

UNIVERSIDAD DE ORIENTE  
NÚCLEO DE ANZOÁTEGUI  
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS  
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



“DISEÑO DE UN SISTEMA DE INFORMACIÓN PARA EL CONTROL,  
EVALUACIÓN Y ESTIMACIÓN DE LAS HORAS HOMBRE INVERTIDAS  
EN EL PROCESO DE DESARROLLO DE SOFTWARE”

Realizado por:

---

Pino Villarroel, Carlos Julio  
C.I. V-17.762.974

Trabajo de grado presentado como requisito parcial para optar al título de  
INGENIERO EN SISTEMAS  
BARCELONA, Enero de 2009.

UNIVERSIDAD DE ORIENTE  
NÚCLEO DE ANZOÁTEGUI

ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS  
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



---

“DISEÑO DE UN SISTEMA DE INFORMACIÓN PARA EL CONTROL,  
EVALUACIÓN Y ESTIMACIÓN DE LAS HORAS HOMBRE INVERTIDAS  
EN EL PROCESO DE DESARROLLO DE SOFTWARE”

Asesor:

---

Lic. Manuel Carrasquero  
Asesor Académico

BARCELONA, Enero de 2009.

UNIVERSIDAD DE ORIENTE  
NÚCLEO DE ANZOÁTEGUI  
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS  
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



---

“DISEÑO DE UN SISTEMA DE INFORMACIÓN PARA EL CONTROL,  
EVALUACIÓN Y ESTIMACIÓN DE LAS HORAS HOMBRE INVERTIDAS  
EN EL PROCESO DE DESARROLLO DE SOFTWARE”

Jurado Calificador

---

Lic. Manuel Carrasquero  
Asesor Académico

---

Prof. Claudio A. Cortínez N.  
Jurado Principal

Prof. Aquiles Torrealba  
Jurado Principal

BARCELONA, Enero de 2009.

## **RESOLUCIÓN**

De acuerdo con el artículo 44 del reglamento de trabajo de grado:

“Los trabajos de grado son de exclusiva propiedad de la Universidad de Oriente y sólo podrán ser utilizados a otros fines con el consentimiento del consejo de núcleo respectivo, quien lo participará al Consejo Universitario”.

## **DEDICATORIA**

Al ser humano más fuerte y amable que conozco, mí madre, de manera única e inamovible. Solo por ella existe este trabajo, y la persona que lo realizó; no solo me trajo al mundo, también construyó gran parte de lo que soy, la parte más importante.

Carlos Julio Pino.

## AGRADECIMIENTOS

Admito que muchas veces al imaginarme escribiendo estas palabras, pensé que no tendría nadie a quien agradecer, más que a mis padres y familia, sin cuyo obvio aporte de todos los tipos no hubiera sido nada de esto posible, al menos se hubiese complicado de sobremanera. Sin embargo aquí me encuentro, nuevamente equivocado, pensando en tantos nombres que trataré en gran medida de resumirlos en estas páginas, con las disculpas de algunos de ellos, por supuesto.

Primero, como ya dije, son mi madre, mi padre y mis hermanos los que más merecen mi agradecimiento. Todos, cada uno a su manera, me ayudaron en incontables ocasiones a minimizar la carga que representa ser “*solo un estudiante*” en este país. Me dieron los consejos que pensaron necesitaba, me regañaron de vez en cuando y mantuvieron siempre la puerta abierta para cuando necesitara un descanso. Estoy seguro así continuará siendo, ante todo, me enorgullece enorgullecerlos, sin importar redundancias.

De manera puntual, este trabajo no hubiese sido posible en su estado actual sin la invaluable ayuda de Linda Inciarte, diseñadora gráfica de profesión y excelente persona por pasatiempo. Su aporte dio el toque que admito faltaba dentro de toda la ciencia que intenté esparcir en estas páginas. Un excelente aporte. Igualmente, agradezco a Carlos Vallenilla y a Ángel Sánchez por su invaluable colaboración, sus ideas los llevarán lejos, y espero que el manicomio que los busca no los encuentre ningún día cercano.

También me gustaría mencionar con especial cuidado a Rosmelis Machado, aunque ella misma no lo creería si leyese estas palabras, por mantenerme cuerdo durante los momentos que llamaban a la locura. Gracias Rosmelis, nunca había una mujer hecho tanto con tan poco como tú lo hiciste. Espero ser al menos la mitad del hombre que mereces que

sea, pues con eso será suficiente para cambiar el mundo mil veces, para bien, ya me conoces.

