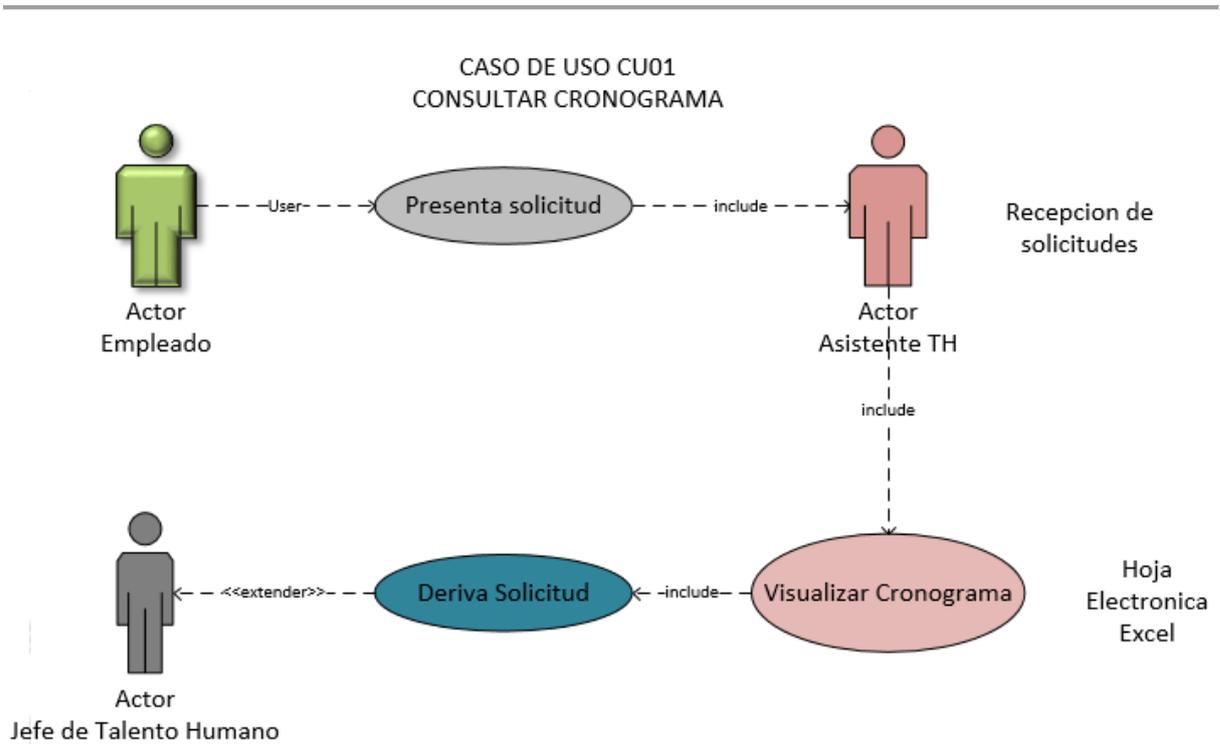
Figura 25. Diagrama general de caso de uso Control de Vacaciones.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

1.2. Especificación de casos de uso.

1.2.1. Consultar Cronograma Anual de Vacaciones.

Tabla 34. Descripción del caso de uso CU01.

Numero:	CU01										
Nombre:	Consultar cronograma Anual de Vacaciones										
Actores:	Asistentes del Departamento de Talento Humano										
Descripción:	El caso de uso descrito permitirá al Asistente de Talento Humano consultar el Cronograma anual de vacaciones.										
Precondiciones:	Abrir la hoja electrónica Excel										
Postcondiciones:	El asistente analizara la Hoja de cálculo										
Flujo Normal	<table border="0"> <tr> <td style="vertical-align: top;">Actor:</td> <td style="vertical-align: top;">Sistema:</td> </tr> <tr> <td>1. Abrir la hoja electrónica Excel</td> <td></td> </tr> <tr> <td>2. Consultar la información</td> <td></td> </tr> <tr> <td></td> <td>3. Configurar y mostrar el Cronograma</td> </tr> <tr> <td></td> <td>4. Fin del caso de uso</td> </tr> </table>	Actor:	Sistema:	1. Abrir la hoja electrónica Excel		2. Consultar la información			3. Configurar y mostrar el Cronograma		4. Fin del caso de uso
Actor:	Sistema:										
1. Abrir la hoja electrónica Excel											
2. Consultar la información											
	3. Configurar y mostrar el Cronograma										
	4. Fin del caso de uso										
Prioridad:	Media										
Referencias:	RF01										



Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

1.3. Control de acceso.

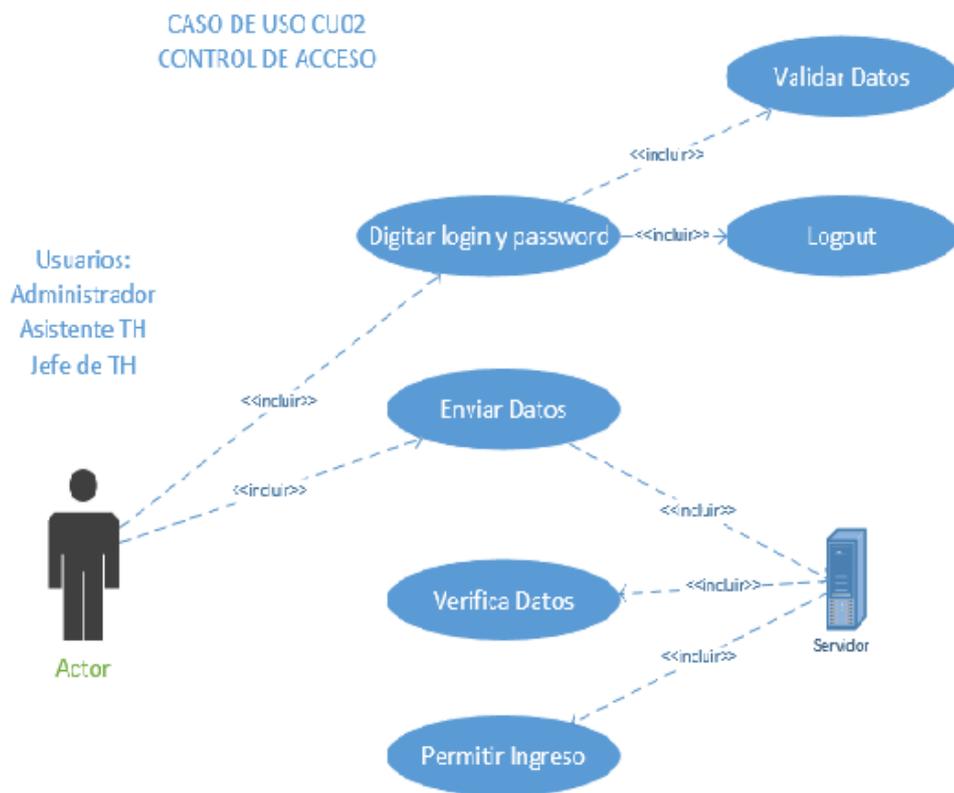
Tabla 35. Descripción del caso de uso CU02.

Numero:	CU02						
Nombre:	Control de acceso.						
Actores:	Usuarios autorizados (Personal registrados previamente por el Departamento de Talento Humano).						
Descripción:	El caso de uso descrito permitirá que solamente el personal autorizado por Talento Humano y Sistemas pueda utilizar el módulo de Control de Vacaciones.						
Precondiciones:	El usuario previamente debe estar registrado en la base de datos de Informix.						
Postcondiciones:	El aplicativo validara el acceso y seguidamente cargara las funcionalidades del módulo.						
Flujo Normal	<table border="0" style="width: 100%;"> <tr> <td style="vertical-align: top; width: 50%;">Actor:</td> <td style="vertical-align: top; width: 50%;">Sistema:</td> </tr> <tr> <td>1. Ejecutar el aplicativo.</td> <td>2. La aplicación presenta la interfaz para el ingreso del identificador login y password.</td> </tr> <tr> <td>3. El usuario ingresa su login y password.</td> <td>4. Si el login y el password son correctos, el aplicativo permitirá el acceso a las funcionalidades del módulo; de lo contrario visualizara un mensaje de error</td> </tr> </table>	Actor:	Sistema:	1. Ejecutar el aplicativo.	2. La aplicación presenta la interfaz para el ingreso del identificador login y password.	3. El usuario ingresa su login y password.	4. Si el login y el password son correctos, el aplicativo permitirá el acceso a las funcionalidades del módulo; de lo contrario visualizara un mensaje de error
Actor:	Sistema:						
1. Ejecutar el aplicativo.	2. La aplicación presenta la interfaz para el ingreso del identificador login y password.						
3. El usuario ingresa su login y password.	4. Si el login y el password son correctos, el aplicativo permitirá el acceso a las funcionalidades del módulo; de lo contrario visualizara un mensaje de error						

que indica que los datos son incorrectos.

5. Fin del caso de uso.

Flujo Alternativo: Si el login y el password son incorrectos; el usuario no podrá acceder al aplicativo
Prioridad: Media
Referencias: RF01 RF02
 CU01 CU02



Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

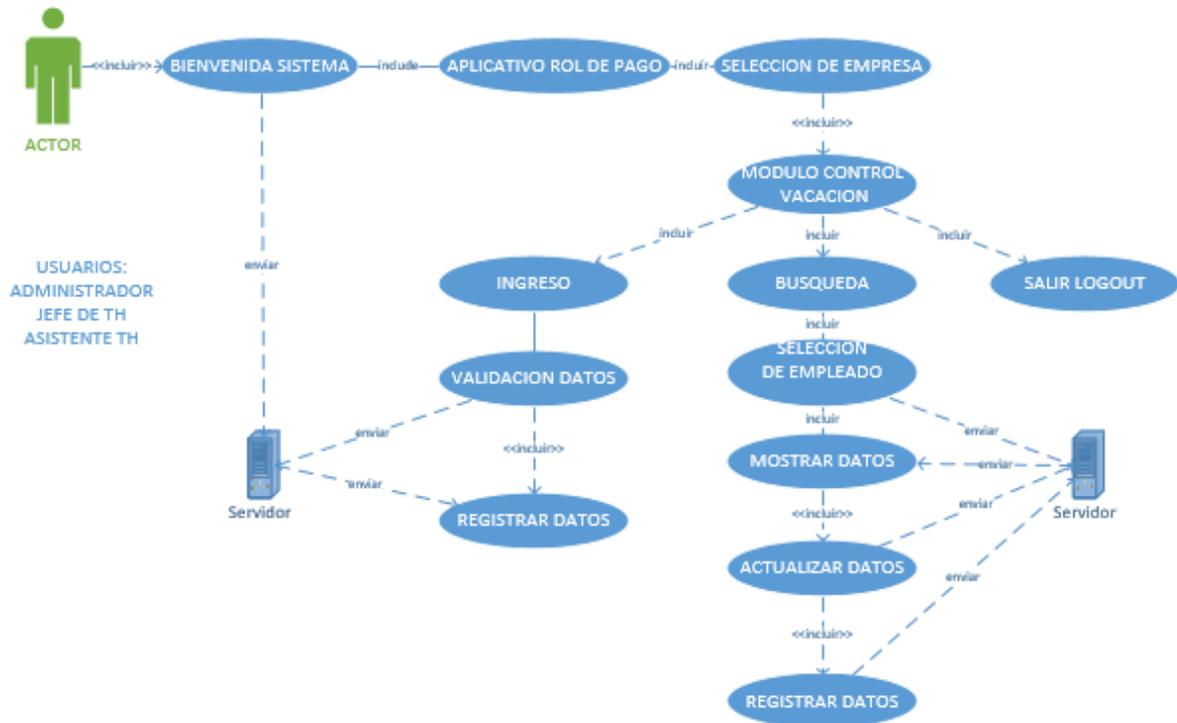
1.4. Registro de datos.

Tabla 36. Descripción del caso de uso CU03.

Numero:	CU03
Nombre:	Registro de datos.
Actores:	Usuarios autorizados (Personal registrados previamente por el Departamento de Talento Humano).

Descripción:	El caso de uso descrito permitirá seleccionar un empleado de la empresa que tenga aprobada su solicitud de vacaciones para proceder a ingresar sus datos mediante el aplicativo en la base de datos.	
Precondiciones:	El usuario previamente debe estar registrado en la base de datos de Informix.	
Postcondiciones:	El empleado seleccionado debe estar habilitado. La aplicación mostrara la interfaz para el registro de datos donde se encuentran todos los empleados de la empresa y permite realizar su validación e ingreso.	
Flujo Normal	Actor:	Sistema:
	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de ingreso. 3. El usuario hace la búsqueda del empleado a ser registrado. 5. El usuario procede a registrar la información pertinente. 7. El aplicativo procede a validar y realizar los cálculos respectivos de asignación de vacaciones. 	<ol style="list-style-type: none"> 2. La aplicación presenta la interfaz para el registro de datos. 4. La aplicación presenta el formulario para el llenado de datos. 6. El usuario procede a guardar la información registrada. 8. Fin del caso de uso.
Flujo Alternativo:	El usuario deberá seleccionar un empleado de la empresa, caso contrario no podrá registrar ninguna solicitud aprobada. La aplicación realizara las validaciones y cálculos respectivos basados en función de las referencias y casos de uso descritos	
Prioridad:	Alta	
Referencias:	RF01 RF02 RF03 CU01 CU02 CU03	

CASO DE USO CU03
REGISTRO DE DATOS



Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

1.5. Anulación de registro de datos.

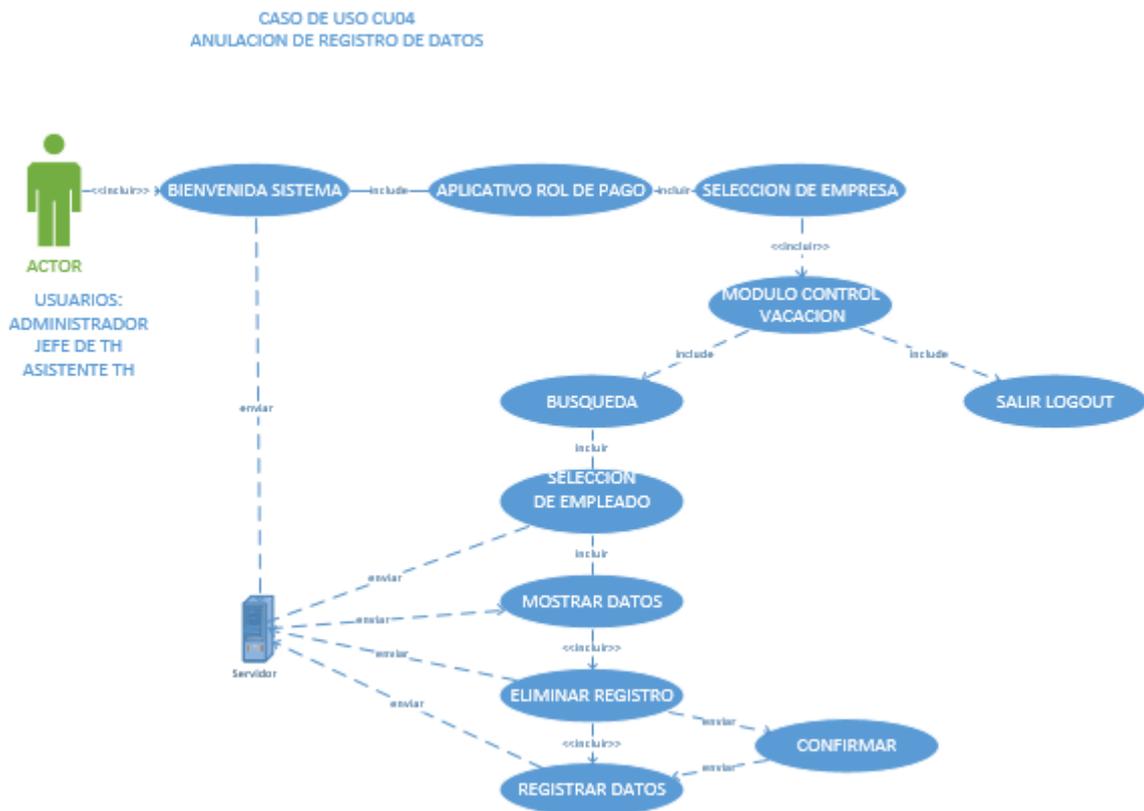
Tabla 37. Descripción del caso de uso CU04.

Numero:	CU04
Nombre:	Anulación de registro de datos.
Actores:	Usuarios autorizados (Personal registrados previamente por el Departamento de Talento Humano y/o Administrador Sistema).
Descripción:	El caso de uso descrito permitirá seleccionar un empleado de la empresa que tenga aprobada su solicitud de vacaciones para proceder a anular sus datos mediante el aplicativo en la base de datos.
Precondiciones:	El usuario previamente debe estar registrado en la base de datos de Informix.
Postcondiciones:	El empleado seleccionado debe estar habilitado.
Flujo Normal	<p>Actor:</p> <ol style="list-style-type: none"> 1. El usuario selecciona la opción de anulación. <p>Sistema:</p>

2. La aplicación presenta la interfaz para el registro de datos.
3. El usuario hace la búsqueda del empleado a ser anulado.
4. La aplicación presenta el formulario para el llenado de datos de validación.
5. El usuario procede a registrar la información pertinente.
6. El usuario procede a confirmar la anulación registrada.
7. El aplicativo procede a validar y realizar los cálculos respectivos de actualización de vacaciones.
8. Fin del caso de uso.

Flujo Alternativo: El usuario deberá seleccionar un empleado de la empresa, caso contrario no podrá registrar ninguna anulación. La aplicación realizara la actualización respectiva basada en función de las referencias y casos de uso descritos

Prioridad: Alta
Referencias: RF01 RF02 RF03 RF04
 CU01 CU02 CU03 CU04



Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio

Anexo G: Documento de arquitectura de software.

**Desarrollo de una solución software empresarial para control de vacaciones anuales
en la Empresa Importadora Comercial El Hierro Cía. Ltda.**

Documento de Arquitectura de Software

Versión 1.1

Historial de Revisiones

Fecha	Versión	Descripción	Autor
15-ABRIL-2015	1.0	Versión Inicial planteada como propuesta de implementación.	Antonio Lloret Orellana
15-JUNIO-2015	1.1	Correcciones realizadas al documento	Antonio Lloret Orellana
6-AGOSTO-2015	1.2	Correcciones realizadas al documento	Antonio Lloret Orellana
21-SEPT-2015	1.3	Correcciones realizadas al documento	Antonio Lloret Orellana

1. Introducción a la Arquitectura de Software.

El software es considerado un elemento fundamental en la evolución de sistemas y productos basados en computadoras. La arquitectura de software incluye procesos, métodos y herramientas que permiten elaborar a tiempo y con calidad sistemas complejos basados en computadoras. El proceso de software incorpora actividades estructurales como son la comunicación, planeación, modelado, construcción y despliegue que son aplicables a todos los proyectos de software. En este documento se pretende describir una visión arquitectónica que mediante herramientas propias de desarrollo de software permitan representar vistas arquitectónicas que caracterizan un aspecto propio de la solución software considerado.

1.1. Propósito.

El objetivo del documento sobre arquitectura de software es proporcionar una descripción de la arquitectura de la solución software implementada, basada en un proceso de investigación sobre diferentes vistas arquitectónicas que permitan aplicar en la solución software planteado.

1.2. Alcance.

Este documento refleja la arquitectura en la cual está sustentada la solución software, configurando de manera detallada los casos de uso y actores que intervienen en las diferentes capas representadas, así como una descripción del patrón arquitectónico Cliente-Servidor seleccionado.

1.3. Definiciones, acrónimos y abreviaturas.

A continuación se presenta una lista de términos utilizados en este documento que muestran una explicación más detallada sobre la temática y que permitan una mejor comprensión e interpretación de los mismos.

Tabla 38. Definiciones, acrónimos y abreviaturas.

Actor	Persona o entidad externa que interactúa con el sistema.
UML	Herramienta que permite realizar el modelado de los diagramas expuestos en este documento.
Caso de uso	Secuencia de eventos que el sistema realiza, lo cual permite obtener un resultado de valor óptico visual.

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

2. Representación de la arquitectura.

La arquitectura seleccionada es el patrón arquitectónico Cliente–Servidor la cual está representada por los siguientes componentes:

- **Cliente:** Este representa la interfaz gráfica, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso, y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario.
- **Servidor:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Aquí se establecen todas las reglas que deben cumplirse; se reciben las solicitudes y se muestran los resultados, residen los programas de aplicaciones que solicitan al gestor de la base de datos para almacenar o recuperar datos.

La arquitectura se sustentara en el modelo denominado 4+1 de kruchten²⁶; Si bien es aclaratorio especificar que no todas las vistas son necesarias de acuerdo al tipo de proyecto que se construye, las vistas que propone kruchten son cuatro + los escenarios. Pero lo más importante lo podemos ver en la trazabilidad, cada una de las vista debe ser trazable a un escenario.

- **Escenarios:** El escenario es representado por diagramas de casos de uso. Desde el escenario se debe poder hacer una trazabilidad a los distintos artefactos de la arquitectura, tanto a una clase, un componente, un proceso, etc. que cubra algunas de las funcionalidades.
- **Vista Lógica:** Debe soporta el análisis y la especificación de los requisitos funcionales. los diagramas uml que satisfacen esta vista son diagrama de clases, de estado y colaboración.
- **Vista de Procesos:** Esta vista debe soportar los requisitos no funcionales. Especificar el ciclo de control de cada operación así como diseñar la concurrencia y sincronización de los procesos de cada una de las clases de la vista lógica. Por lo tanto la vista de procesos debe ser trazable con la vista lógica. Los diagramas que dan soporte a esta

²⁶ **Philippe Krutchen** (nacido en 1952) es un canadiense Ingeniero de software y profesor de ingeniería de Software en la Universidad de British Columbia en Vancouver, Canadá, conocido como Director del proceso de desarrollo (RUP) en Rational Software y Desarrollador de la 4 + 1 arquitectura modelo vista.

vista son de actividad, de estados, de secuencia. El estilo arquitectónico usado cliente/servidor.

- **Vista de Desarrollo:** La vista de desarrollo muestra de forma estática la organización de los módulos del software en entorno de desarrollo. El software es empaquetado en pequeños trozos (librerías de programa, subsistemas, componentes, etc.), los subsistemas se organizan en capas jerárquicas, y cada capa proporciona una interfaz bien definida a sus capas superiores. Esta vista se apoya en los diagramas uml de componentes y paquetes. Esta vista es trazable con la vista lógica y el estilo arquitectónico más usado es el de n-tier.
- **Vista Física:** La vista física contiene los nodos que forman la topología hardware sobre la que se ejecuta el sistema. Se preocupa principalmente de la distribución, entrega e instalación de las partes que constituyen el sistema. Los aspectos estáticos de esta vista se representan mediante los diagramas de despliegue y los aspectos dinámicos con diagramas de interacción, estados y actividades.

3. Metas y restricciones de la arquitectura.

La arquitectura Cliente–Servidor es un modelo que tiene como meta proveer usabilidad, flexibilidad, interoperabilidad y escalabilidad en las comunicaciones; dichos atributos de calidad buscan validar que el uso del patrón arquitectónico Cliente–Servidor sea la selección más adecuada para la solución software a desarrollar.

3.1. Metas.

- Implementar una arquitectura distribuida que permita a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma.
- Desarrollar e implementar instrucciones SQL desde el servidor de base de datos Informix.
- Proveer los procesos de las reglas del negocio.
- Proveer las estructuras fundamentales de las Bases de Datos.
- Proveer los elementos de la interfaz.
- Proveer la infraestructura general de los sistemas, sobre la cual los equipos de trabajo puedan montar sus aplicaciones.
- Configurar técnicamente las seguridades del Servidor.

3.2. Restricciones.

- Los usuarios que podrán ingresar al aplicativo son los que designe el Departamento de Talento Humano previa autorización de Gerencia.
- La solución software operara con Informix 11.6 como motor de la base de datos.
- Los únicos componentes que se utilizaran son los descritos dentro del patrón arquitectónico Cliente-Servidor.

3.3. Vista lógica.

3.3.1. Diagrama de clases.

El diagrama de clases permitirá observar las relaciones existentes entre las clases que configuran la solución software, las mismas que se asocian mediante sus atributos o métodos.

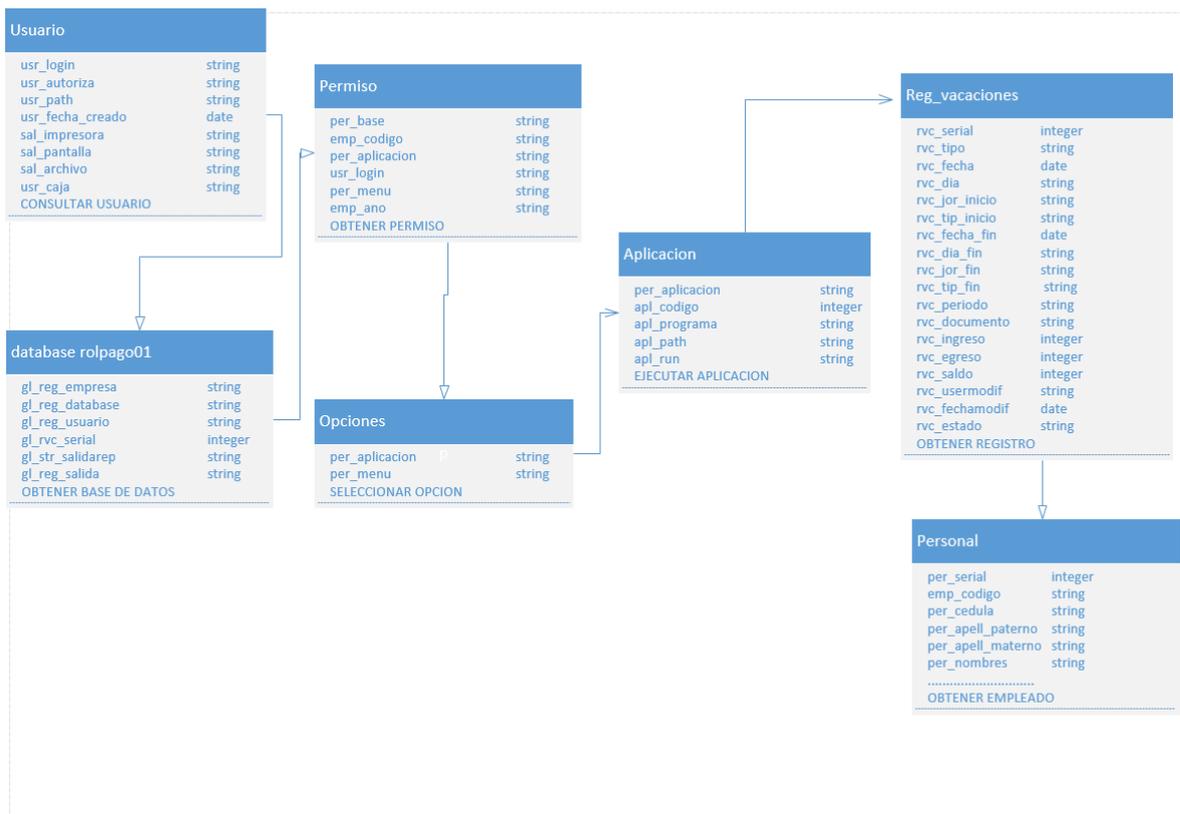


Figura 26 Diagrama de clases
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

A continuación se detalla cada una de las Tablas que configuran el diagrama de clases:

- **USUARIOS:** Esta Tabla almacena todos los usuarios autorizados para ejecutar la aplicación y que se encuentran registrados en la base de datos Informix.
- **ROLPAGO01:** Esta Tabla contiene la información de la base de datos vinculada a la empresa seleccionada.
- **PERMISO:** Esta Tabla registra si el usuario autorizado posee permisos de ejecución de la aplicación.
- **OPCIONES:** Esta Tabla contiene los menús disponibles por el usuario en función de la aplicación a ejecutar.
- **APLICACIÓN:** Esta Tabla contiene los códigos de aplicación vinculada con la ruta y ubicación de los archivos en donde se guarda la información de la base de datos.
- **REG_VACACION:** Esta Tabla contiene toda la información relativa al control de vacaciones de los empleados de la empresa.
- **PERSONAL:** Esta Tabla almacena toda la información relativa al empleado que haya laborado en la Empresa.

3.3.2. Modelo Entidad–Relación.

El modelo E-R permite detallar el modelado de datos de la solución software, representado por medio de objetos en las que se destaca los atributos y propiedades que se encuentran relacionados entre ellos.

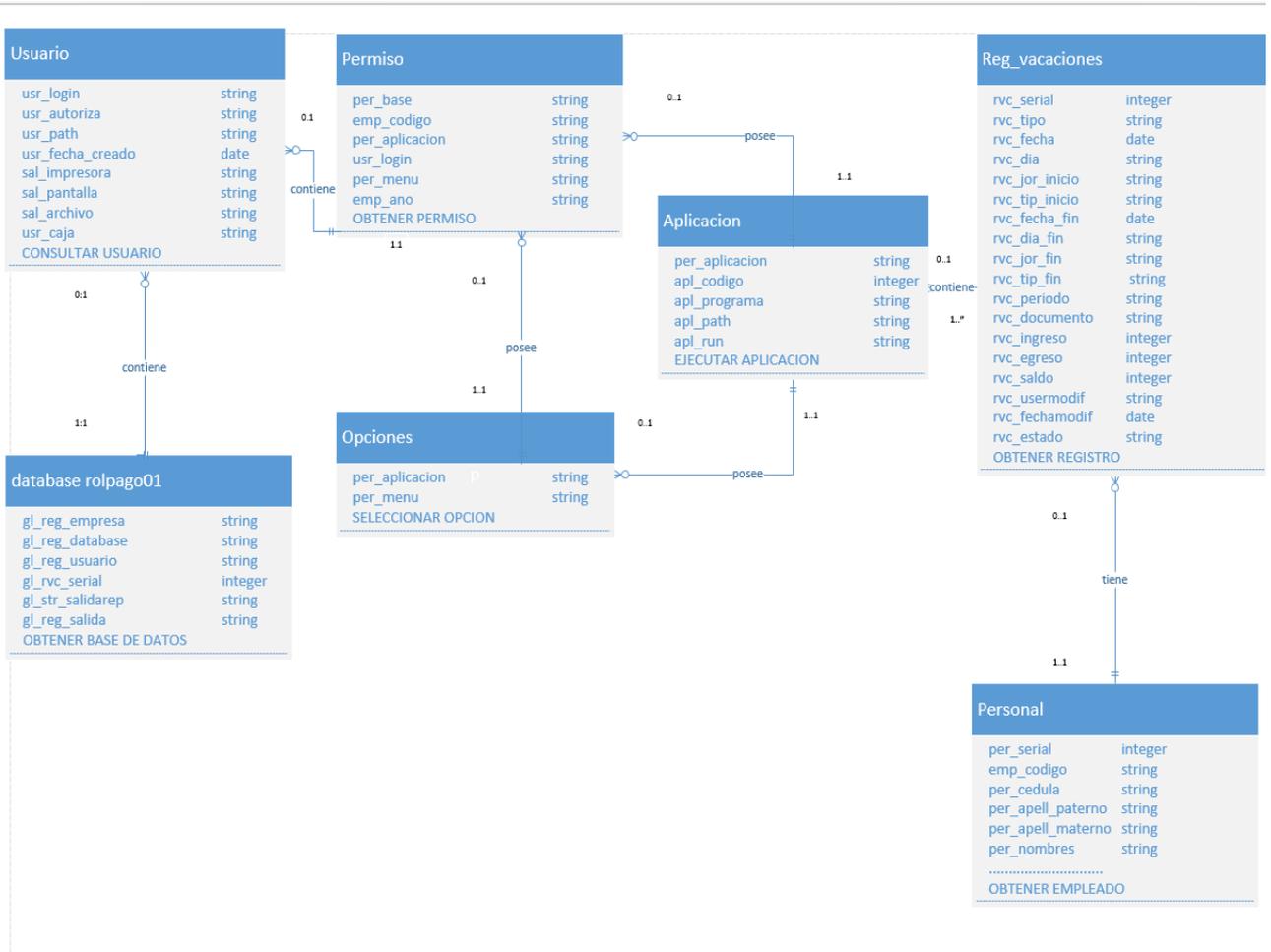


Figura 27. Modelo Entidad-Relación de la solución software.
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

3.4. Vista de Implementación.

3.4.1. Diagrama de Secuencia.

a) Ejecutar la Aplicación.

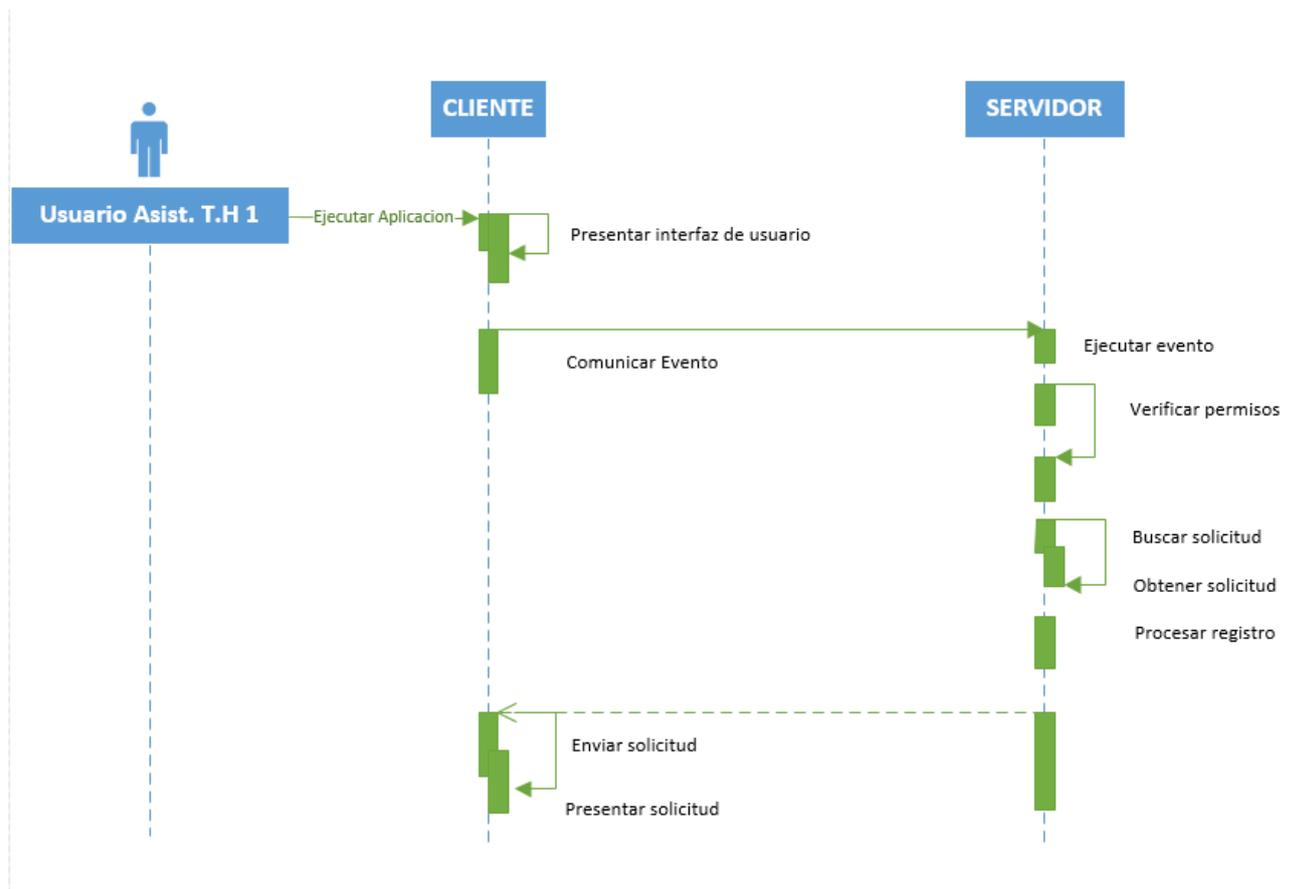


Figura 28. Diagrama de secuencia de la ejecución de la solución software.
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

b) Consultar Usuario

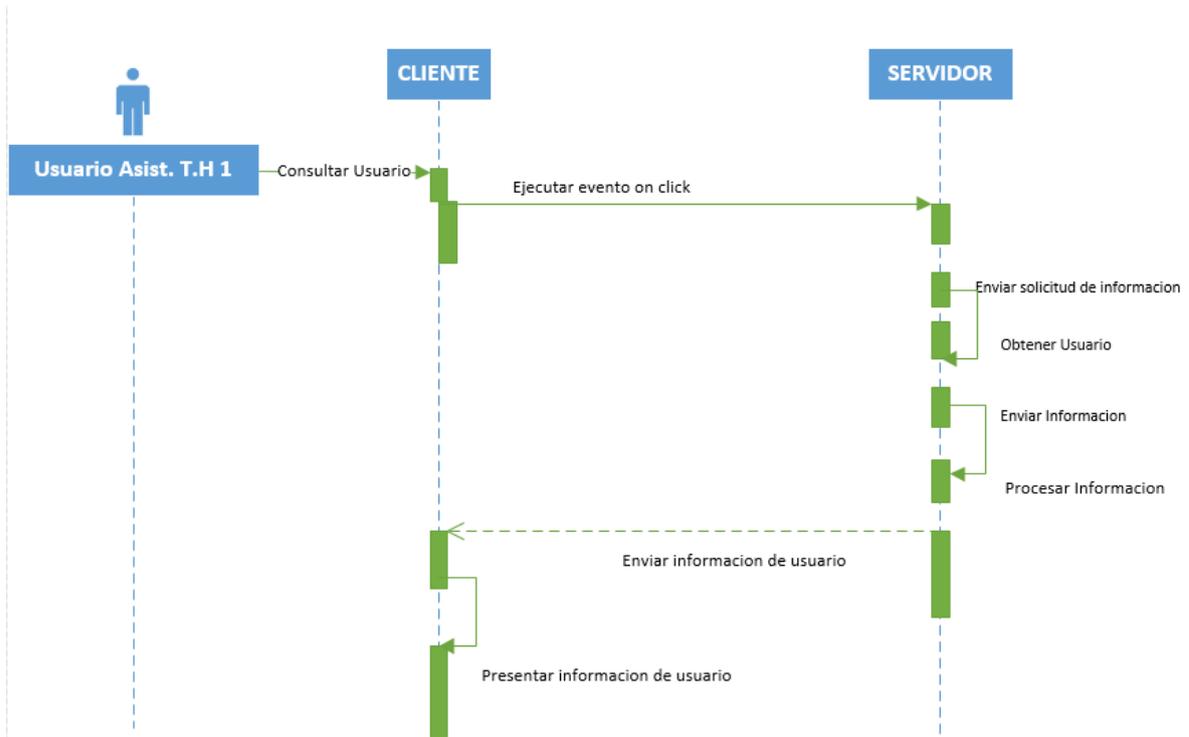


Figura 29. Diagrama de secuencia de consulta de usuarios.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

c) Ingresar claves de acceso

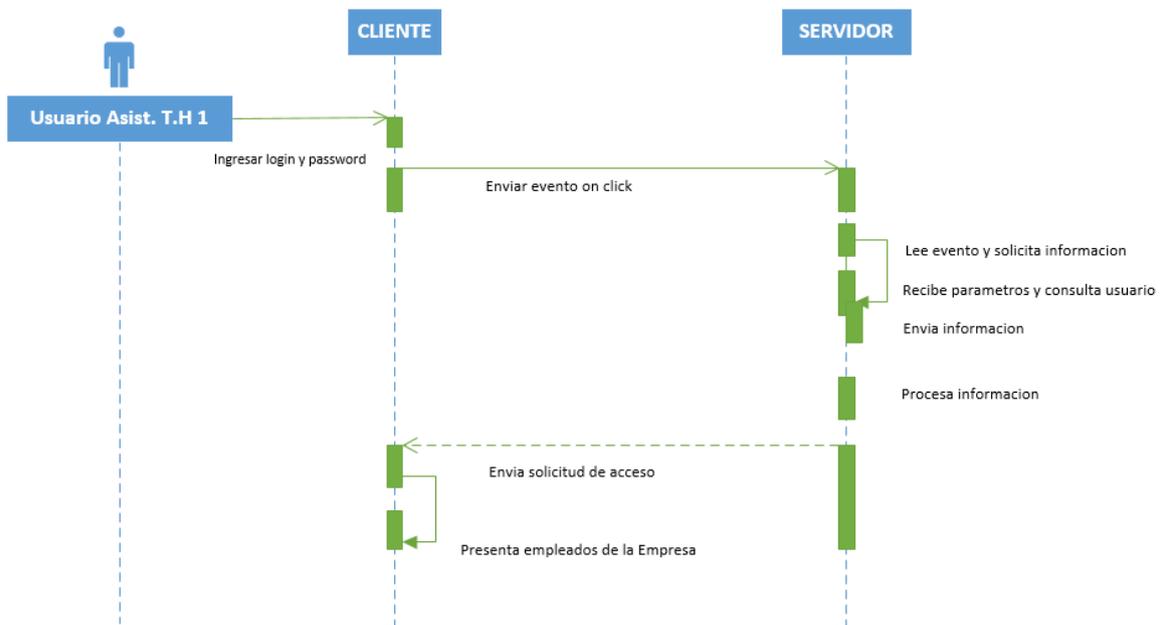


Figura 30. Diagrama de secuencia de validación de claves de acceso.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