¡Mary! ¡Mi terapeuta! Si los psiquiatras fueran tan eficaces como lo eres tú, sería la profesión más lucrativa. Gracias por todo lo que has soportado sin tener por qué. Eres una gran persona y tengo suerte de conocerte.

A todos los que me brindaron su compañía, gracias por intentar incluirme, por evitar excluirme, por reírse de mis bromas y abrir sus mentes a mis discursos. Espero haber hecho una diferencia para mejor en sus vidas, y les deseo la mejor de las suertes.

Finalmente, me gustaría agradecer a dos mujeres. Las separa un mundo y, no obstante, ambas dijeron en su momento que me amaban. Gracias por ese sentimiento que no luché lo suficiente para merecer. Este trabajo y todo lo que haga de aquí en adelante también es por ustedes. Sé que soy un soñador, pero uno consciente de que los sueños se construyen con manos, uñas y dientes. No las defraudaré, sería defraudarme a mí mismo.

Me gustaría poder extenderme un poco con cada persona, e incluir algunas más pero...

¿Quién lee esto?...

Carlos Julio Pino.

## ÍNDICE

RESOLUCIÓN .....	IV
DEDICATORIA .....	V
AGRADECIMIENTOS.....	VI
ÍNDICE.....	VIII
ÍNDICE DE TABLAS .....	XII
ÍNDICE DE FIGURAS.....	XIV
RESUMEN .....	XVIII
CAPÍTULO 1 .....	20
1.1. PLANTEAMIENTO DEL PROBLEMA .....	20
1.2. OBJETIVOS .....	25
1.2.1. Objetivo General.....	25
1.2.2. Objetivos Específicos.....	25
CAPÍTULO 2.....	26
2.1. ANTECEDENTES .....	26
2.2. DEFINICIÓN DE SISTEMA.....	28
2.3. SISTEMA DE INFORMACIÓN.....	28
2.4. OBJETIVOS DE LOS SISTEMAS DE INFORMACIÓN.....	29
2.5. TIPOS DE SISTEMAS DE INFORMACIÓN.....	29
2.6. BASE DE DATOS.....	30
2.6.1. Componentes Principales de las Bases de Datos .....	31



2.6.2. Clasificación de las Bases de Datos .....	31
2.6.3. Modelo de datos .....	32
2.7. SISTEMA DE GESTIÓN DE BASE DE DATOS (DBMS).....	33
2.7.1. Arquitectura de un SGBD .....	34
2.7.2. Clasificación de los Sistemas de Gestión de Bases de Datos	35
2.8. INGENIERÍA DE SOFTWARE. ....	36
2.8.1. Proceso Unificado de Desarrollo de Software .....	37
2.8.2. Fases del Proceso Unificado .....	37
2.9. LENGUAJE DE MODELADO UNIFICADO (UML). ....	39
2.10. MÉTRICAS O INDICADORES DE SOFTWARE .....	43
2.10.1. Clasificación Básica de las Métricas de Software .....	43
2.11. MODELO DE ESTIMACIÓN DE COSTOS (APLICADO AL PROCESO DE DESARROLLO DE SOFTWARE).....	45
2.12. MODELO DE ESTIMACIÓN EMPÍRICA (APLICADO AL PROCESO DE DESARROLLO DE SOFTWARE).....	45
2.13. PROGRAMACIÓN ORIENTADA A OBJETOS.....	47
2.13.1. Principios de los lenguajes orientados a objetos .....	47
2.13.2. Encapsulación .....	47
2.13.3. Herencia .....	48
2.13.4. Polimorfismo .....	48
CAPÍTULO 3.....	49

3.1. INTRODUCCIÓN .....	49
3.2. ESTRUCTURA ORGANIZATIVA DEL SISTEMA OBJETO DE ESTUDIO .....	49
3.2.1. Misión .....	51
3.2.2. Visión .....	51
3.2.3. Filosofía de Servicio de la Empresa.....	51
3.2.4. Descripción y Funciones Principales .....	52
3.3. ANÁLISIS DEL SISTEMA ACTUAL .....	55
3.4. ANÁLISIS DE LA PROBLEMÁTICA.....	58
CAPÍTULO 4.....	60
4.1. INTRODUCCIÓN .....	60
4.2. ANÁLISIS DE LOS REQUERIMIENTOS DEL SISTEMA.....	60
4.2.1. Requerimientos Funcionales .....	61
4.2.2. Especificación de los Requerimientos .....	62
CAPÍTULO 5.....	77
5.1. INTRODUCCIÓN .....	77
5.2. DISEÑO DE LA ESTRUCTURA DEL SOFTWARE.....	77
5.2.1. Diagrama de Paquetes .....	78
5.3. DISEÑO DE LA BASE DE DATOS .....	82
5.3.1. Estructura de Dominio del Sistema.....	83
5.3.2. Diseño del Modelo Relacional de la Base de Datos .....	88