d) Registro de solicitudes y anulaciones

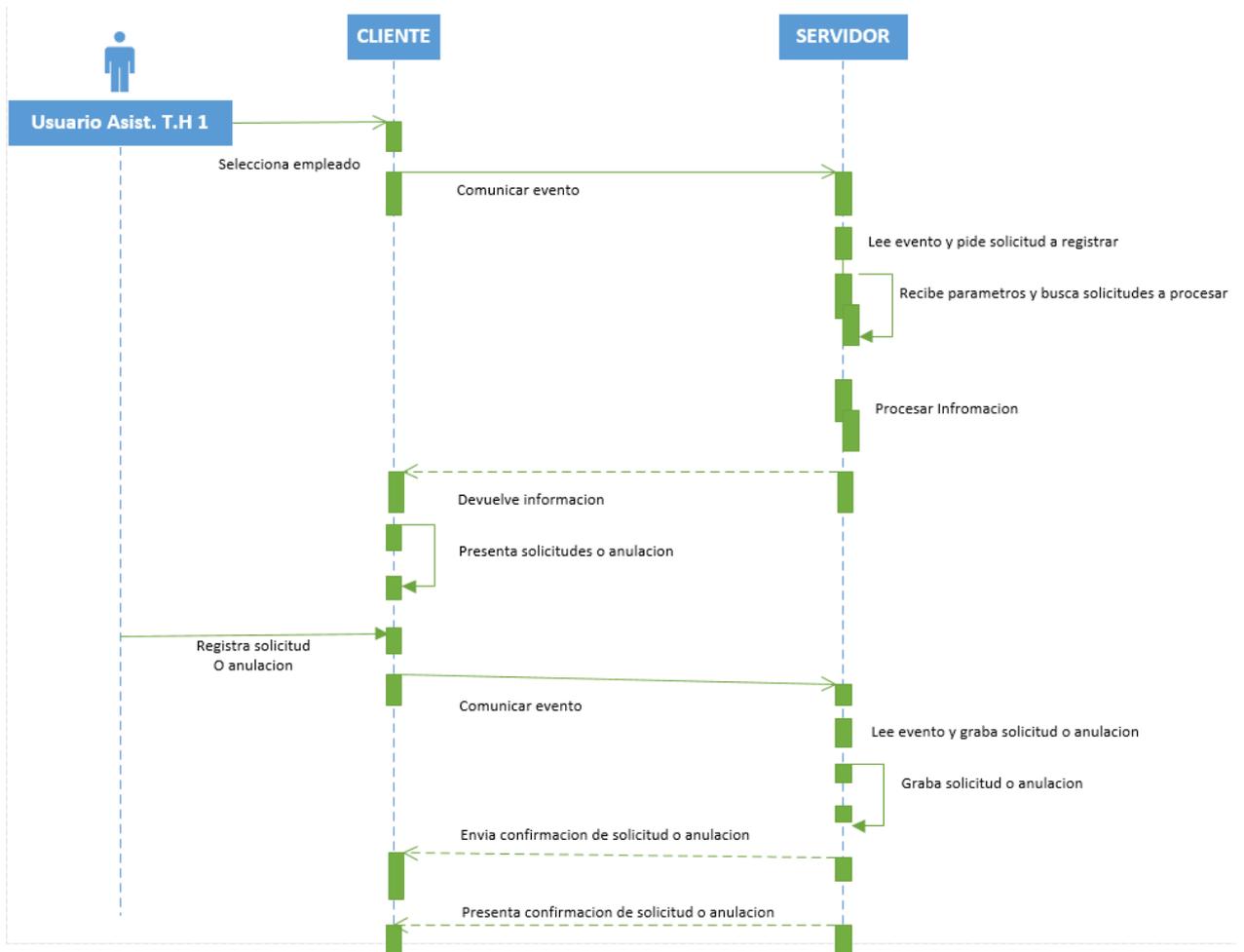


Figura 31. Diagrama de secuencia de solicitudes o anulaciones.
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

3.4.2. Diagrama de estados.

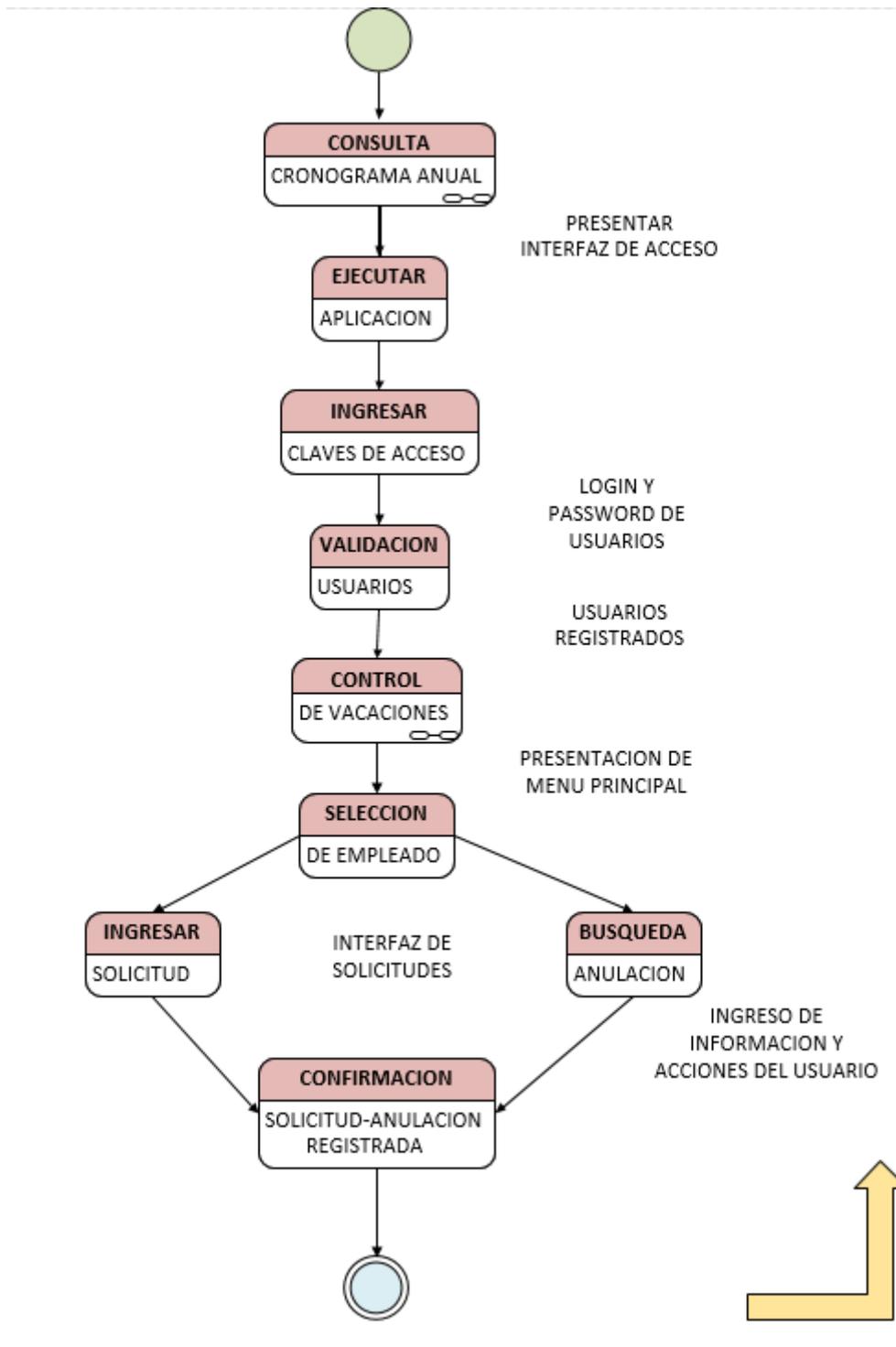


Figura 32. Diagrama de Estados de la solución software
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

3.4.3. Diagrama de colaboración.

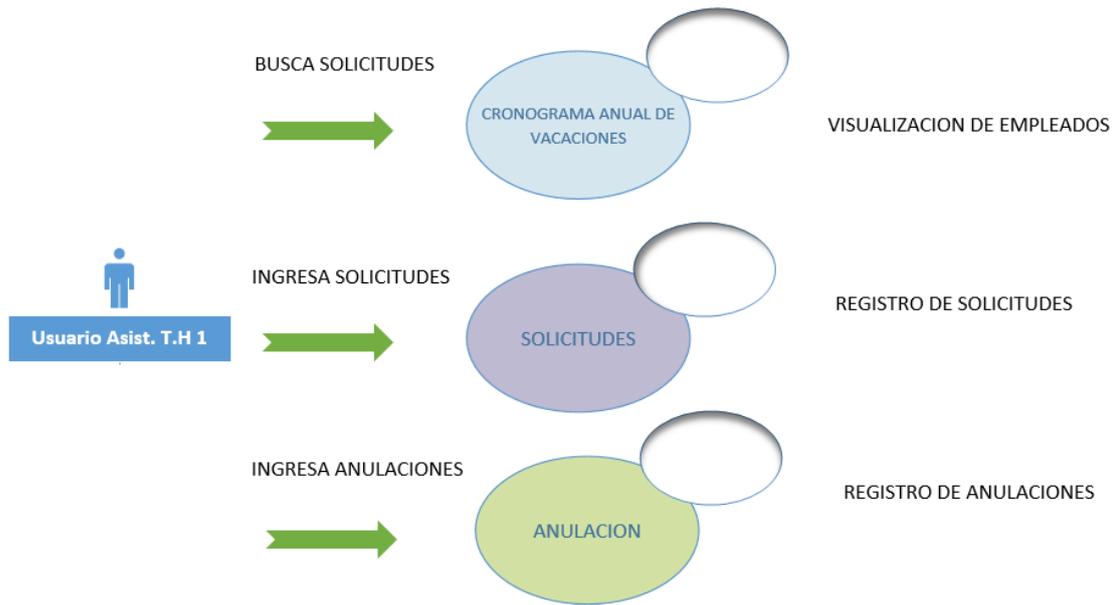


Figura 33. Diagrama de Colaboración de la solución software.
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

3.5. Vista de procesos.

3.5.1. Diagrama de actividad de la solución software.



Figura 34. Diagrama de actividad de la solución software.

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

3.5. Vista de despliegue.

3.6.1 Diagrama de despliegue.

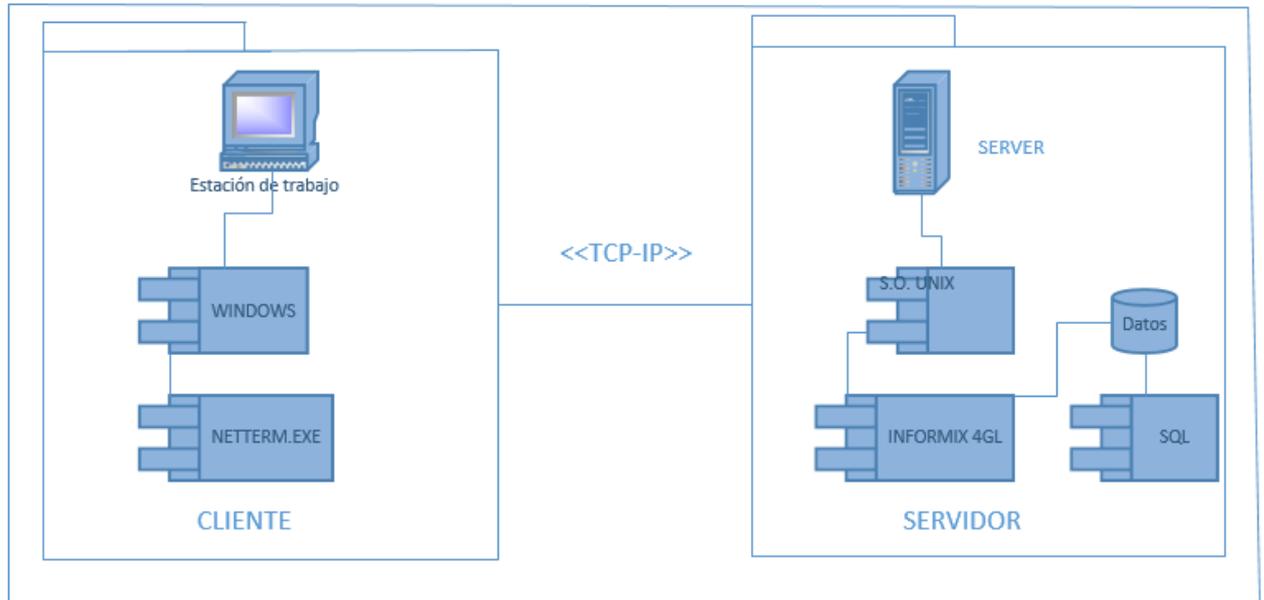


Figura 35. Diagrama de despliegue de la solución software.
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Anexo H: Documento de plan de pruebas

**Desarrollo de una solución software empresarial para control de vacaciones anuales
en la Empresa Importadora Comercial El Hierro Cía. Ltda.**

Documento de Plan de Pruebas

Versión 1.1

Historial de Revisiones

Fecha	Versión	Descripción	Autor
15-ABRIL-2015	1.0	Versión Inicial planteada como propuesta de implementación.	Antonio Lloret Orellana
15-JUNIO-2015	1.1	Correcciones realizadas al documento	Antonio Lloret Orellana
6-AGOSTO-2015	1.2	Correcciones realizadas al documento	Antonio Lloret Orellana
21-SEPT-2015	1.3	Correcciones realizadas al documento	Antonio Lloret Orellana

1. Introducción al plan de pruebas.

Este documento está basado en la aplicación de la metodología de RUP Rational Unified Process; que tiene como propósito sujetarse a las técnicas, herramientas y actividades que provee dicha metodología, de tal manera que al realizar el plan de pruebas se logre comprobar que la solución software cumple con los estándares de calidad y suficiencia.

1.1. Propósito.

El desarrollo del plan de pruebas pretende generar un proceso de validación y comprobación de los requerimientos funcionales de la solución software, de tal manera que permitan verificar si concuerdan con los resultados esperados en la definición del proyecto. Con la realización del plan de pruebas se podrá detectar errores, defectos, y fallas que tenga la aplicación, esta información será de mucha utilidad para poder realizar las mejoras y correcciones que sean necesarias, con el fin de obtener un producto de alta calidad, en función de cumplir con los requerimientos del cliente Imp. Com. El Hierro.

1.2. Alcance.

En el capítulo 4 Pruebas de Validación se detalla con precisión cada uno de los procesos y tipos de prueba que se planificaron para la demostración del funcionamiento de la solución software; es por ello que en este anexo nos centraremos en puntos específicos de la solución software para efectos de validación y comprobación de su correcto funcionamiento.

Para efectos de desarrollo de software, los tipos de prueba se basaran en la metodología RUP, teniendo en cuenta que esta metodología está orientada a la documentación y formalización de todas las actividades realizadas. Cabe resaltar que las pruebas se pueden realizar en cualquier momento del desarrollo de la solución software; a continuación se presenta un resumen de los tipos de prueba:

Tabla 39. Tipos de prueba aplicadas a la solución software.

TIPO DE PRUEBA	DEFINICIÓN	FASE DE RUP
Unitarias	Consisten en la ejecución de actividades que le permitan verificar al desarrollador que los componentes unitarios están codificados bajo condiciones de robustez.	Elaboración
Integración	Consisten en la comprobación de que elementos del software que interactúan entre sí, funcionan de manera correcta.	
Sistema	Consiste en la ejecución de actividades de prueba en donde se debe verificar que la	Construcción

	funcionalidad total de un sistema fue implementada de acuerdo a los documentos de especificación definidos en el proyecto; deben cubrir los aspectos funcionales y no funcionales del sistema, debe utilizar como bases de prueba entregables tales como: requerimientos iniciales, casos de uso, historias de usuario, diseños, manuales técnicos y de usuario final, etc. Por último, es importante que los tipos de pruebas ejecutados en este nivel se desplieguen en un ambiente de pruebas.	
Funcionales: Usabilidad Aceptación	Usabilidad permite encontrar problemas relacionados con el factor humano o denominado de usabilidad. La prueba de aceptación es considerada la prueba final basada en las especificaciones de los usuarios	Transición

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

En RUP, las pruebas son enfocadas a través del uso de un proceso iterativo y de herramientas. Un enfoque iterativo para la evaluación permite a la Empresa tratar las pruebas casi de la misma forma que el desarrollo de software es enfocado.

1.3. Descripción general del flujo de trabajo de pruebas.

El propósito de este flujo de trabajo llamado también workflow de RUP es:

- Verificar la interacción entre objetos
- Verificar la interacción apropiada de todos los componentes del software
- Verificar que todos los requerimientos hayan sido implementados correctamente
- Identificar y asegurar que los defectos se hayan atendido y resuelto antes del despliegue del software.

Este enfoque permite a la Empresa:

- Identificar posibles riesgos al inicio de un proyecto.
- Reducir el costo de corregir fallas enfocando los recursos cuando y donde tendrán el mayor impacto.

- Maximizar la efectividad por medio de adaptar el enfoque, el proceso o el presupuesto según va progresando el proyecto.

En la Figura 36 podemos apreciar las etapas que componen el proceso de pruebas

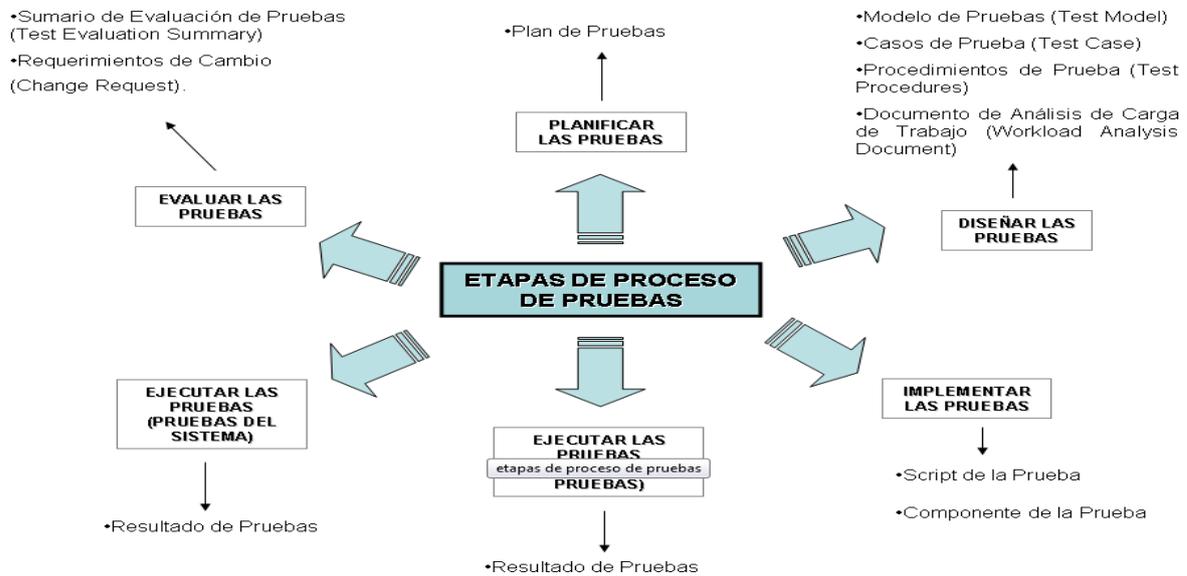


Figura 36. Etapas de proceso de pruebas.

Fuente: (Tomado del libro electrónico de RUP Versión 2003.06.12.01).

Elaborado por: Lloret Antonio.

1.4. Estrategia de pruebas.

Una estrategia de prueba del software debe incluir pruebas de bajo nivel que verifiquen que todos los segmentos de código fuente se han implementado correctamente, así como pruebas de alto nivel que validen las principales funciones del sistema frente a los requisitos del cliente.

1.5. Herramientas para pruebas.

En función del tipo de prueba, se usaran herramientas de software vinculadas con el motor de la base de datos Informix y sistema operativo Red Hat;

Informix incorpora una serie de herramientas con el fin de facilitar y agilizar el trabajo al usuario; así como permitir que el desarrollador pueda realizar pruebas de verificación e integridad de la base de datos y probar que la solución software cumpla con los requerimientos del usuario. En la siguiente Tabla 40 se enumera algunas de las herramientas a utilizar.

Tabla 40. Herramientas para la ejecución de pruebas.