5.4. DISEÑO DE LA INTERFAZ DE USUARIO.....	98
5.4.1. Pantalla de Presentación del Sistema .....	99
5.4.2. Pantalla de Bienvenida .....	100
5.4.3. Módulo de Seguridad.....	100
5.4.4. Módulo de Configuración .....	102
5.4.5. Módulo de Gestión de Proyectos .....	106
5.4.6. Módulo de Control de Actividades .....	111
5.4.7. Módulo de Control de Tareas Personales.....	119
5.4.8. Módulo de Reportes.....	121
CONCLUSIONES .....	125
RECOMENDACIONES.....	126
BIBLIOGRAFIA.....	128

## ÍNDICE DE TABLAS

Tabla 5.1. Dat_Usuario (Datos de usuarios).....	88
Tabla 5.2. TipoUsu (Datos de tipos de usuarios).....	89
Tabla 5.3. Log_Usuario (Datos de validación de cuentas de usuarios) ...	89
Tabla 5.4. Costo_HH (Historial de costos asociados a empleado) .....	89
Tabla 5.5. Dat_Estatus (Datos de estatus posibles para tareas) .....	90
Tabla 5.6. Dat_Mod (Datos de módulos del sistema) .....	90
Tabla 5.7. Dat_Apli (Datos de aplicaciones del sistema) .....	90
Tabla 5.8. Apli_Mod (Estructura de aplicaciones por módulo) .....	91
Tabla 5.9. Acces_Apli (Relación entre usuarios y aplicaciones permitidas) .....	91
Tabla 5.10. Dat_Proj (Datos de proyectos).....	92
Tabla 5.11. Dat_Fase (Datos de fases) .....	93
Tabla 5.12. Dat_Activ (Datos de actividades) .....	93
Tabla 5.13. Dat_Tarea (Datos de tareas) .....	94
Tabla 5.14. Dat_Registro (Datos de registros).....	94
Tabla 5.15. Tarea_Asig (Tareas que han sido asignadas a un usuario)..	95
Tabla 5.16. Reg_Usuario (Registros hechos por usuarios) .....	95
Tabla 5.17. Fase_Proj (Fases declaradas dentro de un proyecto específico) .....	96
Tabla 5.18. Activ_Fase (Actividades declaradas dentro de una fase específica) .....	96

Tabla 5.19. Tarea_Activ (Tareas declaradas dentro de una actividad específica) .....	97
Tabla 5.20. Macro Manejadores de Dificultad (Parte 1).....	107
Tabla 5.21. Macro Manejadores de Dificultad (Parte 2).....	107
Tabla 5.22. Matriz de Cálculo de Dificultad por Macro Manejadores .....	108
Tabla 5.23. Micro Manejadores de Dificultad .....	114
Tabla 5.24. Determinante del Grado de Dificultad .....	116
Tabla 5.25. Criterio de Cálculo de los Puntos de Productividad .....	118

## ÍNDICE DE FIGURAS

Figura 3.1. Organigrama Estructural de <i>Empresas Saetha</i> .....	50
Figura 3.2. Estructura Organizacional de <i>Saetha Design &amp; Systems</i> . ....	55
Figura 4.1. Diagrama de Casos de Uso del Sistema .....	64
Figura 4.2. Diagrama de Clases de Análisis del caso “ <i>Gestionar Proyectos</i> ” .....	71
Figura 4.3. Diagrama de Clases de Análisis del caso “ <i>Gestionar Actividades</i> ” .....	71
Figura 4.4. Diagrama de Clases de Análisis del caso “ <i>Configurar Sistema</i> ” .....	<b>¡Error! Marcador no definido.</b>
Figura 4.5. Diagrama de Clases de Análisis del caso “ <i>Visualizar Reportes</i> ” .....	<b>¡Error! Marcador no definido.</b>
Figura 5.1. Diagrama de Paquetes SISTEMA .....	79
Figura 5.2. Diagrama de Clases de Diseño para los paquetes SISTEMA.Seguridad y SISTEMA.Estructura .....	80
Figura 5.3. Diagrama de Clases de Diseño para los paquetes SISTEMA.Configuración y SISTEMA.Servicios .....	81
Figura 5.4. Diagrama de Clases de Diseño para el paquete SISTEMA.Interfaz .....	82
Figura 5.5. Leyenda de Estructura de Dominio del Sistema .....	84
Figura 5.6. Estructura del Dominio del Usuario	<b>¡Error! Marcador no definido.</b>
Figura 5.7. Estructura del Dominio del Proyecto	<b>¡Error! Marcador no definido.</b>

Figura 5.8. Estructura del Dominio de la Fase;¡Error!	<b>Marcador</b>	<b>no definido.</b>
Figura 5.9. Estructura del Dominio de la Actividad;¡Error!	<b>Marcador</b>	<b>no definido.</b>
Figura 5.10. Estructura del Dominio de la Tarea;¡Error!	<b>Marcador</b>	<b>no definido.</b>
Figura 5.11. Estructura del Dominio del Registro;¡Error!	<b>Marcador</b>	<b>no definido.</b>
Figura 5.12. Estructura del Dominio del Estatus;¡Error!	<b>Marcador</b>	<b>no definido.</b>
Figura 5.13. Estructura del Dominio de la Aplicación;¡Error!	<b>Marcador</b>	<b>no definido.</b>
Figura 5.14. Estructura del Dominio del Módulo;¡Error!	<b>Marcador</b>	<b>no definido.</b>
Figura 5.15. Estructura del Dominio del Sistema;¡Error!	<b>Marcador</b>	<b>no definido.</b>
Figura 5.16. Modelo Relacional de la Base de Datos del Sistema.... ¡Error!	<b>Marcador no definido.</b>	
Figura 5.17. Pantalla de Presentación y Acceso al Sistema.....		100
Figura 5.18. Pantalla de Bienvenida .....		100
Figura 5.19. Cambio de Contraseña .....		101
Figura 5.20. Control de Usuarios .....		101
Figura 5.21. Registro de Nuevo Usuario .....		102
Figura 5.22. Fases de Proyecto .....	<b>¡Error! Marcador no definido.</b>	

Figura 5.23. Modificación de Porcentajes Estándar de Fases .....	¡Error!
<b>Marcador no definido.</b>	
Figura 5.24. Registro de Nueva Fase Estándar	¡Error! Marcador no definido.
Figura 5.25. Actividades de Fase.....	¡Error! Marcador no definido.
Figura 5.26. Registro de Nueva Actividad Estándar	¡Error! Marcador no definido.
Figura 5.27. Tareas de Fase.....	105
Figura 5.28. Gestión de Costo de Hora Hombre.....	105
Figura 5.29. Historial de Costos de “Empleado”	¡Error! Marcador no definido.
Figura 5.30. Registro de Nuevo Costo de Hora Hombre	¡Error! Marcador no definido.
Figura 5.31. Listado de Proyectos .....	¡Error! Marcador no definido.
Figura 5.32. Planificación de un Nuevo Proyecto	¡Error! Marcador no definido.
Figura 5.33. Seguimiento de Proyectos .....	¡Error! Marcador no definido.
Figura 5.34. Lista de Estimaciones .....	¡Error! Marcador no definido.
Figura 5.35. Nueva Estimación .....	¡Error! Marcador no definido.
Figura 5.36. Lista de Proyectos en Proceso	¡Error! Marcador no definido.
Figura 5.37. Lista de Fases del Proyecto.....	112
Figura 5.38. Lista de Actividades de Fase .....	113
Figura 5.39. Lista de Tareas de Actividad...	¡Error! Marcador no definido.



Figura 5.40. Registro de Nueva Tarea.....	¡Error! Marcador no definido.
Figura 5.41. Evaluación del Desempeño ....	¡Error! Marcador no definido.
Figura 5.42. Evaluación del Desempeño por Empleado	¡Error! Marcador no definido.
Figura 5.43. Historial de Trabajo de Tarea .	¡Error! Marcador no definido.
Figura 5.44. Lista de Tareas .....	¡Error! Marcador no definido.
Figura 5.45. Nuevo Registro de Trabajo Realizado	¡Error! Marcador no definido.
Figura 5.46. Nuevo Registro de Trabajo Realizado .....	121
Figura 5.47. Reporte de Proyectos .....	122
Figura 5.48. Reporte de Carga Laboral por Empleado .....	122
Figura 5.49. Reporte Detallado de Proyectos .....	123
Figura 5.50. Reporte de Actividades de Producción .....	124

## **RESUMEN**

Durante esta investigación se diseñó un sistema Web para una empresa desarrolladora de Software; este permite llevar un control completo de las actividades de producción. A través de él se registran todos los proyectos activos de la empresa, se subdividen en fases, estas a su vez en actividades y estas últimas en tareas asignables a equipos de trabajo donde cada miembro es poseedor de una interfaz propia para el registro de sus actividades diarias, las cuales actualizan el estatus total del proyecto en tiempo real. Adicionalmente, la información de control recopilada es usada para estimar costos en el futuro de forma confiable, tomando en cuenta dificultad en los procesos y experiencia de los participantes. Para realizar la labor de diseño se conceptualizaron nuevas metodologías de diagramación de entidades y cálculo de costos, así como un indicador de gestión apropiado y veraz orientado a la evaluación justa del trabajo.