TIPO DE HERRAMIENTA	DESCRIPCIÓN	TIPO DE USO
ONCHECK	Comprueba los discos e índices de la base de datos y los repara.	Reparación de índices en caso de fallos del servidor
ONMONITOR	Visualiza un menú de administración	Administra en forma directa funciones de recuperación y registro de datos
ISQL	Muestra un menú para consultas, formularios y SQL	Ejecución de scripts y procedimientos de consulta y verificación de datos.
DBACCESS	Menú con herramientas de administración de la base de datos.	Creación, Modificación, Eliminación de registros de la base de datos

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

1.6. Configuración del ambiente de pruebas.

El ambiente de pruebas para la solución software se desarrollara aprovechando las instalaciones de infraestructura de software y hardware que dispone la Empresa Imp.Com. El Hierro; y la computadora portátil del desarrollador en la cual se configura una máquina virtual que tiene como propósito simular el entorno de trabajo de la Empresa.

Es importante destacar que en el servidor principal que representa la ubicación física del modelo en el cual se halla configurado el patrón arquitectónico cliente-servidor; que interactúa con la base de datos Informix.

1.6.1. Instalación de VirtualBox.

Lo primero que debemos hacer, es instalar esta herramienta en nuestro sistema. Para ello debemos descargar la versión correspondiente a nuestro sistema operativo. Una vez descargada la instalamos como un programa convencional, teniendo en cuenta que durante el proceso de instalación nos pedirá permiso para instalar unos controladores de red para conectar las máquinas virtuales a Internet. El proceso de instalación es sencillo, rápido y libre de malware.

Una vez instalado el software en nuestro sistema lo ejecutamos y veremos una ventana similar a la siguiente.



Completada la instalación; a continuación elegiremos el sistema operativo que vamos a instalar en ella. Podemos ponerle el nombre que queramos y elegir tanto un fabricante (Microsoft, Informix, Linux, Oracle, etc.) como la versión del sistema operativo (Windows 7, Windows 8, etc.).

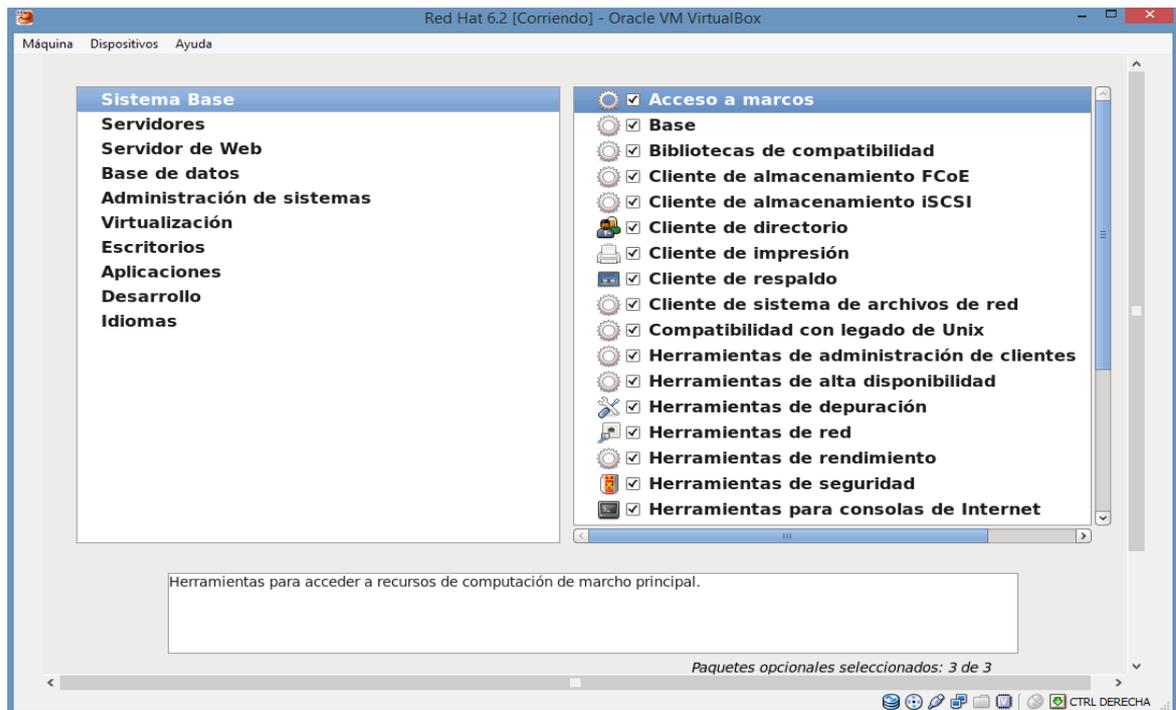
La instalación de la máquina virtual se implementó satisfactoriamente.

1.6.2 Instalación de sistema operativo Linux Red Hat 6.2.

Antes de ejecutar la máquina virtual debemos cargar un disco para instalar el sistema operativo ya que, por defecto, el disco duro estará vacío. Para ello abrimos el menú de configuración y en el apartado "Almacenamiento" podemos elegir el medio desde el que instalaremos el sistema operativo en la máquina de VirtualBox.

Luego de proceder con la re inicialización de la maquina se continúa con la instalación del sistema operativo en la máquina virtual.

Para finalizar instalamos todas las opciones que nos permite la herramienta de Virtualbox con el fin de garantizar que el entorno de simulación funcione óptimamente.



Felicitaciones, ha completado la instalación de Red Hat Enterprise Linux.

Por favor, reinicie su equipo para poder utilizar el nuevo sistema instalado. Tenga en cuenta que ya podrían existir disponibles actualizaciones al software de su sistema que garanticen su correcto funcionamiento, y que luego de reiniciar su equipo, es recomendable instalarlas.

1.6.3. Instalación de la base de datos Informix.

Agregamos los hosts para poder realizar la comunicación con los distintos servidores de la red.

- 1.- Para ello se ejecuta la terminal.
- 2.- Nos ubicamos en la ruta `cd /etc`.
- 3.- Una vez en este directorio crear un archivo con el nombre `hosts`. -> `vi hosts`.
- 3.1.- Presione **a** para editar el documento. Y agregar la dirección IP 192.168.0.222 del servidor presione la tecla **"tab"**, e ingresamos el nombre del servidor en el que está trabajando `appserver`, además se agrega la IP y el nombre del servidor principal.
- 3.2.- En el mismo directorio crear un archivo llamado `hosts.equiv`. Presione **a** para editar y agregue el nombre del servidor principal ej.: `"hpg8principal"`.
- 4.- Para crear el último archivo cambiar de directorio. -> `cd /root`
- 5.- Crear un archivo con el nombre `.rhosts` -> `vi .rhosts`
- 5.1.- Presione **a** para editar el archivo y agregue el nombre del servidor principal.
- 6.- Realizar la copia de archivos para el funcionamiento remoto del servidor
 - `rcp hpg8principal:/etc/securetty . /etc`
 - `rcp hpg8principal:/etc/.hosts.equiv . /etc`
 - `rcp hpg8principal:/root/.rhosts . /root`

Nota: Se debe agregar los datos del nuevo servidor en los archivos de todos los servidores para el acceso remoto.

Seguidamente debemos copiar el archivo `Profile` que es el que contiene datos acerca de la configuración del sistema con la base de datos.

Realizar la copia de este archivo dependiendo de la versión de sistema operativo, copiarlo de un servidor con las mismas características.

- 1.- Realizar copia del archivo en este caso desde el servidor principal.
 - `rcp hpg8principal:/etc/profile .`
- 2.- Modificar el archivo copiado para ello se realiza **"vi /etc/profile"**
- 3.- Cambiar el nombre del servidor por el actual ej. `Informixserver = ol_appserver`
- 4.- Guardar los cambios, cerrar e iniciar sesión para que se actualicen las variables de ambiente.

➤ **Programas necesarios para la instalación.**

- 1.- Crear un usuario con los siguientes datos: usuario: `informix`, contraseña: `informix`
- 2.- Copiar los instaladores necesarios para la base de datos desde el disco a la carpeta `home/informix`:

- SQL development x86_64 bits
- 4gk rds development FC5 x86_64 bits
- Informix Growth edition x86_64 bits

➤ **Instalación de SQL.**

1.- Los archivos copiados a **/home/informix**, se encuentran comprimidos con extensión **.tar**, descomprimirlos con el siguiente comando. **"tar -xvf *SQL*"**

2.- Para instalar **SQL** ejecutar. **"/installsql"**, y seguir las instrucciones.

➤ **Instalación 4GL.**

1.- Para instalar 4gl necesita descomprimir, para ello ejecute el siguiente comando:

"tar -xvf *4GL*", y siga las instrucciones.

2.- Instalar 4gl con el siguiente comando: **"/install4gp"**.

➤ **Instalación de la Base de datos Informix.**

1.- Para instalar la base de datos necesita descomprimir el paquete de instalación para ello ejecutar el siguiente comando:

tar -xvf *Growth*

2.- Para ejecutar el instalador de la base de datos ejecutar el siguiente comando:

"/ids_install".

3.- Elija instalación custom, y siga las instrucciones.

Después de terminar la instalación se necesita, realizar el siguiente procedimiento para el correcto funcionamiento del sistema y de la base de datos.

Informix Tools v7.50 FC5, Informix Dynamic Server v11.70FC5GE

Ambos en el mismo directorio

Adicionalmente en otro directorio pe: **/tmp/tools** (setear temporalmente **INFORMIXDIR=/tmp/tools**) y sobre el instalar las Tools v7.50FC5.

Tras la instalación volver a setear el **INFORMIXDIR** a la ruta original (**/home/informix**) y copiar desde **/tmp/tools/bin/*** hacia **/home/informix/bin/** adicionalmente **/tmp/tools/lib/*** hacia **/home/informix/lib**

➤ Configuración de raw

1.- Para la creación de chunks se necesita modificar el siguiente archivo: **vi /etc/fstab**

1.1.- Eliminar las líneas con las etiquetas **inf01, inf02 e inf03**, presionando “**dd**”.

1.2.- Guardar y cambiar de directorio “**cd /etc/udev/rules.d**”.

2.- Modificar el archivo **60-raw.rules**

2.1.- Agregar las siguientes líneas

2.1.1.- **ACTION=="add", KERNEL=="cciss/c0d1p3", RUN+="/bin/raw /dev/raw/raw1 %N"**

ACTION=="add", KERNEL=="cciss/c0d1p2", RUN+="/bin/raw /dev/raw/raw2 %N"

ACTION=="add", KERNEL=="cciss/c0d1p1", RUN+="/bin/raw /dev/raw/raw3 %N"

Donde “cciss” es el nombre de disco y “c0d1p3” la partición en la que está alojado “inf01” es decir la partición para la base de datos, “c0d1p2” es el nombre de la partición para datos temporales o “inf02”, c0d1p1 es el nombre de la partición para replicas o “inf03”.0

```
# This file and interface are deprecated.
# Applications needing raw device access should open regular
# block devices with 0_DIRECT.
#
# Enter raw device bindings here.
#
# An example would be:
# ACTION=="add", KERNEL=="sda", RUN+="/bin/raw /dev/raw/raw1 %N"
# to bind /dev/raw/raw1 to /dev/sda, or
# ACTION=="add", ENV{MAJOR}=="8", ENV{MINOR}=="1", RUN+="/bin/raw /dev/raw/raw
2 %N %n"
# to bind /dev/raw/raw2 to the device with major 8, minor 1.
ACTION=="add", KERNEL=="cciss/c8d1p3", RUN+="/bin/raw /dev/raw/raw1 %N"
ACTION=="add", KERNEL=="cciss/c8d1p2", RUN+="/bin/raw /dev/raw/raw2 %N"
ACTION=="add", KERNEL=="cciss/c8d1p1", RUN+="/bin/raw /dev/raw/raw3 %N"

```

Guardar los cambios y reiniciar el equipo para que se creen los raw.

```
root@hpgiron rawl# pwd
/dev/raw
root@hpgiron rawl# l
total 0
-rw-r--r-- 1 root    root      0 oct 12 11:11 global.log
-rw-rw---- 1 informix informix 162, 1 oct 12 13:48 raw1
-rw-rw---- 1 informix informix 162, 2 oct 12 10:37 raw2
-rw-rw---- 1 informix informix 162, 3 oct 12 10:37 raw3
-rw-r--r-- 1 root    root      0 oct 12 11:11 root.log
root@hpgiron rawl#
```

Una vez comprobado que existan los raw.

Para el funcionamiento de los raw, se necesita asignar el usuario de informix. Para eso se cambia de directorio “**cd /dev/raw**”.

Asignar el usuario con “**chown informix.informix ***”.

Dar los siguientes permisos “**chmod 660 ***”

➤ Configuración base de datos

Para configurar la base de datos se necesita copiar 3 archivos importantes de otro servidor que tenga las mismas características en el que estamos instalando, para realizar los mínimos cambios posibles.

1.- Cambiar de directorio a “**cd /home/informix/etc**”.

2.- Realizar la copia de archivos necesarios de la siguiente manera:

- rcp hpservidor:/home/informix/etc/sqlhosts .
- rcp hpservidor:/home/informix/etc/onconfig .
- rcp hpservidor:/home/informix/etc/termcap .

3.- Editar archivo “**sqlhosts**”, copiar y pegar las 2 últimas líneas.

Editar el nombre del servidor, dirección IP, y número de servidor que corresponde.

4.- Cambiar de directorio a **cd /etc**

5.- Editar el archivo services, “**vi services**”.

5.1.- Agregar la siguiente línea:

- sqlturbo 1526/tcp #Servidor informix

➤ Crear chunks, dbspace, blobspace.

1.- Crear un directorio en la carpeta /home/informix, con el nombre chunks, para ello se ejecuta “**mkdir chunks**”.

2.- En el directorio creado ejecutar el siguiente comando para generar los chunks.

In -s /dev/raw/raw1 rootdbs.000

In -s /dev/raw/raw2 tempdbs.000

In -s /dev/raw/raw3 sbospace.000

3. – Ejecute “**onmonitor**” en la terminal para crear el dbspace para la base de datos. Al copiar el archivo “**onconfig**” de otro servidor se copia la configuración del “**dbsapce**” por lo que lo único que se debe cambiar es el espacio de root y la ruta del chunk o primary path y “**number of logical logs**” que serán la cantidad de transacciones permitidas.

```
INITIALIZATION: Make desired changes and press ESC to record changes.
Press Interrupt to abort changes. Press F2 or CTRL-F for field editing.
DISK PARAMETERS
Page Size [ 2 ] Kbytes Mirror [N]
Tape Dev. [ /dev/null ]
Block Size [ 32 ] Kbytes Total Tape Size [ 10240 ] Kbytes
Log Tape Dev. [ /dev/null ]
Block Size [ 32 ] Kbytes Total Tape Size [ 10240 ] Kbytes
Stage Blob [ ]
Root Name [ rootdbs ] Root Size [ 50000 ] Kbytes
Primary Path [ /home/informix/data/rootdbs ] Root Offset [ 100 ] Kbytes
Mirror Path [ ] Mirror Offset [ 0 ] Kbytes
Phy. Log Size [ 2000 ] Kbytes Log. Log Size [ 2000 ] Kbytes
Number of Logical Logs [ 6 ]
```

En esta pantalla modifique el nombre de servidor y el tamaño de memoria física compartida para la base de datos.

```

SHARED MEMORY: Make desired changes and press ESC to record changes.
Press Interrupt to abort changes. Press F2 or CTRL-F for field-level help.
          SHARED MEMORY PARAMETERS
Server Number          [ 10 ]      Server Name [onlineuch ]
Server Aliases [
Dbospace Temp [
Deadlock Timeout      [ 60 ] Secs  Number of Page Cleaners [ 1 ]
Forced Residency      [N]          Stack Size (K) [ 32 ]
Non Res. SegSize (K)  [ 8000 ]     Optical Cache Size (K) [ 0 ]

Physical Log Buffer Size [ 32 ] K    Dbospace Down Option [0]
Logical Log Buffer Size [ 32 ] K    Preserve Log For Log Backup [N]
Max # of Logical Logs [ 9 ]        Transaction Timeout [ 300 ]
Max # of Locks [ 2000 ]           Long TX HWM [ 50 ]
Max # of Buffers [ 200 ]          Long TX HWM Exclusive [ 60 ]
                                          Index Page Fill Factor [ 90 ]
                                          Add SegSize (K) [ 8192 ]
                                          Total Memory (K) [ 0 ]

Resident Shared Memory size [ 508 ] Kbytes      Page Size [ 2 ] Kbytes

```

Guarde los cambios presionando “**esc**” para finalizar presione “**n**” para no inicializar.

Antes de levantar la base de datos se necesita cambiar de usuario a los archivos, “**onconfig**” y “**sqlhosts**”. Para ello ejecute el siguiente comando:

```
chown informix.informix onconfig
```

```
chown informix.informix sqlhosts
```

Una vez realizados todos los pasos se procede a levantar la base de datos.

Ejecutar “**oninit -ivy**”.

Nota: el comando oninit -ivy inicializa toda la base de datos; tener precaución al usar este comando.

➤ **Crear Dbospace.**

1.- Ingresar a “**onmonitor**”, seleccionar dbospace -> Create.

2.- Ingresar el nombre del dbospace “**tempdbs**”, ingresar la ruta y el espacio dedicado del chunk.

➤ **Crear BlobSpace.**

1.- Para crear un blobspace, ejecutamos el siguiente comando.

```
onospace -c -S sbospace -o 0 -p /home/informix/chunks/sbospace.000 -s 20000
```

➤ **Migrar Source4gl desde servidor principal**

1.- Ingresar a “**r4gl**”.

2.- Seleccionar Modify-> yes->Crear base

3.- Se crea una base de datos con el siguiente nombre “**syspgm4gl**”

4.- Ingresar a la base de datos creada para ello ejecutar: **dbaccess syspgm4gl**

5.- Ingresar y ejecutar el siguiente query:

```
Insert into source4gl select * from syspgm4gl@ol_appserver:source4gl
```

➤ **Importar base de datos.**

Para importar las bases de datos, se necesitan copiar los respaldos del día anterior del servidor principal.

Estos se encuentran en **/appl/export/LUNES**, en el servidor principal

Cambiar de directorio **cd /appl**, en el nuevo servidor.

Crear el directorio export: **mkdir export**

Cambiar al directorio creado: **cd export**

Una vez en este directorio realizar la copia de los respaldos desde el servidor principal para ello ejecutar el siguiente código.

```
rcp -r appserver:/appl/export/LUNES .
```

Cuando se haya terminado de copiar la carpeta cambiamos de directorio: **cd LUNES**

Descomprimir las bases de datos: **gunzip -f /*gz***

Exportar las bases de datos en el siguiente orden:

- global
- comercial
- rolpago
- contab

Todas las bases se importan con el siguiente comando: **“dbimport -l nombrebase”**, excepto la base de datos comercial que se importa de la siguiente manera **“dbimport comercial”** una vez esta termine de subirse la ponemos en modo transaccional **“ontape -s -U”** comercial.

Por último se necesita actualizar las estadísticas para ello ingresamos a la base de datos y ejecutamos el siguiente query: **update statistics.**

1.7. Ejecución y resultados de las pruebas.

La fase de pruebas que se ejecutaran está descritas en la Tabla 40 y se llevara acabo de acuerdo a las siguientes prácticas:

- Buscar y documentar defectos en la calidad del software.
- Servir como guía para el logro de la calidad del software.
- Validar que el software funciona de acuerdo a las especificaciones que se hicieron en el diseño.

1.8. Pruebas unitarias.

El propósito de las pruebas unitarias es comprobar la funcionabilidad y estructura de cada componente individual de la solución software luego de haber finalizado su codificación en el lenguaje de programación seleccionado; este proceso técnico permitirá verificar que:

- Los programas fuentes estén libre de errores luego de su proceso de compilación
- Que los componentes funcionales se ejecuten correctamente
- Que los registros de información se almacenen o actualicen correctamente
- Que todas las transacciones se generen sin interrupciones
- Que los mecanismos de excepción, control y parametrización funcionen eficientemente.

Tabla 41. Estrategia de pruebas unitarias en la solución software.