# CAPÍTULO 1

## 1.1. PLANTEAMIENTO DEL PROBLEMA

*Saetha D&S*, registrada en el año 2004 bajo el nombre de *Saetha Design & Systems C.A.*, es una empresa dedicada al desarrollo integral de sistemas de información, soluciones Web y comunicación gráfica. Está enfocada en la creación de productos y servicios innovadores con el fin de buscar nuevas formas de integración de los recursos tecnológicos y artísticos con el procesamiento de información para lograr arquitecturas comunicativas, de producción y evaluación altamente competitivas. Tiene su sede principal en la Torre BVC, localizada en el sector Las Garzas de la avenida Intercomunal, en Lechería, estado Anzoátegui.

Desde su creación la empresa ha perseguido firmemente el objetivo de crear planes de acción que le permitan maximizar los recursos de sus clientes, así como generar nuevas oportunidades de negocios mediante la implementación de soluciones creativas adaptadas a las necesidades de los mismos. Es precisamente esta flexibilidad en la adecuación de sus productos y servicios la estrategia que se ha planteado la organización para mantener siempre altos estándares de satisfacción y expandir sus horizontes de mercado.

*Saetha D&S* ofrece a sus clientes los servicios listados a continuación:

- Desarrollo de Software (sistemas de procesamiento de transacciones, sistemas de información administrativa, sistemas de soporte a la toma de decisiones, sistemas expertos, sistemas estratégicos).
- Desarrollo de portales de Colaboración.

- Desarrollo integral de Websites (Conceptualización y diseño de imagen, estructuración de contenido, programación de aplicaciones Web, hospedaje y registro de dominios).
- Desarrollo de imagen Visual y multimedia (Diseño gráfico general, presentaciones multimedia con integración audiovisual, banners publicitarios, animaciones en *Adobe Flash Y After Effect*).
- Capacitación y soporte técnico.

En la actualidad, la empresa se encuentra en una etapa de crecimiento crítica para la consolidación de la marca dentro de este mercado siempre cambiante y altamente competitivo; enfrenta, como es común en estos casos, el problema de la *necesidad de adaptación*. Muchas compañías, tanto en las Tecnologías de la Información (TI) como en otras ramas del conocimiento, perecen en esta etapa de su evolución debido en buena parte a las grandes dificultades que presenta un aumento constante en el volumen de proyectos en ejecución. Empresas de mayor envergadura deben enfrentarse consecuentemente a elevadas cargas de trabajo, estrictas necesidades de control interno y, por supuesto, considerablemente mayores costos de producción. De ser la tasa de crecimiento de estas necesidades mayor a la capacidad interna de adaptación a los nuevos requisitos, la organización termina consumiéndose en sí misma, desapareciendo víctima de la inadecuación de su estructura de trabajo.

A nivel de TI, esta característica empresarial se hace presente de una manera más contundente al considerar la tremenda velocidad con la que se presentan nuevas tecnologías, métodos y estructuras que mantienen fresco el desafío de innovar siempre los esquemas previamente establecidos. Ejecutar eficientemente esta renovación metodológica no es posible si no existe para ello una plataforma de control y evaluación eficiente, que permita a la organización saber en todo

momento dónde se encuentra y, sobretodo, hacia dónde puede ir. Entendiendo e integrando las mejoras que existen y aquellas que son necesarias y factibles para la empresa, se obtiene un plan de acción que realmente se puede considerar no solo para mantener un estatus competitivo dentro del mercado, sino inclusive obtener ventajas que permitan superar a empresas con mayor experiencia y alcance.

Esta comprensión cabal de su devenir y posible porvenir solo puede darse si la directiva encargada tiene al alcance la información relevante a tal fin de una manera oportuna y eficiente, extraída automáticamente de los procesos operativos normales de la empresa, evitando las sobrecargas en el flujo de trabajo que tal movilización pudiera causar.