Objetivo de la prueba:	Validar los componentes individuales de la solución software como una unidad independiente.
Estrategia:	Se realizan por cada componente en el cual se pueda identificar
	<ul style="list-style-type: none">➤ Que su ejecución cumpla con lo especificado.➤ Que las instrucciones de bifurcación obtengan uno o varios resultados.
Herramienta requerida:	SISTEMA OPERATIVO RED HAT, SQL, 4GL
Observaciones:	Ninguna

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Las configuraciones indicadas se realizaron de forma correcta, como se puede apreciar en la Figura 41.

```

DATABASE INFO:  dBspace  Nls Routine Databases Exit
Display DBSPACE information for a database.

----- rolpago01@ol_appserver ----- Press CTRL-W for Help

  Id      Name                Number of  When      Mirror
  --      -                -         -         -
   1      rootdbs                1         07/18/2015  N

```

Figura 41. Dbspace.
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

```

INFO - informix.reg_vacaciones:  ... References  Status  cOnstraints
Display status information for a table.

----- rolpago01@ol_appserver ----- Press CTRL-W for Help

Table Name      reg_vacaciones
Owner           informix
Row Size        104
Number of Rows  1582
Number of Columns  19
Date Created    15/07/20

```

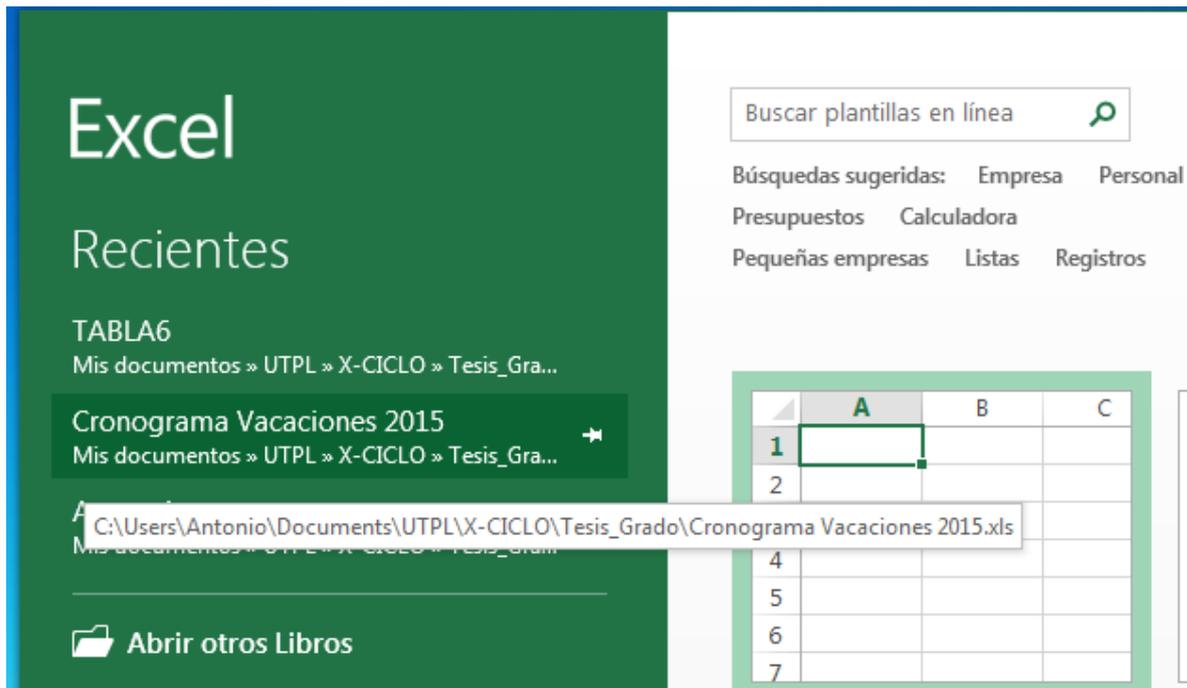
Figura 42. Status
Fuente: (Lloret Orellana, 2015)
Elaborado por: Lloret Antonio.

1.8.1. Pruebas unitarias de RF01 y M01.

El propósito de la siguiente prueba es realizar las pruebas unitarias al requisito funcional RF01 y módulo M01.

- **RF01:** Verificar cronograma anual de vacaciones
- **M01:** Cronograma anual de vacaciones

Una vez establecido el ambiente de pruebas se procedió con la codificación de la prueba unitaria mediante el uso del utilitario de Microsoft Office 2013 mediante la apertura de la hoja electrónica Excel, la misma que se detallan a continuación.



CRONOGRAMA DE VACACIONES AÑO 2015 imp. Com. El Hierro Cia. Ltda.

Nº	EM	SER	NOMBRES	F.ING.	CARGOS	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE
6	55	2075	Escobar Velasco Maria de los Angeles	05/05/2014	Cajero Vendedor												
7	55	1696	Maiza Lalaeo Monica Elizabeth	22/02/2011	Superv. Cajas												
8	55	1942	Haro Ortuño Patricio Bolivar	25/03/2013	Agente Ven. Ferreteria												
9	55	2106	Llerena Paredes Segundo Oswaldo	04/11/2014	Chofer												
10	55	2002	Munilo Zambrano Eulogio Manuel	12/11/2012	Empacador-Cargador												
11	55	2056	Nieto Fiallos Mario Oswaldo	03/03/2014	Empacador-Cargador												
12	55	1975	Portero Naranjo Klever Eduardo	03/03/2014	Empacador-Cargador												

EM	SER	NOMBRES	F.ING.	CARGOS	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE
1	58	1884	MEGAHIERRO YUNGULLA Amijos Valdívieso Luz Vicenta	17/08/2012	Ventas. Mostrador											
2	58	1764	Cajamarca Morales Thomay Eduardo	01/08/2011	Subjefe Almacén											
3	58	958	Chambo Suquilanda Silvia Ester	11/09/2014	Jefe Almacén											
4	58	1730	Merchan - Luis Guillermo	21/11/2011	Empacador-Cargador											
5	58	1779	Monge Chimboga Denis Jackeline	10/10/2011	Ventas Mostrador											
6	58	1762	Ortega Barros Jorge Eduardo	25/07/2011	Bodeguero											
7	58	1788	Suarez Morales Eddy Darwin	14/11/2011	Chofer											

EM	SER	NOMBRES	F.ING.	CARGOS	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE
1	40	1914	MEGAHIERRO GIRON Clavijo Araujo Edwin Fabiana	23/10/2012	Bodeguero											
2	40	1910	Pardo Pinos Juan Fernando	18/10/2012	Ventas Mostrador											
3	40	1913	Sanchez Cabrera Josa Victoriano	19/10/2012	Chofer											
4	40	1918	Vivar Marquez Santiago Agustin	14/11/2012	Empacador-Cargador											
5	40	2129	Castro Intiango Carlos Lenin	27/01/2015	Jefe Almacén											

EM	SER	NOMBRES	F.ING.	CARGOS	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE
1	02	893	GUARDIAS DE SEGURIDAD Amay Juncay Juan Pedro	12/03/2010	Guardia Mega Politécnica											
2	04	893	Barba Gomez Luis Gonzalo	12/03/2010	Guardia Mega Paute											
3	02	1611	Camion Edison Patricio	23/06/2010	Guardia Mega Politécnica											
4	02	2037	Espin Saldaña Luis Enrique	16/01/2014	Guardia Mega Politécnica											
5	02	2040	Garcia Carabazo Christian Jovanny	03/02/2014	Guardia Mega Politécnica											
6	02	1872	Giron Rosales Richard Francisco	18/06/2012	Guardia Local Principal											
7	03	1637	Pacheco Pinto Eduardo Fabian	02/09/2011	Guardia Mega Arenal											
8	02	502	Palacios Bravo Manuel Jesus	24/07/2002	Guardia Mega Principal											
9	21	490	Valdez Bustamante Mauro Antonio	01/07/2002	Guardia Mega Machala											
10	03	1766	Vazquez Cedillo Kleber Teodoro	28/02/2000	Guardia local el Arenal											
11	01	1462	Vazquez Guiracocha Juan Carlos	18/04/2009	Guardia Local Principal											

Elaborado por:
Departamento de Talento Humano

IMP. COM. EL HIERRO CIA. LTDA.
EXT. 7241-7242-7243
Fecha: 5-enero-2015

Figura 42. Cronograma anual de vacaciones
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

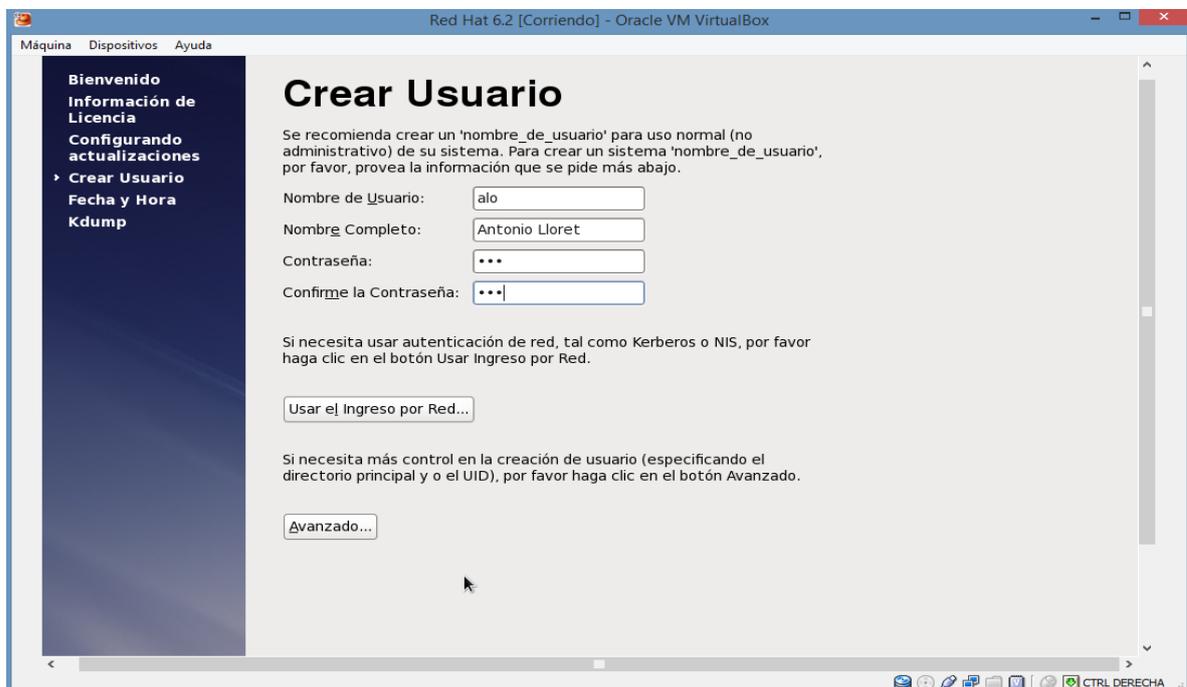
Las configuraciones indicadas se realizaron de forma correcta, como se puede apreciar en la Figura 42.

1.8.2. Pruebas unitarias de RF02 y M02.

El propósito de la siguiente prueba es realizar las pruebas unitarias al requisito funcional RF02 y módulo M02.

- **RF02:** Autenticación de usuarios
- **MO2:** Autenticación de usuarios.

Una vez establecido el ambiente de pruebas se procedió con la codificación de la prueba unitaria mediante el administrador del sistema, las mismas que se detallan a continuación.



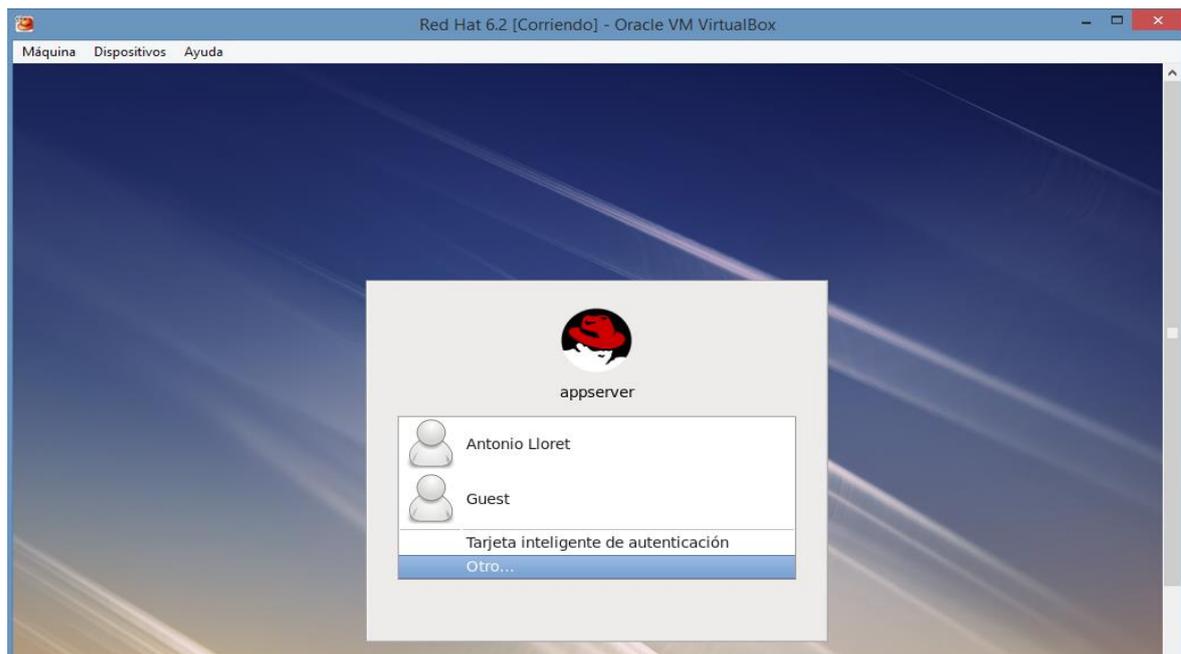


Figura 43. Creacion de usuarios.
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Las configuraciones indicadas se realizaron de forma correcta, como se puede apreciar en la Figura 43.

1.8.3. Pruebas unitarias de RF04-RF05 y M03.

El propósito de la siguiente prueba es realizar las pruebas unitarias al requisito funcional RF04-RF05 y módulo M03.

- **RF04:** Registro de vacaciones
- **RF05:** Anulación de vacaciones
- **M03:** Registro de vacaciones.

Una vez establecido el ambiente de pruebas se procedió con la codificación de la prueba unitaria mediante la ejecución del módulo programado en código fuente en 4GL y verificación de datos por medio de scripts en SQL y dbaccess.

La codificación de la prueba permite en primer lugar realizar la búsqueda de un empleado que se encuentre registrado en la base de datos.

Con la codificación de la presente prueba se pudo verificar lo siguiente:

CÓDIGO	DESCRIPCIÓN	ENTRADA	SALIDA
RF04	Búsqueda de registros	Apellido	OK
RF04	Búsqueda de registros	Serial	OK

En la codificación de la presente prueba permitió observar y verificar la información referente al registro de vacaciones del empleado seleccionado; como se puede apreciar en la Figura 44.



EMP	EMPLEADO	EST	CARGO	SAL	DIAS	FECHA ING	FECHA S.
01	246 SANCHEZ BERNAL MARIA ELE	OK	JEFE DE TEGNOL	25.0		00/07/01	
40	1913 SANCHEZ CABRERA JOSE VIC	OK	CHOFER DESPACH	0.0		12/10/19	

Ser.	Tipo	Tra	Fech/D	Fech Fin/D	Tra.Periodo	Vacac.	Ing	Egr	Saldo	Est.
1	SI	S.INIC	14/04/30	X			15.0	0.0	15.0	a * DK
2	V	VAC/CA	14/05/05	L 14/05/07	X		0.0	3.0	12.0	a * DK
3	V	VAC/CA	14/06/30	L 14/07/13	D		0.0	14.0	-2.0	n * DK
4	VA	VAC.AN	14/10/19	D	13/10/19-14/10/18		15.0	0.0	13.0	a * DK
5	V	VAC/CA	15/02/03	M 15/02/03	M		0.0	0.5	12.5	n * DK
6	V	VAC/CA	15/05/19	M 15/05/31	D		0.0	12.5	0.0	n * DK

Figura 44. Registro de vacaciones
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Con la codificación de la presente prueba se pudo verificar lo siguiente:

CÓDIGO	DESCRIPCIÓN	ENTRADA	SALIDA
RF04	Ingreso de registros	Apellido	OK
RF04	Ingreso de registros	Serial	OK

En la codificación de la presente prueba permitió observar y verificar la información referente al registro de vacaciones del empleado seleccionado; como se puede apreciar en la Figura 45.

```

CONTROL VACACIONES:  Ingresar  Buscar  Modificar  imprimir  aNular  ...
ROLPAGO CONTROL VACACIONES PARA EMPLEADOS MEGA PRINCIPAL MATRI
2015 rolpago01
EM
CA ROLPAGO REG. TRANSACCION VACACIONES MEGA PRINCIPAL MATRI
2015 rolpago01
S EMPLEADO: 1913 SANCHEZ CABRERA JOSE VICTORIA 0702709106 GRUPO : 1
CARGO : CHDE CHOFER DESPACHADO EMP: 40 F5 GIRON ALM. EST : OK
FECH. ING: 12/10/19 FECH.SAL: S.DIAS: 0.00
DATOS DE LA TRANSACCION
NO.REGISTRO : 6 TIPO: J VAC/CARG
FECHA INICIO : 15/05/19 DIA : M MARTES JORNADA: M MEDIO DIA T
FECHA TERMINO: 15/05/31 DIA : D DOMINGO JORNADA: C DIA COMPLETO T
PERIODO VACAC: DOCUMENTO PAGO:
DIAS INGRESO : 0.0 DIAS EGRESO: 12.5 DIAS SALDO: 0.0
USUARIO CR/MO: ng1 FECHA MODIF: 2015-05-25 11:25:26 EST: OK

Ingresando datos <ESC> Grabar CTRL-C> Cancela

```

Figura 45. Registro de vacaciones.

Fuente: (Lloret Orellana, 2015)

Elaborado por: Lloret Antonio.

Con la codificación de la presente prueba se pudo verificar lo siguiente:

CÓDIGO	DESCRIPCIÓN	ENTRADA	SALIDA
RF05	Anulación de registros	Apellido	OK
RF05	Anulación de registros	Serial	OK

En la codificación de la presente prueba permitió observar y verificar la información referente al proceso de anulación de un registro de vacaciones del empleado seleccionado; como se puede apreciar en la Figura 46.

CONTROL VACACIONES: Ingresar Buscar Modificar imprimir aNular ...

ROLPAGO ANULANDO REGISTRO DE VACAMEGA P INCIPAL MATRI MATRI

2015 rolpago01 rolpago01

EMPLEADO: 1913 SANCHEZ CABRERA JOSE VICTORI 0702709106 GRUPO: 1 ESTA.: DK
 CARGO: CHDE CHOFER DESPACHAD EMP: 40 F.ING: 12/10/19 F.SAL: -1.0

Ser.	Tipo	Tra	Fech/D	Fech Fin/D	Tra.Periodo	Vacac.	Ing	Egr	Saldo	Est.
1	SI	S.INIC	14/04/30	X			15.0	0.0	15.0	a * DK
2	V	VAC/CA	14/05/05	L 14/05/07	X		0.0	3.0	12.0	a * DK
3	V	VAC/CA	14/06/30	L 14/07/13	D		0.0	14.0	-2.0	n * DK
4	VA	VAC.AN	14/10/19	D	13/10/19-14/10/18		15.0	0.0	13.0	a * DK
5	V	VAC/CA	15/02/03	M 15/02/03	M		0.0	0.5	12.5	n * DK
6	V	VAC/CA	15/05/19	M 15/05/31	D		0.0	12.5	0.0	n * DK
7	V	VAC/CA	15/07/10	V 15/07/10	V		0.0	1.0	-1.0	a * DK

Esta seguro de anular el registro en la linea ? S/N?

CONTROL VACACIONES: Ingresar Buscar Modificar imprimir aNular ...

ANULAR TRANSACCION VACACIONES

2015 rolpago01 rolpago01

EMPLEADO: 1913 SANCHEZ CABRERA JOSE VICTORI 0702709106 GRUPO: 1 ESTA.: DK
 CARGO: CHDE CHOFER DESPACHAD EMP: 40 F.ING: 12/10/19 F.SAL: 0.00

Ser.	Tipo	Tra	Fech/D	Fech Fin/D	Tra.Periodo	Vacac.	Ing	Egr	Saldo	Est.
1	SI	S.INIC	14/04/30	X			15.0	0.0	15.0	a * DK
2	V	VAC/CA	14/05/05	L 14/05/07	X		0.0	3.0	12.0	a * DK
3	V	VAC/CA	14/06/30	L 14/07/13	D		0.0	14.0	-2.0	n * DK
4	VA	VAC.AN	14/10/19	D	13/10/19-14/10/18		15.0	0.0	13.0	a * DK
5	V	VAC/CA	15/02/03	M 15/02/03	M		0.0	0.5	12.5	n * DK
6	V	VAC/CA	15/05/19	M 15/05/31	D		0.0	12.5	0.0	n * DK
7	V	VAC/CA	15/07/10	V 15/07/10	V		0.0	1.0	-1.0	a * AN

Registro ANULADO, saldo actualizado en reg. personal

Figura 46. Proceso de anulación de registro de vacaciones.

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

Finalmente vale destacar que para la ejecución de las pruebas unitarias en el proceso de codificación en 4GL se incluyen rutinas de cálculo para asignación automática de los días de vacación en función de la fecha de ingreso del empleado, de igual forma se incluyen rutinas de validación de fechas de ingreso, fechas de salida, estado del empleado activo o anulado,

tipo de transacción, etc. Lo cual permitió verificar el funcionamiento de cada requisito funcional y el modulo principal de la aplicación. También es importante mencionar que se realizaron scripts en SQL con el fin de comprobar que la información este guardada y disponible en la base de datos.