Es en este último apartado en el que *Saetha D&S* falla al no tener en funcionamiento ningún tipo de sistema que permita evaluar efectivamente el desempeño interno, más allá del control de calidad ejecutado al resultado de cada proyecto individualmente. Este esquema se basa en niveles de calidad aceptables en función de los tiempos de ejecución de los proyectos. Se desempeña con cierta exactitud, adquirida sin duda con la experiencia obtenida por la directiva a partir de casos de éxito previos, pero resulta inútil cuando se desea percibir información más detallada sobre el rol que toma cada empleado en la realización de cada uno de ellos. Se hacen aún más evidentes sus carencias cuando la naturaleza del proyecto es nueva para la empresa, y no existen criterios formados con respecto al consumo de recursos que pueda acarrear su ejecución.

El propósito de este proyecto será el diseño de un sistema de información que cubra las insuficiencias relacionadas con ese tan

necesario control del trabajo, integrando a todo el personal del área de producción de la empresa en una red de trabajo que mantenga cada actividad relacionada con los proyectos bajo control, evaluando los tiempos, costos y recursos humanos asociados a cada una de ellas en función de estándares establecidos. Adicionalmente, el sistema proveerá a la empresa de una ventaja competitiva considerable a través de un modelo de estimaciones que caracterizará los proyectos próximos en función de los pasados, comparando factores como el tipo de proyecto y su complejidad para obtener información de gran valor para la planificación y gerencia administrativa. Es en esta última característica que radica la originalidad que plantea la realización de este proyecto, ya que no ha sido desarrollado en el núcleo Anzoátegui de la Universidad de Oriente sistema alguno que aplique un modelo de estimación de costos a los futuros trabajos de una organización. De hecho, incluso a nivel mundial, no está muy arraigada en el ámbito empresarial la idea de utilizar los llamados Modelos de Estimación de Costos, ni siquiera en sus niveles más básicos, para tomar decisiones de importancia que afecten las actividades venideras de una organización. Sin embargo, esta funcionalidad será cabalmente desarrollada dentro del alcance de esta tesis de pregrado, el cual queda delimitado por las etapas de inicio y elaboración del proceso unificado de desarrollo de software, siendo éste luego continuado para una posterior implementación por la Gerencia de Desarrollo de la empresa.

En conjunto, el sistema estará diseñado para permitir conocer los costos ocultos causados por la ineficiencia de los procesos, medir la efectividad de los empleados y grupos de trabajo, proyectar y pronosticar el efecto de los cambios en la estructura de desarrollo de los proyectos, identificar áreas en las cuales pueden implementarse mejoras en la cadena de producción, verificar la evolución histórica tanto de la empresa como de sus empleados, evaluar las modificaciones realizadas para

incrementar la productividad, así como justificarlas, entre muchas otras ventajas estratégicas. Su gran importancia queda de manifiesto al encontrarse la empresa, como ya se ha mencionado, en una etapa de crecimiento acelerado, en la cual estos factores cobran un carácter de urgencia que no se puede ignorar, y requiere la implementación de esta solución personalizada a su modelo de trabajo para asegurar su continuidad.

## **1.2. OBJETIVOS**

### **1.2.1. Objetivo General**

Diseñar un sistema de información para el control, evaluación y estimación de las Horas Hombre consumidas en un proyecto de desarrollo de software.

### **1.2.2. Objetivos Específicos**

1. Describir el sistema de estudio
2. Identificar los requerimientos del sistema a desarrollar
3. Diseñar la estructura del Software
4. Diseñar la base de datos del sistema
5. Diseñar la interfaz de usuario del sistema



## CAPÍTULO 2

### 2.1. ANTECEDENTES

Dentro de la empresa objeto de estudio se han realizado en múltiples ocasiones sistemas de información enfocados a ejercer cierto tipo de control sobre las actividades realizadas en las empresas cliente, sin embargo, ninguno ha sido desarrollado para sus propios procesos. Más aún, la empresa no se ha visto involucrada en desarrollos que permitan evaluar la efectividad de los empleados o grupos de trabajo con respecto a métricas establecidas, ni estimar costos basados en estadísticas de uso previo. En la Universidad de Oriente existe una cantidad considerable de trabajos relacionados con el desarrollo de sistemas de información y, aunque ninguno de ellos ataca estos temas con precisión, sirven para sentar un precedente metodológico para este tipo de proyectos.

Rodríguez, M. (2004) desarrolló un trabajo de grado titulado **“Diseño de un sistema de información para el control de los datos generales de los yacimientos oficiales de los campos petroleros de PDVSA Oriente”**. Este sistema en particular atacaba la problemática de la falta de coordinación que existía entre los diferentes departamentos asociados a la Gerencia de Tecnología de la empresa petrolera, causada por su compleja estructura organizativa no soportada por la infraestructura apropiada. Particularmente, se producían inconvenientes en el trámite de los denominados “datos generales de exploración y producción” en el Departamento de Calidad de Datos. La entonces bachiller resolvió el problema diseñando un sistema capaz de llevar un control sobre todas las transacciones de información ejecutadas en ese departamento, registrando los accesos de cada empleado a la información y las modificaciones que ellos hiciesen. [1]

Domínguez, N. (2005) desarrolló un trabajo de grado titulado **“Diseño de un sistema de información para la automatización de las actividades llevadas a cabo en el área de operación y almacén de una empresa de energía eléctrica”** en el cual llevó a cabo la conceptualización de un sistema enfocado en la automatización de flujos de información con el objetivo de optimizar los tiempos necesarios para desarrollar las actividades del Área de Operaciones de la empresa objeto de estudio. El trabajo desarrollado produjo información que resultó de utilidad para introducir cambios en la estructura de operaciones de la organización, logrando notables mejoras en los costos en recursos humanos y materiales. [2]

Sánchez, M. (2005) desarrolló un trabajo de grado titulado **“Diseño de un sistema de información para automatizar algunas de las actividades relacionadas con el proceso de producción de crudo y gas, desde el yacimiento hasta las estaciones de flujo, que se realizan en una empresa petrolera, en Punta de Mata”**, en el que enfocó de nuevo el problema de la falta de automatización que tan desfavorables consecuencias produce en las organizaciones de gran envergadura, tal como es el caso de las empresas petroleras. En este caso, la información, debido a su alta tasa de generación, se acumulaba de manera que terminaba por ser inclasificable, incontrollable y, como resultado, inútil. El sistema automatizado desarrollado permitió procesar, almacenar e inclusive aportar a la generación de toda esta información relativa a las actividades y costos de la operación de los pozos y estaciones de flujo, de una manera organizada, entendible y utilizable para la toma de decisiones de la empresa. [3]

Cedeño, R. (2006) desarrolló un trabajo de grado titulado **“Diseño de un sistema de información para el proceso de formulación del presupuesto de inversiones de PDVSA gas distrito Anaco”**, en el cual

se enfrentó al problema causado por la metodología manual utilizada por la Gerencia de Finanzas de la empresa para la formulación de su presupuesto de inversiones. Tratándose de planes y actividades relacionadas con el tan importante recurso financiero de la empresa, la problemática tuvo que contemplarse a través de una estructura que garantizara tanto la automatización del proceso como la seguridad del mismo. El resultado del trabajo permitió obtener información rápida y relevante del proceso solo a los usuarios autorizados para ello, garantizando que los presupuestos fueran los justos para las situaciones que surgían en el día a día de la empresa. [4]

Rodríguez, M. (2006) desarrolló un trabajo de grado titulado **“Diseño de un sistema de información para el control interno y el manejo de proyectos en la gerencia de proyectos de una empresa consultora”** en el ámbito de una empresa consultora que se enfrentaba a problemas tan graves como desigualdades en fechas, y otros detalles de los proyectos, causados por la falta de sincronización que existía entre sus diferentes componentes, entre otros. La solución obtenida al final de la investigación permitió crear, manipular y actualizar de manera segura y confiable los datos de los proyectos manejados por la empresa, estableciendo nuevos parámetros de control interno que optimizaron sus operaciones. [5]

## **2.2. DEFINICIÓN DE SISTEMA**

“Un sistema es un todo complejo y organizado; una reunión de cosas y partes que forman un todo unitario y complejo. La idea de sistema da una connotación de plan, método, orden, arreglo. Lo antagónico de sistemas es el caos”. (Jonson, Kast y Rosenweig). [6]

## **2.3. SISTEMA DE INFORMACIÓN.**

Es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. [7]

Un sistema de información también puede definirse como “una federación de sistemas de información que está diseñado para apoyar los subsistemas funcionales de una organización”. Cada subsistema funcional requiere de las aplicaciones para realizar todo el procesamiento de información relacionado con dicha función, incluyendo archivos propios para cada subsistema, además una base común de datos. [8]

#### 2.4. OBJETIVOS DE LOS SISTEMAS DE INFORMACIÓN.

Los sistemas de información deben cumplir tres objetivos básicos dentro de las organizaciones, los cuales son:

- Automatizar los procesos operativos.
- Proporcionar información que sirva de apoyo al proceso de toma de decisiones.
- Lograr ventajas competitivas a través de su implantación y uso. [7]