2. Pruebas de integración.

La prueba de integración es una técnica para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. Esta prueba permite validar la integración entre las subrutinas que dispone el modulo principal, es decir su estructura de control, se revisa todas las decisiones lógicas en sus lados verdadero y falso, que se ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas y se revisen estructuras de datos internas para garantizar su validez.

En las pruebas de integración se deben considerar algunos aspectos técnicos que se mencionan a continuación:

- El funcionamiento de las subrutinas que integran el modulo principal deben estar libres de errores.
- Verificar todas las dependencias entre las subrutinas.
- Comprobar consistencia e integridad de la base de datos.
- Verificar asignación y configuración de recursos de software y hardware.
- Probar todo el flujo de control y flujo de datos por medio de todos los componentes del patrón arquitectónico cliente-servidor.

Tabla 42. Estrategia para las pruebas de integración.

Objetivo de la Prueba:	Validar la integración entre las subrutinas que componen el modulo principal de la solución software con el propósito de garantizar que su operación integrada es adecuada.
Estrategia:	Se generan pruebas de integración que permita: <ul style="list-style-type: none"> ➤ Ejecutar subrutinas que se acoplan correctamente en la aplicación. ➤ Verificar la llamada entre subrutinas ➤ Se prueban las subrutinas que se integren y se adapten a la aplicación.
Herramienta requerida:	4GL
Observaciones:	Las pruebas se realizaran a las siguientes subrutinas: <ul style="list-style-type: none"> ➤ Consulta de registros

	<ul style="list-style-type: none"> ➤ Modificación de registros ➤ Búsqueda de registros ➤ Ingreso de registros ➤ Anulación de registros
--	--

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Para la ejecución de las pruebas de integración se usó la misma configuración del ambiente de pruebas aplicado en las pruebas unitarias y se generó en 4GL.

La codificación de las pruebas de integración permitió comprobar de manera conjunta la integración entre las diferentes subrutinas de programación, como se demuestra en el resumen siguiente:

```

FUNCTION mantenimiento_regvac()
  OPEN WINDOW win_menucar AT 2,3 WITH 2 ROWS, 75 COLUMNS
  ATTRIBUTE(BORDER)
  OPEN WINDOW win_formcar AT 4,2 WITH FORM "mnt_regvac" ATTRIBUTE(BORDER)
  CURRENT WINDOW IS win_menucar
  LET cnt_actual = 0
  LET gl_rvc_serial = NULL
  MENU "CONTROL VACACIONES"
  BEFORE MENU
    HIDE OPTION "Modificar"
    HIDE OPTION "impRimir"
    HIDE OPTION "aNular"
    HIDE OPTION "siGuiente"
    HIDE OPTION "Anterior"
    HIDE OPTION "Primero"
    HIDE OPTION "Ultimo"
  COMMAND "Ingresar" "REGISTRAR UNA NUEVA TRANSACCION"
  CURRENT WINDOW IS win_formcar
    CALL inicializa_regvac()
    CALL ingresa_regvac()
    CALL asigna_regvac()
    MESSAGE ""
  CURRENT WINDOW IS win_menucar
  IF cnt_actual != 0
  THEN
    SHOW OPTION "Modificar"
    SHOW OPTION "impRimir"
    SHOW OPTION "aNular"
    SHOW OPTION "Ultimo"
    SHOW OPTION "Primero"
    SHOW OPTION "Anterior"
    SHOW OPTION "siGuiente"
  END IF
  COMMAND "Buscar" "BUSCAR UN REGISTRO DE VACACIONES"
  CURRENT WINDOW IS win_formcar
    CALL busca_regvac()

```

```

        CALL asigna_regvac()
        MESSAGE ""
CURRENT WINDOW IS win_menucar
IF cnt_actual != 0
THEN
    SHOW OPTION "Modificar"
    SHOW OPTION "impRimir"
    SHOW OPTION "aNular"
    SHOW OPTION "Ultimo"
    SHOW OPTION "Primer"
    SHOW OPTION "Anterior"
    SHOW OPTION "siGuiente"
END IF
COMMAND "Modificar" "MODIFICAR REGISTRO DE VACACIONES"
CURRENT WINDOW IS win_formcar
    CALL actualiza_regvac()
    CALL asigna_regvac()
    MESSAGE ""
CURRENT WINDOW IS win_menucar
COMMAND KEY(R) "impRimir" "IMPRIME REGISTRO DE VACACIONES"
CURRENT WINDOW IS win_formcar
    CALL imprime_regvac()
COMMAND KEY(N) "aNular" "ANULAR TRANSACCION VACACIONES"
CURRENT WINDOW IS win_formcar
    CALL anula_regvac()
    CALL asigna_regvac()
CURRENT WINDOW IS win_menucar
COMMAND KEY(G) "siGuiente" "EXHIBE EL SIGUIENTE REGISTRO DE LA LISTA"
CURRENT WINDOW IS win_formcar
    CALL siguiente_regvac()
CURRENT WINDOW IS win_menucar
COMMAND "Anterior" "EXHIBE EL ANTERIOR REGISTRO DE LA LISTA"
CURRENT WINDOW IS win_formcar
    CALL anterior_regvac()
CURRENT WINDOW IS win_menucar
COMMAND "Primer" "EXHIBE EL PRIMER REGISTRO DE LA LISTA"
CURRENT WINDOW IS win_formcar
    CALL primer_regvac()
CURRENT WINDOW IS win_menucar
COMMAND "Ultimo" "EXHIBE EL ULTIMO REGISTRO DE LA LISTA"
CURRENT WINDOW IS win_formcar
    CALL ultimo_regvac()
CURRENT WINDOW IS win_menucar
COMMAND KEY(S,INTERRUPT) "Salir" "RETORNA AL MENU ANTERIOR"
LET INT_FLAG=FALSE
EXIT MENU
END MENU
CLOSE WINDOW win_menucar
CLOSE WINDOW win_formcar
END FUNCTION

```

3. Pruebas de Sistema.

El propósito de las pruebas de sistema en realidad es un conjunto de acciones que permitan ejercitar la solución software basado en computadora, además verificar que los elementos del sistema se hayan integrado de manera adecuada y que realicen las funciones designadas (Ver anexo E). En la Tabla 43 se refleja la estrategia a utilizar:

Tabla 43. Estrategia para las pruebas de sistema.

Objetivo de la prueba:	Validar que los volúmenes de datos se generen dentro de un periodo determinado de tiempo, con un nivel de concurrencia específico, y validar los requerimientos no funcionales de la aplicación.
Estrategia:	Se realizan pruebas de sistema que permitan entre otras cosas: <ul style="list-style-type: none">➤ Simular la carga esperada en un entorno de producción.➤ Probar cada uno de los requerimientos no funcionales configurados para la aplicación.➤ Identificar problemas de rendimiento
Herramienta requerida:	SQL scripts
Observaciones:	Las pruebas requeridas son: <ul style="list-style-type: none">➤ Prueba de Estrés➤ Prueba de interfaz de usuario➤ Prueba de seguridad➤ Prueba de rendimiento

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

3.1. Prueba de estrés.

Los test de estrés, son test diseñados para verificar el comportamiento de un sistema, y el objetivo es comprobar la robustez del software cuando es sometido a condiciones diferentes a la de operación normal. Estos test son particularmente importantes para software considerado crítico, y hacen énfasis en características como robustez, disponibilidad o manejo controlado de errores y situaciones de excepción. Generalmente el tiempo de respuesta es uno de los requerimientos no funcionales de sistemas cliente-servidor o servidor, e impacta directamente en la agilidad percibida por el usuario final. Es necesario definir y verificar los requerimientos mínimos determinando de antemano el ambiente y un perfil de uso. A

continuación se detalla algunos comandos de administración de la base de datos de Informix que nos permiten observar la disponibilidad de la base de datos:

Onmode

Oninit

Finderror

Ontape -s -L 0 -U [database]

Dbimport

Onstat -d

Update statistics

```
IBM Informix Dynamic Server Version 11.70.FC5GE -- On-Line -- Up 04:10:47 -- 31s

Dbspaces
address          number  flags      fchunk  nchunks  pgsize  flags      owner e
c98f7028         1      0x60001    1       1        2048    N BA      informs
cbd40df8         2      0x68001    2       1        2048    N SBA     informe
cbd43028         3      0x42001    3       1        2048    N TBA     informs
 3 active, 2047 maximum

Chunks
address          chunk/dbs  offset  size      free      bpages  flage
c98f71d0         1 1        0        49000000 27861208 PO-B0
cbd431d0         2 2        0        1000000   932620   932620  POSB0
                Metadata 67327    50099    67327
cbd433d0         3 3        0        1000000   999947   PO-B0
 3 active, 32766 maximum

NOTE: The values in the "size" and "free" columns for DBspace chunks are
      displayed in terms of "pgsize" of the DBspace to which they belong.

Expanded chunk capacity mode: always
```

Informix siempre persiste los datos escribiendo a alguno de sus buffers de log lógico. Si la base de datos en la cual se están haciendo cambios está trabajando en modo unbuffered, estos cambios se escriben al disco duro inmediatamente después de que un registro de commit o rollback llega al log lógico. Si la base de datos está trabajando en modo buffered, el servidor mantiene estos cambios se mantienen en el log el mayor tiempo posible, hasta que ocurre alguno de los siguientes eventos:

- Un buffer se llena
- Se ejecuta un checkpoint
- Se cierra la conexión contra la base de datos
- Se hace un commit a una base que está trabajando en modo unbuffered

En el modo buffered, existen problemas si el sistema se bloquea antes de que el buffer de log lógico sea escrito al disco duro pero después de la transacción que el usuario completó ejecutando un commit. La transacción se considerará incompleta y se ejecutará un rollback cuando el motor sea reiniciado y se recupere. En modo unbuffered esto no sucede ya que luego del commit se escribirán los datos de la transacción en el disco duro y no se perderán. Por defecto, se trabaja en modo unbuffered. Si bien retarda un poco el motor, es más robusto que el modo buffered frente a caídas del motor en lo que a pérdida de datos respecta. El tamaño de los buffers de logs lógicos se pueden averiguar en el archivo onconfig (parámetro LOGBUFF).

3.2. Prueba de interfaz de usuario.

El objetivo de las pruebas de interfaz de usuario es verificar la navegación por medio de elementos que se están ejecutando y que reflejan las funciones del negocio a través de los requerimientos funcionales. Para la solución software se utilizó la herramienta de emulación NetTerm (Ver Figura 47) que es un terminal de red que puede actuar como un cliente telnet, la terminal de conexión directa y módem conectado al terminal de anfitriones compatibles con el protocolo telnet o módems dial-in. NetTerm apoya contraseña, SRP y la autenticación de Microsoft NTLM. El método de autenticación seleccionado se determina de forma automática a través de una serie de intercambios de mensajes con el anfitrión durante la fase de conexión. Para mayor detalle de cómo se encuentra el diseño de las interfaces de la aplicación remitirse al anexo I en donde se explica las acciones que el usuario debe realizar para ejecutar la aplicación.

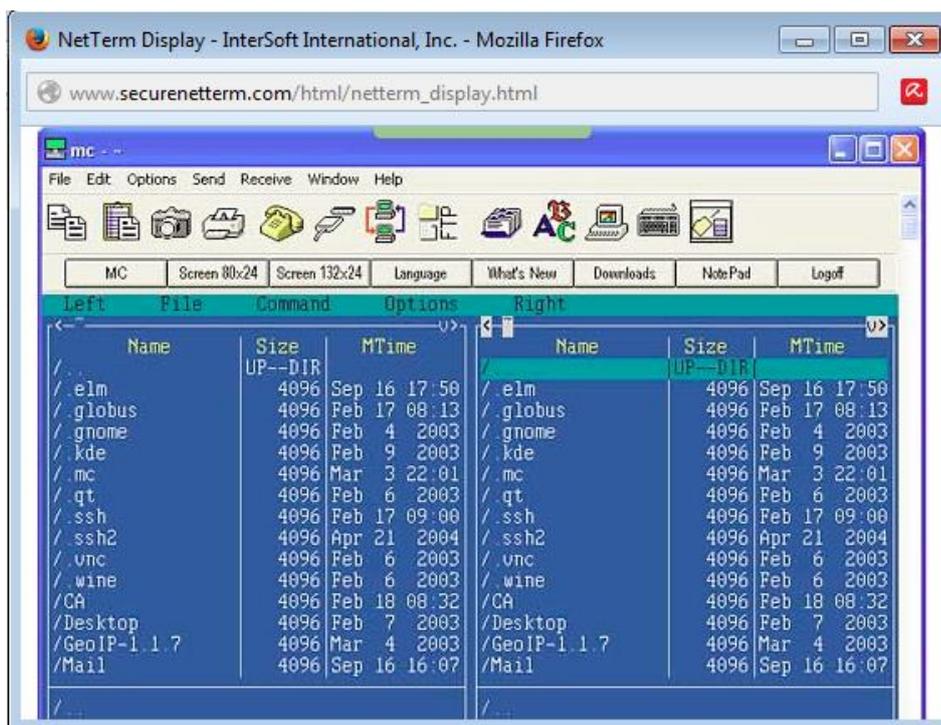


Figura 47. NetTerm

Fuente: (<http://www.securenetterm.com/html/netterm.html>, 2015).

Elaborado por: Lloret Antonio.

Otra de las herramientas que se probaron en la ejecución de las pruebas de estrés y que permitieron a los usuarios navegar por cada una de las opciones de la aplicación es por medio la utilidad que proporciona Informix denominada DB-Access que proporciona una interfaz controlada por menús para entrar, ejecutar y depurar sentencias de Lenguaje de consulta estructurado (SQL) y rutinas de Lenguaje de procedimientos almacenados (SPL). También puede ejecutar DB-Access de manera interactiva desde la línea de mandatos. El programa de utilidad DB-Access se incluye con el servidor Informix y con Informix Client Software Development Kit. En un entorno UNIX, el servidor de bases de datos debe tener el terminal correspondiente que se ha configurado entre los terminales que muestra la variable de entorno INFORMIXTERM.

La interfaz de menús de pantalla completa de DB-Access sirve de guía en la ejecución de sentencias SQL.

La interfaz de usuario de DB-Access combina las características siguientes:

- Una jerarquía de menús
- Pantallas que le solicitan respuestas breves y elecciones en listas de selección

- Pantallas de AYUDA según contexto
- El Editor de esquemas interactivo que le ayuda a estructurar Tablas
- Un entorno de programador de SQL, que incluye las características siguientes:
 - El editor de SQL incorporado donde se entran y modifican las sentencias SQL y SPL.
 - Una opción para utilizar otro editor de su elección
 - El depurador en tiempo de ejecución y el comprobador de sintaxis del servidor de bases de datos
 - Almacenamiento, recuperación y ejecución de rutinas SQL y SPL
- Una opción de salida para consultas e informes de base de datos.

El uso de la interfaz descrita permitió a los usuarios navegar por las diferentes opciones de la aplicación y comprobar que:

- La aplicación de escritorio se adapta de manera correcta a cualquier pantalla de los equipos de computación que dispone la Empresa.
- La navegación entre las opciones del menú de la solución software no reflejaron ningún problema.
- Ejecución de las funcionalidades sin ningún problema
- Colores monocromáticos de la interfaz de usuario con nitidez
- Mensajes de error y alerta de acuerdo a lo programado.

3.3. Pruebas de rendimiento.

Las pruebas de rendimiento se diseñan para poner a prueba el rendimiento del software en tiempo de ejecución o corrida dentro del contexto de un sistema integrado, por lo tanto es necesario medir la utilización de los recursos como ciclos de procesador, interrupciones, intervalos de ejecución, conforme ocurren, y los muestreos del estado de la máquina de manera regular, esto permite descubrir situaciones que derivan a degradación y fallas en el sistema. Para verificar estos recursos se realizó la ejecución y funcionamiento del aplicativo que arrojaron los siguientes resultados:

Tabla 44. Tiempos de respuesta en las pruebas de rendimiento.

ACCION EJECUTADA	TIEMPO DE RESPUESTA
Ingreso a la aplicación	1 segundo

Búsqueda de registros	2 segundos
Consulta de registros	1 segundo
Modificación de registros	1 segundo
Ingreso de registros	2 segundos
Anulación de registros	1 segundo
Impresión de registros	1 segundo

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

Tiempo promedio 1.29 segundos

3.4. Pruebas de seguridad.

La prueba de seguridad intenta verificar que los mecanismos de protección cumplan con su función de impedir cualquier acción de penetración impropia que traten de vulnerar y dañar la integridad del aplicativo. En el caso de Informix las características de seguridad incluyen la réplica de datos de alta disponibilidad y el control de accesos avanzado para obtener un mayor control de los datos sensibles.

El software Informix se puede instalar y administrar sin privilegios root en sistemas operativos UNIX y Linux. No es necesario que el usuario que realiza la instalación no root sea un administrador del sistema y este usuario es el administrador del servidor de bases de datos (DBSA) para que el servidor de bases de datos pueda ejecutarse en el sistema sin el usuario Informix y el grupo Informix.

Informix maneja distintos niveles de seguridad y privilegios que son administrados por el administrador de la base de datos a nivel de seguridad son: base de datos, Tablas y columnas y privilegios que se describen en la siguiente Tabla:

	Connect	Resource	DBA
Acceso a las tablas de la BD			
Crear vistas y tablas temp.	X	X	X
Crear y borrar tablas e índices, alter tabla		X	X
Otorgar y revocar privilegios			X
Borrar la Base de Datos			X

Privilegios a nivel de Base de Datos:

Ejemplo: grant connect to public

 grant resource to Antonio

 grant DBA to Daniel

 revoke connect from public

 revoke resource from Antonio

 revoke dba from Daniel

Para acceder a una Base de datos se debe tener como mínimo privilegios de connect; el DBA es quien puede otorgar o revocar estos privilegios.

Privilegios a nivel de Tabla / columna:

Pueden ser de tipo: Alter, delete, index, select, update, insert, reference, all.

Ejemplo: grant all on table to public

 grant update on table to usuario1 with grant option.

 grant insert, delete on table to usuario1 as usuario2

El usuario puede otorgar con la opción with grant permiso a otro usuario; AS hace otorgador del permiso a otro usuario.

Ejemplo: revoke all on Tabla from public

 Revoke update on table from public

Para otorgar privilegios sobre columnas se usan las sentencias select, update y references.

Grant select (apellido, nombres) on personal to public

Grant insert, update (fecha_registro), select on ingreso to public

3.5. Pruebas funcionales.

Las pruebas funcionales permiten verificar que los casos de uso (ver Anexo F) sean los requeridos por la Empresa, para lo cual se desarrollan todas la pruebas necesarias que permitan asegurar el correcto funcionamiento de los flujos determinados en los casos de uso y confirmar el cumplimiento de todos los requerimientos funcionales establecidos.

Tabla 45. Estrategia para las pruebas funcionales:

Objetivo de la prueba:	Verificar el ingreso de datos, su procesamiento y los resultados que emitan estos requerimientos funcionales.
Estrategia:	Ejecutar pruebas de sistema que permitan: <ul style="list-style-type: none"> ➤ Constatar mensajes de error cuando los datos ingresados no sean correctos o fuera de formato. ➤ Verificar la lógica del negocio de la aplicación
Herramienta:	Lista de verificación
Observaciones:	Se aplicaran la prueba de usabilidad y de aceptación.

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Tabla 46. Resultados de las pruebas funcionales:

CU01	Cronograma de Vacaciones
Descripción:	El usuario constata que esté instalado el utilitario Microsoft Office y procede a abrir la hoja electrónica donde consta el cronograma anual de vacaciones.
Condiciones de Ejecución:	El usuario debe: <ul style="list-style-type: none"> ➤ Hacer click en el icono de Excel ➤ Abrir la hoja cronograma.xsl
Entrada:	Archivo Abrir cronograma
Resultado esperado:	Datos de los empleados: Empresa, serial, nombres, fecha de ingreso, cargo, mes de periodo de vacación
Excepciones:	ninguna
Resultado de prueba:	Prueba satisfactoria.

CU02	Control de acceso
-------------	--------------------------

Descripción	Por medio del sistema operativo Linux el usuario registra su username y password.
Condiciones de ejecución:	El usuario previamente debe estar registrado por el administrador del sistema.
Entrada:	El usuario debe digitar su username y clave: <ul style="list-style-type: none"> ➤ Login: ➤ Password:
Resultado esperado:	Acceso al módulo principal de la solución software
Excepciones:	Si el username y/o password son incorrectos el sistema operativo emitirá un mensaje de alerta indicando el error producido y solicitara un nuevo intento.
Resultado de prueba:	Prueba satisfactoria.

CU03	Registro de datos
Descripción:	Permite al usuario las siguientes funcionalidades: <ul style="list-style-type: none"> ➤ Ingresar ➤ Buscar ➤ Modificar ➤ Imprimir ➤ Anular
Condiciones de ejecución:	El usuario debe: <ul style="list-style-type: none"> ➤ Ejecutar el aplicativo ➤ Seleccionar opción del menú principal ➤ Seleccionar empleado a registrar
Entrada y proceso:	Dependiendo de la opción seleccionada, el usuario podrá ingresar, buscar, modificar, imprimir o anular un registro de control de vacaciones, para lo cual deberá ingresar la información que le solicita el aplicativo y procederá de manera automática con la validación de los campos registrados y finalmente presentara la información pertinente para su posterior aceptación o rechazo por parte del usuario.
Resultado esperado	Presentación y/o grabación del registro de vacaciones
Excepciones:	Si la información ingresada no cumple con las normativas legales y de validación, el registro no será procesado y emitirá los mensajes de alerta que sean necesarios para resguardar la integridad de la información a ser procesada.
Resultado de prueba:	Prueba satisfactoria

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Anexo I: Documento de manual de usuario

**Desarrollo de una solución software empresarial para control de vacaciones anuales
en la Empresa Importadora Comercial El Hierro Cía. Ltda.**

Manual de Usuario

Versión 1.1

Historial de Revisiones

Fecha	Versión	Descripción	Autor
15-ABRIL-2015	1.0	Versión Inicial planteada como propuesta de implementación.	Antonio Lloret Orellana
15-JUNIO-2015	1.1	Correcciones realizadas al documento	Antonio Lloret Orellana
6-AGOSTO-2015	1.2	Correcciones realizadas al documento	Antonio Lloret Orellana
21-SEPT-2015	1.3	Correcciones realizadas al documento	Antonio Lloret Orellana

1. Introducción.

El propósito de este documento también denominado guía del usuario, es brindar una información detallada del funcionamiento del aplicativo, de tal manera que el usuario pueda consultar y hacer uso en cualquier momento sobre las características que dispone la solución software, su simbología, indicaciones, requisitos, validaciones, funciones, etc. que le permitirán aprovechar los recursos y bondades que presenta la aplicación desarrollada para llevar un control de vacaciones anuales de la Empresa Imp.Com. El Hierro. De igual forma este documento permite transmitir la estructura, conceptos, e información básica del aplicativo que servirá como material de capacitación previa de los asistentes del departamento de Talento Humano que son los usuarios finales para el manejo del módulo de control de vacaciones,

2. Ejecución de la aplicación.

Este documento considera cada aspecto de la aplicación, iniciando con el ingreso al sistema, para luego describir cada una de las opciones que dispones la aplicación y su respectiva función.

2.1. Convenciones sobre el uso del teclado.

En la Figura 48 se presenta nomenclatura del teclado.

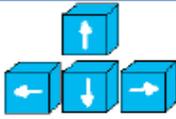
Tecla	Significado
	Teclas que se utilizan como método abreviado para ejecutar un proceso.
	Tecla que se utiliza para confirmar y ejecutar un proceso. Si en el manual dice "Presione Esc".
	Tecla que se utiliza para seleccionar opciones. Si en el manual dice "Presione Enter".
	Puede utilizar las teclas direccionales para moverse (Izquierda, Derecha, Arriba, Abajo) en las diferentes opciones de la pantalla un elemento a la vez.
Otras Teclas	Adicionalmente puede utilizar otros elementos del teclado. Por Ejemplo: CTRL + C es utilizado para cancelar una acción y regresar al punto anterior.

Figura 48. Nomenclatura del teclado.
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

2.1. Interfaz NetTerm.

Se trata de un emulador robusto, la terminal de acceso es configurable y está diseñada para conexiones de internet e intranet con soporte para Telnet, rlogin, acceso telefónico, y otros protocolos de conexión.

NetTerm opera en un entorno de Windows XP o versiones superiores, sus requerimientos de hardware son mínimos procesador Pentium o superior, memoria de 256 MB de RAM o superior, tarjeta de video, resolución mínima de 800x600 pixeles o superior.

Componentes de la barra de herramientas estándar del programa

En la Figura 49 se describe los componentes de la barra de herramientas:

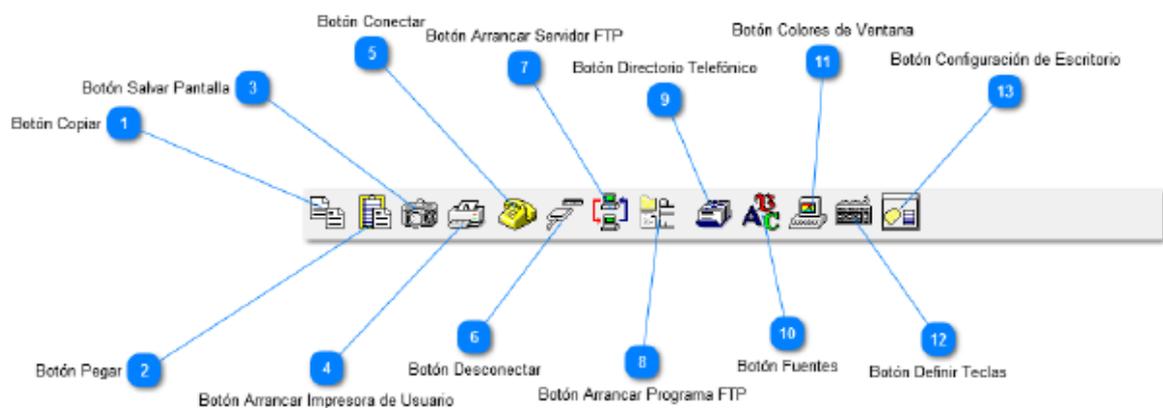


Figura 49. barra de herramientas.
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

Botones

Los botones que componen la barra de herramientas permiten acceder de manera rápida a la configuración del terminal y otras tareas comunes de edición. En la Tabla 47 se describe las acciones y funciones que realiza cada botón.

Tabla 47. Imagen, acción y función del panel de botones.

IMAGEN	ACCION	DESCRIPCION
	Copiar	Copia un texto seleccionado al portapapeles de Windows.
	Pegar	Pega el texto contenido en el portapapeles en donde está ubicado el cursor.
	Salvar Pantalla	Guarda el contenido de la pantalla en un archivo de texto.
	Arrancar impresora de usuario	Inicia la impresora de un usuario dentro de la red
	Conectar	Conecta el emulador NetTerm con el sistema de la empresa.
	Desconectar	Desconecta el emulador NetTerm del sistema de la empresa.
	Arrancar servidor FTP	Iniciar el servidor FTP de archivos.
	Arrancar programa FTP	Inicia el programa FTP de archivos.
	Directorio telefónico	Muestra el directorio telefónico de la empresa.
	Fuentes	Cambia el tipo de fuente del programa.
	Colores de pantalla	Cambia el color de la pantalla del programa.
	Definir teclas	Cambia las combinaciones de teclas del programa.
	Configuración de escritorio	Configura el escritorio del programa a gusto del usuario.

Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

3. Ingreso al programa.

En el escritorio de Windows previamente debe estar instalado y generado el acceso directo del emulador NetTerm, seguidamente procedemos de la siguiente manera:

- El usuario se ubica en la barra de tareas de Windows y procede con la búsqueda del acceso directo de NetTerm con la siguiente secuencia:

Iniciar > Todos los programas > InterSoft International, Inc > NetTerm

- También puede directamente presionar doble click sobre el icono de NetTerm.



- Inmediatamente aparecerá la siguiente pantalla de acceso.

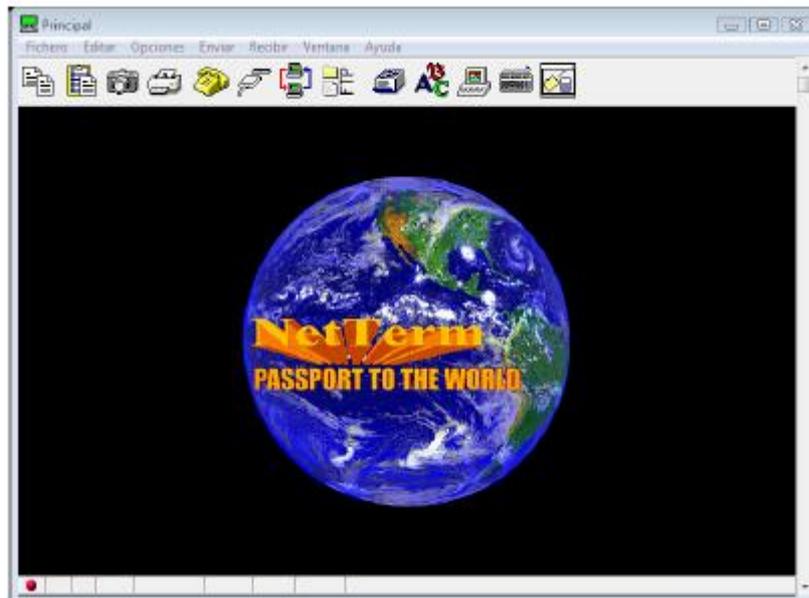


Figura 50. Pantalla de acceso
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

3.1. Ingreso al sistema.

El usuario selecciona el botón conectar  (Ver Figura 50) y el sistema le solicitará ingresar su login (usuario) y password (contraseña) como se visualiza en la Figura 51; en caso de que el usuario o contraseña sean incorrectos, el sistema desplegará un mensaje de error.

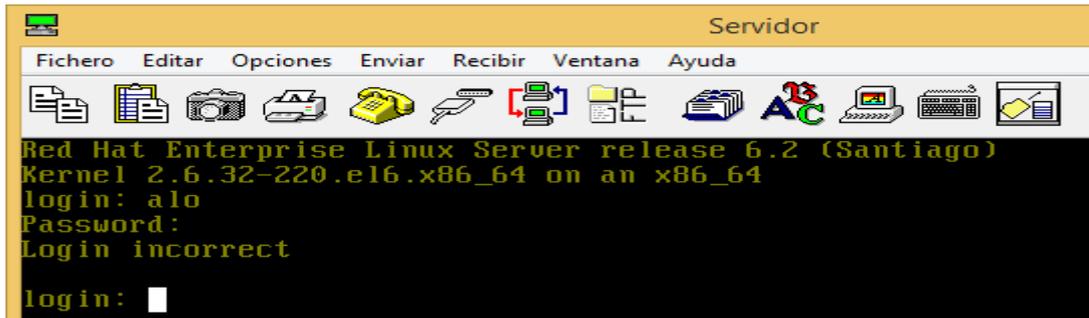


Figura 51. Mensaje de Error

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

En caso de que el login y password estén correctos, aparecerá una pantalla de bienvenida con un listado de los empleados que cumplen años en el mes corriente; para omitir dicho detalle se puede presionar la combinación de teclas ctrl-C. Ver Figura 52.



Figura 52. Bienvenida al sistema.

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

Seguidamente se presenta las aplicaciones disponibles que contienen el modulo principal de la Empresa.



Figura 53. Aplicaciones disponibles.

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

La aplicación desarrollada se encuentra localizada luego de seleccionar el módulo rolpago, seguidamente seleccionamos la Empresa en la cual deseamos procesar la información de Control de Vacaciones; cabe anotar que el sistema es multi empresarial, es decir permite trabajar en varias empresas que pertenecen al grupo corporativo de la familia Matute Zamora dueña de la Empresa Imp. Com. El Hierro. (Ver Figura 54).

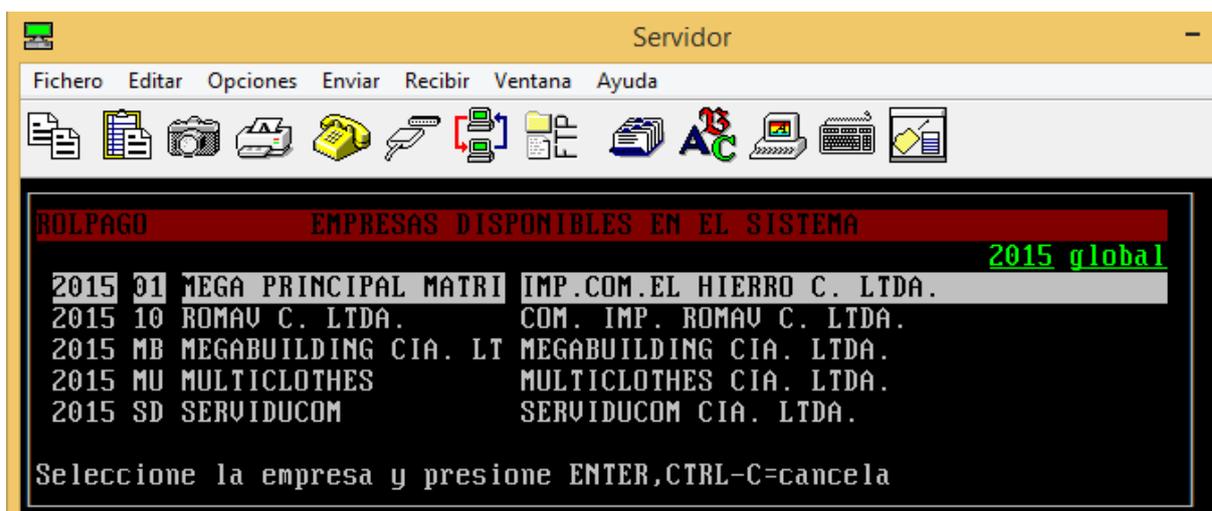


Figura 54. Empresas del grupo corporativo.

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

Una vez seleccionada la Empresa; el usuario de la aplicación debe tener los permisos y accesos correspondientes, de no disponer estas autorizaciones se visualiza un mensaje de alerta.

```

...Usuario no autorizado CONSULTE CON EL ADMINISTRADOR
    
```

A continuación se presenta el menú principal de la solución software.

4. Control de vacaciones.

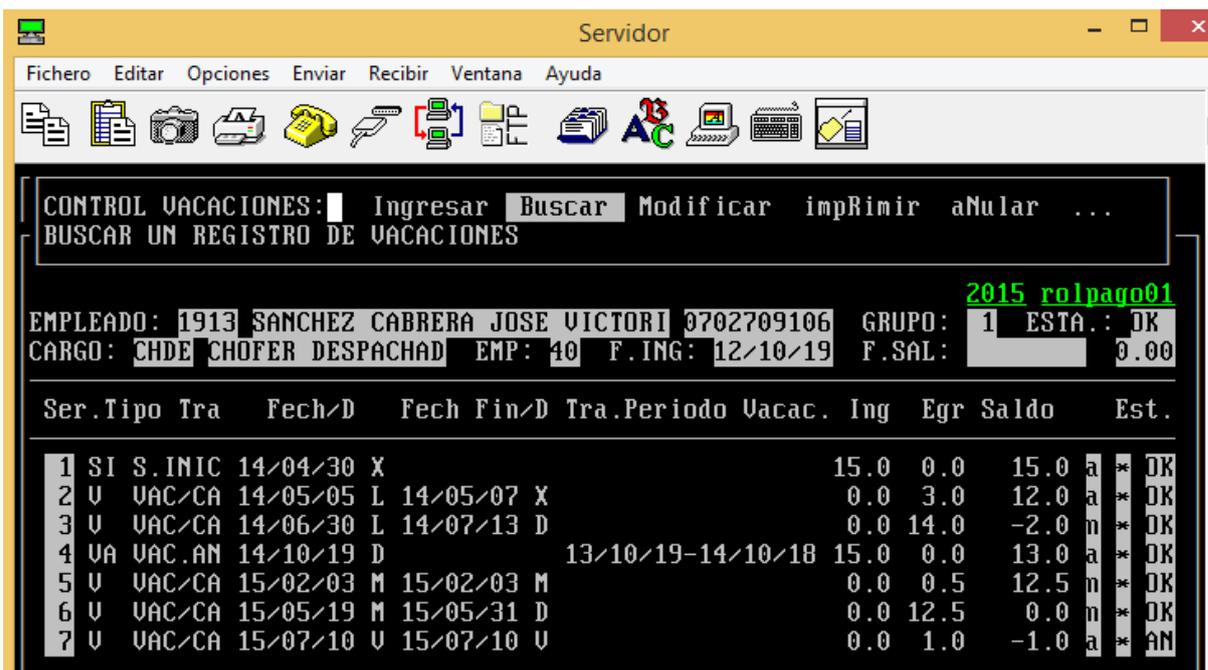


Una vez ingresado a dicho programa, se manejan las siguientes opciones:



Ingresar: Permite registrar una transacción de vacaciones, para un empleado que aún no tiene registros de vacaciones.

Buscar: Permite consultar historial de vacaciones, para empleados que tienen ya registrados transacciones de vacaciones.



Modificar: Agrega registros de vacaciones para un empleado, y adicionalmente permite visualizar los detalles de transacciones anteriores.

Imprimir: Imprime un listado de las transacciones de vacaciones de un empleado, mostrando ingresos, egresos y saldo.

Anular : Permite anular las transacciones de un empleado.

4.1. Opción Ingresar.

Esta opción permite procesar la información que se encuentra previamente aprobada en la solicitud de vacaciones (ver Anexo B); a continuación se presenta las secuencias del proceso de ingreso en las Figuras 54-55-56.

CONTROL VACACIONES: Ingresar Buscar Modificar imprimir aNular ...

ROLPAGO CONTROL VACACIONES PARA EMPLEADOS MEGA PRINCIPAL MATRI
2015 rolpago01

EMPLEADO: 2062 FIGUEROA VILLEGAS CRISTIAN A 5101190871 GRUPO: 10 ESTA.:
CARGO: JTMO VIAS. MOSTRADOR EMP: 02 F.ING: 14/04/01 F.SAL: 0.00

Ser.	Tipo	Tra	Fech/D	Fech Fin/D	Tra.Periodo	Vacac.	Ing	Egr	Saldo	Est.
------	------	-----	--------	------------	-------------	--------	-----	-----	-------	------

<Esc> Acepta <CTRL-C> Cancela

Figura 54. Opción Ingresar
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

El usuario puede digitar el serial del empleado directamente basado en la información de la solicitud, o también puede ubicarse en el siguiente campo de datos y digitar parte o nombre completo del empleado y presionar la tecla de función PF2 para realizar una búsqueda en la base de datos y seleccionar el empleado que solicita vacaciones.

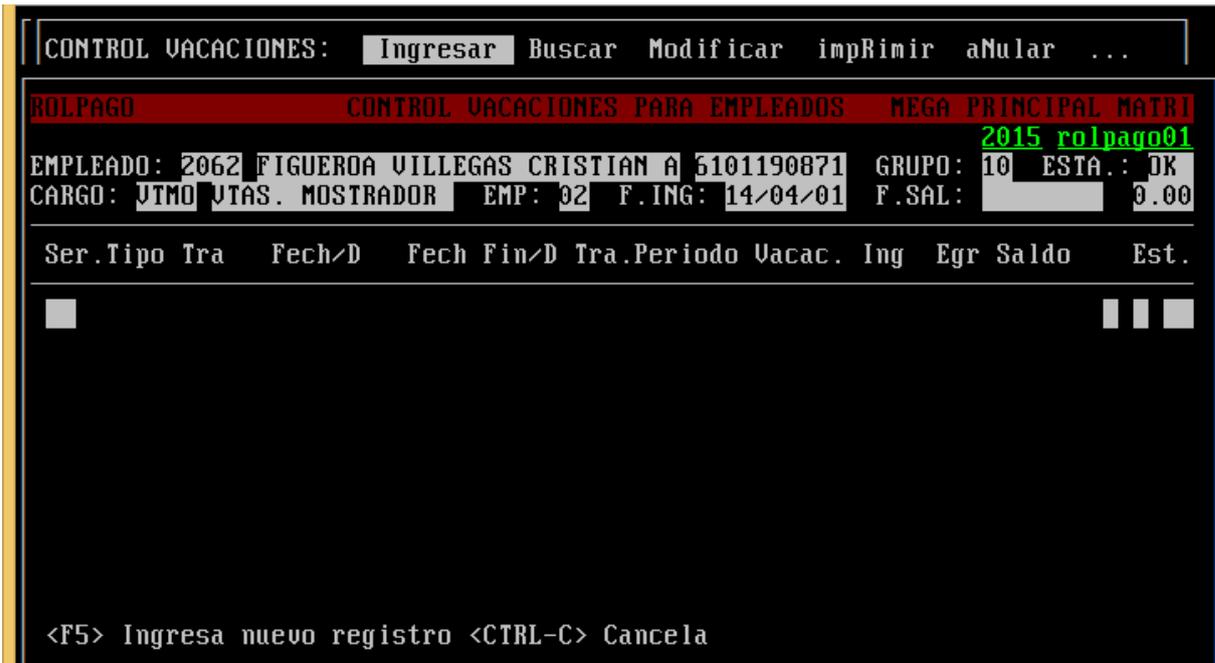


Figura 55. Registro del empleado
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

Una vez seleccionado el serial o el apellido paterno, se cargan los datos del empleado. Al presionar la tecla F5, se pueden agregar transacciones de vacaciones, para lo cual se presentará la siguiente ventana:

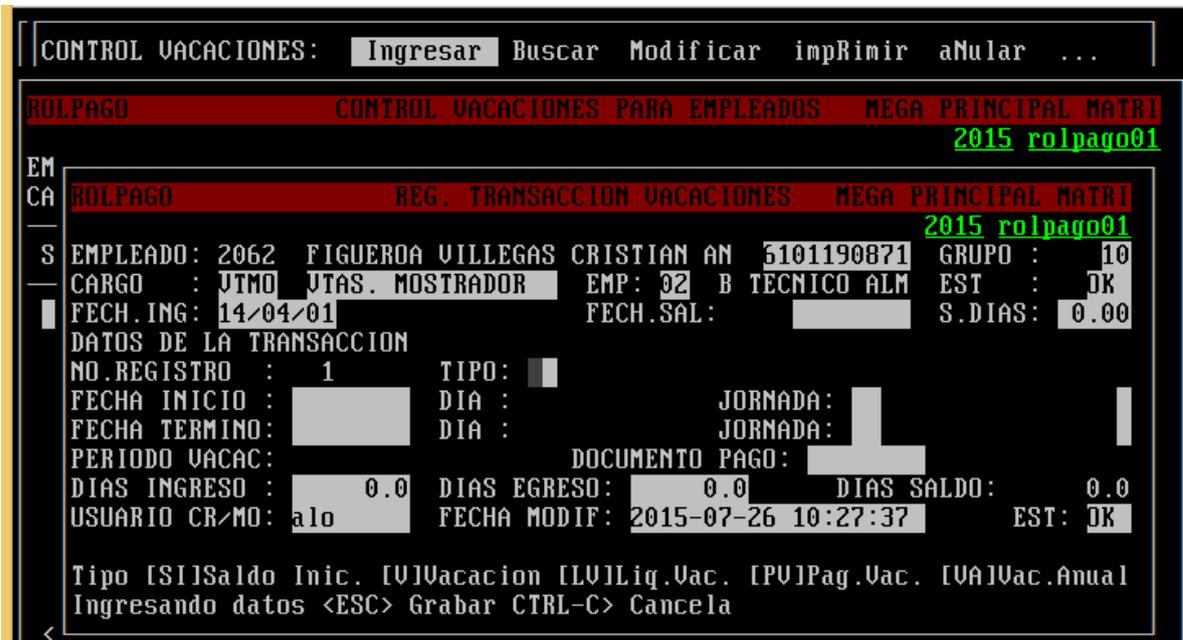


Figura 56 Presentación de la información del empleado seleccionado.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

- En esta ventana se pueden registrar los siguientes tipos de transacciones:
 - SI** : Saldo inicial de ingreso de días
 - VA**: Vacaciones anuales asignadas automáticamente por el sistema (Ingreso).
 - V** : Días de vacaciones o cargo a vacaciones (Egreso)
 - PV**: Pago de vacaciones (Egreso)
 - LV**: Liquidación de días de vacaciones, cuando un empleado se retira de la Empresa. (Egreso)

- **Según el tipo de transacción se deben ingresar los datos correspondientes a:**
 - Fecha Inicio**: Día de registro de transacción o día de inicio de vacaciones
 - Jornada**: En caso de día de vacaciones, para indicar si es completa o de medio día.
 - Tipo de Jornada**: En caso de que la jornada sea de medio día, indicar si es la mañana o la tarde.

- **Datos solo para días o cargo a vacaciones**
 - Fecha Final**: Día de termino de vacaciones
 - Jornada Final**: Indicar si la jornada de vacaciones es completa o de medio día.
 - Tipo de Jornada Final**: En caso de que la jornada sea de medio día, indicar si es la mañana o la tarde.
 - Periodo Vacacional**: Dato informativo, se presenta en las transacciones tipo “VA” para indicar el periodo al cual corresponde las vacaciones anuales.
 - Documento Pago**: Identifica el asiento contable con el cual se realizó el pago de vacaciones, o la liquidación de vacaciones según el caso. Solamente utilizado en las transacciones tipo PV o LV.
 - Días de Ingreso**: Días a ingresar para el empleado, solo se puede modificar este campo en el caso de un registro de saldo inicial SI.
 - Días de Egreso**: Días a egresar para el empleado, este campo se calcula de manera automática para las transacciones tipo V. Se puede modificar únicamente en los registros de pago de vacaciones.
 - Saldo**: Días de saldo después de la transacción generada. Se calcula de manera automática.

Adicionalmente, se presentan los datos para usuario, fecha de modificación y estado, los cuales son para procesos de control y auditoria.

Una vez ingresados todos los datos, con las validaciones correspondientes, se presenta un mensaje que solicita la confirmación del usuario para guardar la transacción y actualizar el saldo del empleado. Una vez confirmada la transacción se presenta en el listado, como muestra la siguiente pantalla (ver Figura 57):

```

CONTROL VACACIONES:  Ingresar  Buscar  Modificar  imprimir  aNular  ...
ROLPAGO                CONTROL VACACIONES PARA EMPLEADOS  MEGA PRINCIPAL MATRI
                                                                2015 rolpago01
EM
CA ROLPAGO                REG. TRANSACCION VACACIONES  MEGA PRINCIPAL MATRI
                                                                2015 rolpago01
S EMPLEADO: 2062  FIGUEROA VILLEGAS CRISTIAN AN  5101190871  GRUPO : 10
CARGO : VTMO  UTAS. MOSTRADOR  EMP: 02  B TECNICO ALM  EST : DK
FECH.ING: 14/04/01  FECH.SAL:  S.DIAS: 0.00
DATOS DE LA TRANSACCION
NO.REGISTRO : 1  TIPO: S1 S.INICIAL
FECHA INICIO : 15/04/01  DIA : X MIERCOLES  JORNADA:
FECHA TERMINO:  DIA :  JORNADA:
PERIODO VACAC:  DOCUMENTO PAGO:
DIAS INGRESO : 15.0  DIAS EGRESO: 0.0  DIAS SALDO: 15.0
USUARIO CR/MO: alo  FECHA MODIF: 2015-07-26 10:27:37  EST: DK

Esta seguro de salir y guardar la nueva transaccion S/N

```

Figura 57. Grabación del registro de ingreso.

Fuente: (Lloret Orellana, 2015).

Elaborado por: Lloret Antonio.

Nota: Se pueden ingresar varias tipos de transacciones desde esta opción.

4.1. Opción Modificar.

Esta opción es muy utilizada para realizar cambios en el registro de vacaciones; para lo cual es necesario seleccionar mediante la opción de búsqueda el nombre de la persona a modificar; luego de lo cual podemos ingresar una nueva novedad presionando la tecla PF5, o también nos permite visualizar una novedad ya registrada con anterioridad mediante la tecla PF6, como se puede apreciar en la Figura 58.

CONTROL VACACIONES: Ingresar Buscar Modificar impRimir aNular ...										
ROLPAGO INGRESE INFORMACION DEL EMPLEADO MEGA PRINCIPAL MATRI										
EMPLEADO: 2062 FIGUEROA VILLEGAS CRISTIAN A 5101190871 GRUPO: 10 ESTA.: DK CARGO: UTMO UTAS. MOSTRADOR EMP: 02 F.ING: 14/04/01 F.SAL: 0.00										
2015 rolpago01										
Ser.	Tipo	Tra	Fech/D	Fech Fin/D	Tra.Periodo	Vacac.	Ing	Egr	Saldo	Est.
1	U	VAC/CA	14/07/10 J	14/07/10 J			0.0	1.0	-1.0	n * DK
2	UA	VAC.AN	15/04/01 X		14/04/01-15/03/31		15.0	0.0	14.0	a * DK
3	U	VAC/CA	15/05/06 X	15/05/19 M			0.0	14.0	0.0	n * DK
<F5> Ingresa nuevo registro <F6> Visualizar Registro <CTRL-C> Cancela										

Figura 58. Modificación de registros
Fuente: (Lloret Orellana, 2015).
Elaborado por: Lloret Antonio.

4.2. Opción Imprimir.

Esta opción nos permite imprimir el registro de vacaciones de la persona previamente seleccionada; la misma que podemos direccionar a través de los medios de salida como pueden ser: Impresora, archivo, o pantalla, en caso de que la salida sea a una impresora podemos seleccionar una impresora que se halle configurada en el sistema mediante la opción de Opciones. Ver Figura 59-60-61.

CONTROL VACACIONES: Ingresar Buscar Modificar impRimir aNular ...										
ROLPAGO INGRESE INFORMACION DEL EMPLEADO MEGA PRINCIPAL MATRI										
EMPLEADO: 2062 FIGUEROA VILLEGAS CRISTIAN A 5101190871 GRUPO: 10 ESTA.: DK CARGO: UTMO UTAS. MOSTRADOR EMP: 02 F.ING: 14/04/01 F.SAL: 0.00										
2015 rolpago01										
Ser.	Tipo	Tra	Fech/D	Fech Fin/D	Tra.Periodo	Vacac.	Ing	Egr	Saldo	Est.
1	U	VAC/CA	14/07/10 J	14/07/10 J			0.0	1.0	-1.0	n * DK
2	UA	VAC.AN	15/04/01 X		14/04/01-15/03/31		15.0	0.0	14.0	a * DK
3	U	VAC/CA	15/05/06 X	15/05/19 M			0.0	14.0	0.0	n * DK
SALIDA: <input type="checkbox"/> Impresora <input type="checkbox"/> Archivo <input checked="" type="checkbox"/> Pantalla <input type="checkbox"/> Opciones Visualiza el reporte en pantalla										

Figura 59. Impresión de registro de vacaciones
Fuente: (Lloret Orellana, 2015)
Elaborado por: Lloret Antonio.

IMP. COM. EL HIERRO CIA. LTDA
C O N T R O L D E V A C A C I O N E S

Fecha Impresion: 15/07/26 10:57:05

Empleado : 2062 - FIGUEROA VILLEGAS CRISTIAN ANDREY
 No. de Cedula : 6101190871
 Empresa y Cargo: 02 - MEGAHIERRO POLITECNI - UTAS. MOSTRADOR
 Grupo : 10 Fecha Ingreso: 14/04/01 Fecha Salida: Estado: OK Saldo Dias:

NO.	TIPO_TRANSAC.	FECH_INI	DIA	TIP JORNADA	FECH_FIN	DIA	TIP JORNADA	DOCUMENT.	D. ING	D. EGR	DIAS SAL
1	U VAC/C.UAC.	14/07/10	J JUEVES	C COMPLETO	14/07/10	J JUEVES	C COMPLETO		.00	1.00	(1.00
2	UA UC.ANUALES	15/04/01			PERIODO VACACIONAL : 14/04/01-15/03/31				15.00	.00	14.00
3	U VAC/C.UAC.	15/05/06	X MIERCOLES	C COMPLETO	15/05/19	M MARTES	C COMPLETO		.00	14.00	0.00

06
 one teclas de cursor para moverse teclas de cursor para moverse
 <C> Abandona, <I> Inicio, <F> Fin, Buscar Cadena, <F3>, <F4> Av. mas

Figura 60. Impresión en papel del registro de vacaciones.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

CONTROL VACACIONES: Ingresar Buscar Modificar **impRimir** aNular ...

ROLPAGO INGRESE INFORMACION DEL EMPLEADO MEGA PRINCIPAL MATRI
 2015 rolpago01

EMPLEADO: 2062 FIGUEROA VILLEGAS CRISTIAN A 5101190871 GRUPO: 10 ESTA.: DK
 CARGO: JIMO UTAS. MOSTRADOR EMP: 02 F.ING: 14/04/01 F.SAL: 0.00

Ser.	Tipo	Tra	Fech/D	Fech Fin/D	Tra.Periodo	Vacac.	Ing	Egr	Saldo	Est.
1	U	VAC/CA	14/07/10	J 14/07/10	J		0.0	1.0	-1.0	n ** DK
2	UA	VAC.AN	15/04/01	X	14/04/01-15/03/31		15.0	0.0	14.0	a ** DK
3	U	VAC/CA	15/05/06	X 15/05/19	M		0.0	14.0	0.0	n ** DK

SALIDA: Impresora Arc
 Opciones para el control

reportes
 romau02
 rrhh
 sistemas

Seleccione la salida y

Figura 61. Selección de impresora.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

4.4. Opción Anular.

Permite anular una transacción de vacaciones y modifica automáticamente el saldo en días del personal.

Las transacciones deben anularse únicamente desde el final. Por favor tomar en cuenta, que todos los datos son validados, tampoco se puede anular un registro que este en estado anulado; también se presentan muchos mensajes de confirmación antes de grabar el registro, por lo que la anulación de transacciones debería ser mínima. Ver Figura 62-63-64.

CONTROL VACACIONES: Ingresar Buscar Modificar imprimir aNular ...

ROLPAGO ANULANDO REGISTRO DE VACAMEGA P INCIPAL MATRI MATRI

2015 rolpago01 rolpago01

EMPLEADO: 1913 SANCHEZ CABRERA JOSE VICTORI 0702709106 GRUPO: 1 ESTA.: OK
 CARGO: CHDE CHOFER DESPACHAD EMP: 40 F.ING: 12/10/19 F.SAL: 0.00

Ser.	Tipo	Tra	Fech/D	Fech Fin/D	Tra.Periodo	Vacac.	Ing	Egr	Saldo	Est.
1	SI	S.INIC	14/04/30	X			15.0	0.0	15.0	a * OK
2	V	VAC/CA	14/05/05	L 14/05/07	X		0.0	3.0	12.0	a * OK
3	V	VAC/CA	14/06/30	L 14/07/13	D		0.0	14.0	-2.0	n * OK
4	VA	VAC.AN	14/10/19	D	13/10/19-14/10/18		15.0	0.0	13.0	a * OK
5	V	VAC/CA	15/02/03	M 15/02/03	M		0.0	0.5	12.5	n * OK
6	V	VAC/CA	15/05/19	M 15/05/31	D		0.0	12.5	0.0	n * OK
7	V	VAC/CA	15/07/10	V 15/07/10	V		0.0	1.0	-1.0	a * AN

Anulacion: Ubiquese en la ult.linea activa y presione F7

Figura 62. Opción Anulacion.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

CONTROL VACACIONES: Ingresar Buscar Modificar imprimir aNular ...

ANULAR TRANSACCION VACACIONES

2015 rolpago01 rolpago01

EMPLEADO: 1913 SANCHEZ CABRERA JOSE VICTORI 0702709106 GRUPO: 1 ESTA.: OK
 CARGO: CHDE CHOFER DESPACHAD EMP: 40 F.ING: 12/10/19 F.SAL: 0.00

Ser.	Tipo	Tra	Fech/D	Fech Fin/D	Tra.Periodo	Vacac.	Ing	Egr	Saldo	Est.
1	SI	S.INIC	14/04/30	X			15.0	0.0	15.0	a * OK
2	V	VAC/CA	14/05/05	L 14/05/07	X		0.0	3.0	12.0	a * OK
3	V	VAC/CA	14/06/30	L 14/07/13	D		0.0	14.0	-2.0	n * OK
4	VA	VAC.AN	14/10/19	D	13/10/19-14/10/18		15.0	0.0	13.0	a * OK
5	V	VAC/CA	15/02/03	M 15/02/03	M		0.0	0.5	12.5	n * OK
6	V	VAC/CA	15/05/19	M 15/05/31	D		0.0	12.5	0.0	n * OK
7	V	VAC/CA	15/07/10	V 15/07/10	V		0.0	1.0	-1.0	a * AN

Error.- No puede anular un registro que no esta en estado OK

Figura 63. Mensaje de error.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

CONTROL VACACIONES: Ingresar Buscar Modificar imprimir aNular ...

ROLPAGO ANULANDO REGISTRO DE VACAMEGA P INCIPAL MATRI MATRI
2015 rolpago01 rolpago01

EMPLEADO: 1913 SANCHEZ CABRERA JOSE VICTORI 0702709106 GRUPO: 1 ESTA.: OK
 CARGO: CHDE CHOFER DESPACHAD EMP: 40 F.ING: 12/10/19 F.SAL: 0.00

Ser.	Tipo	Tra	Fech/D	Fech Fin/D	Tra.Periodo	Vacac.	Ing	Egr	Saldo	Est.
1	SI	S.INIC	14/04/30	X			15.0	0.0	15.0	a**OK
2	U	VAC/CA	14/05/05	L 14/05/07	X		0.0	3.0	12.0	a**OK
3	U	VAC/CA	14/06/30	L 14/07/13	D		0.0	14.0	-2.0	n**OK
4	VA	VAC.AN	14/10/19	D	13/10/19-14/10/18		15.0	0.0	13.0	a**OK
5	U	VAC/CA	15/02/03	M 15/02/03	M		0.0	0.5	12.5	n**OK
6	U	VAC/CA	15/05/19	M 15/05/31	D		0.0	12.5	0.0	n**OK
7	U	VAC/CA	15/07/10	U 15/07/10	U		0.0	1.0	-1.0	a**AN

Esta seguro de anular el registro en la linea 6 S/N? █

Figura 64. Proceso de anulación de registros de vacaciones.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

4.5. Consideraciones adicionales.

Adicional, a las opciones ya indicadas, se generarán automáticamente los registros **VA** y **LV**. Las transacciones “**VA**” aplican, cuando el empleado cumpla un año de trabajo, y serán incluidos los días extras a los cuales tienen derecho una vez pasados los 5 años. También se generará esta transacción, cuando se registre la salida de un empleado (ver Figura 65), los días asignados estarán en proporción del tiempo trabajado.

CONTROL VACACIONES: Ingresar Buscar Modificar imprimir aNular ...

ROLPAGO INGRESE INFORMACION DEL EMPLEADO MEGA PRINCIPAL MATRI
2015 rolpago01

EM
 CA **ROLPAGO REG. TRANSACCION VACACIONES MEGA PRINCIPAL MATRI**
2015 rolpago01

S EMPLEADO: 1392 CABRERA JUCA WILLIAN PAUL 0103846622 GRUPO: 1
 CARGO: ASDC ASIST. DE COMPRAS EMP: 01 A PRINCIPAL A EST: ANU
 FECH.ING: 08/11/12 FECH.SAL: 15/01/13 S.DIAS: 0.00

DATOS DE LA TRANSACCION

NO.REGISTRO: 8 TIPO: JA VAC.ANU.
 FECHA INICIO: 14/11/12 DIA: X MIERCOLES JORNADA: █
 FECHA TERMINO: █ DIA: █ JORNADA: █
 PERIODO VACAC: 13/11/12-14/11/11 DOCUMENTO PAGO: █
 DIAS INGRESO: 16.0 DIAS EGRESO: 0.0 DIAS SALDO: 38.0
 USUARIO CR/MO: autc FECHA MODIF: 2014-12-09 07:55:39 EST: OK

Ingresando datos <ESC> Grabar CTRL-C> Cancela

Figura 65. Documento de salida de empleado.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.

Al momento de registrar la salida, aparecerá un mensaje que indique se va generar dicho registro y pedirá la confirmación.

De la misma manera, el registro “LV”, permitirá dar de baja con la liquidación de haberes (Documento Contable LH ver Figura 66) los días que se encuentran acumulados para el empleado. Al momento de mayorizar el documento contable, se solicitará el número de serial del empleado al cual aplica la liquidación, y se pedirá la confirmación para grabar el registro con los datos a liquidar.

CONTABILIDAD		CONSULTANDO DOCUMENTO		EL HIERRO	
Fecha(aa/mm/dd): 15/02/06			Comprobante: LH000004		
			2015 conta15 d		
Lin.	Codigo cuenta	Descripcion	Debito	Credito	
1	2 1 1 07 01 000	DECIMO TERCER SUELDO	50.09		
2	2 1 1 07 02 001	DEC.CUARTO SUELDO SIERRA	160.28		
3	2 1 1 07 03 000	VACACIONES	225.20		
4	2 1 1 07 03 000	VACACIONES	49.49		
5	2 2 4 02 00 000	DESAHUCIO	654.65		
6	1 1 2 03 06 062	VIATICOS CABRERA JUCA WIL		2.40	
7	2 1 1 08 02 000	PRESTAMOS I.E.S.S.		202.95	
8	1 1 1 02 03 000	BANCO BOLIVARIANO CTA. 40		934.36	
TOTALES:			1,139.71	1,139.71	
CONCEPTO					
CABRERA JUCA WILLIAM PAUL, PAGO EN LIQUIDACION DE HABERES					
PERIODO: DIC/14-ENE/15.					
<ESC> Regresa ,F5=Imp. ,F6=Gastos ,F7=Reversa ,F8=Dat.Ret ,F9=Serv ,F10=copia					

Figura 66. Documento contable de liquidación de haberes.
 Fuente: (Lloret Orellana, 2015).
 Elaborado por: Lloret Antonio.