#### 2.5. TIPOS DE SISTEMAS DE INFORMACIÓN.

- **Sistemas transaccionales:** son sistemas de información que logran la automatización de los procesos operativos dentro de una organización, ya que su función primordial consiste en procesar transacciones tales como: pagos, cobros, pólizas, entradas, salidas, etc.
- **Sistemas de soporte a la toma de decisiones (DSS):** un DSS no soluciona problemas, ya que solo apoya el proceso de toma de decisiones. La responsabilidad de tomar una decisión, de adoptarla y ponerle en práctica es de los administradores, no del DSS.
- **Sistemas de soporte para la toma de decisiones en grupos (GDSS):** su objetivo es lograr la participación de un grupo de personas durante la toma de decisiones en ambientes de anonimato y consenso, apoyando decisiones simultáneas.
- **Sistemas expertos de soporte para la toma de decisiones (EDSS):** Permiten cargar bases de conocimiento integrados por

una serie de reglas de sentido común para que diferentes usuarios las consulten, apoyen la toma de decisiones.

- **Sistemas de información para ejecutivos (EIS):** Están dirigidos a apoyar el proceso de toma de decisiones de los altos ejecutivos de una organización, presentan información relevante y usan recursos visuales y de fácil interpretación, con el objetivo de mantenerlos informados. [7]

## 2.6. BASE DE DATOS.

Es un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquina accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente y no predicable en el tiempo. [9]

Es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. [10]

Una base de datos es una colección de datos relacionados. Por datos, se quiere decir hechos conocidos que pueden registrarse y que tienen un significado implícito. Una base de datos tiene las siguientes propiedades implícitas:

- Una base de datos representa algunos aspectos del mundo real, en ocasiones denominado minimundo o Universo del Discurso (UdD). Los cambios en el minimundo se reflejan en la base de datos.
- Una base de datos es una colección coherente de datos con significados inherentes. Un conjunto aleatorio de datos no puede considerarse como una base de datos.
- Una base de datos se diseña, construye y puebla con datos para un propósito específico. Está destinada a un grupo de usuarios

concreto y tiene algunas aplicaciones preconcebidas en las cuales están interesados dichos usuarios. [11]

### 2.6.1. Componentes Principales de las Bases de Datos

- **Datos:** los datos son la Base de Datos propiamente dicha.
- **Hardware:** el hardware se refiere a los dispositivos de almacenamiento en donde reside la base de datos, así como a los dispositivos periféricos (unidad de control, canales de comunicación, etc.) necesarios para su uso.
- **Software:** está constituido por un conjunto de programas que se conoce como Sistema Manejador de Base de Datos (DMBS: Data Base Management System). Este sistema maneja todas las solicitudes formuladas por los usuarios a la base de datos.
- **Usuarios:** existen tres clases de usuarios relacionados con una Base de Datos: El programador de aplicaciones, quien crea programas de aplicación que utilizan la base de datos. El usuario final, quien accede la Base de Datos por medio de un lenguaje de consulta o de programas de aplicación. El administrador de la Base de Datos (DBA: Data Base Administrator), quien se encarga del control general del Sistema de Base de Datos. [11]

### 2.6.2. Clasificación de las Bases de Datos

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al criterio elegido para su clasificación:

- Según la variabilidad de los datos almacenados existen bases de datos estáticas y bases de datos dinámicas.
- Según su contenido pueden ser bases de datos bibliográficas, bases de datos numéricas, bases de datos de texto completo,

directorios, banco de Imágenes (audio, vídeo, multimedia, etc.) y por último las bases de datos o "bibliotecas".

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de datos:

- **Bases de datos jerárquicas:** estas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.
- **Bases de datos de red:** éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).
- **Bases de datos relacionales:** su artículo principal es el modelo relacional debido a que es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas".
- **Bases de datos orientadas a objetos:** este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento).
- **Bases de datos documentales:** permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes.

[11]

### 2.6.3. Modelo de datos

Una característica fundamental del enfoque de base datos es que proporciona cierto nivel de abstracción de los datos. Al ocultar detalles de almacenamiento que la mayoría de los usuarios no necesitan conocer. Un modelo de datos (colección de conceptos que sirven para describir la estructura de una base de datos) proporciona los medios necesarios para conseguir dicha abstracción. Con el concepto estructura de una base de datos se refiere a los tipos de datos, los vínculos y las restricciones que deben cumplirse para esos datos. La mayoría de los modelos de datos contienen un conjunto de operaciones básicas para especificar lecturas y actualizaciones de la base de datos. Entre los modelos de datos ampliamente utilizados se encuentran el modelo de datos relacional, el modelo de red, y el modelo jerárquico. [11]

## **2.7. SISTEMA DE GESTIÓN DE BASE DE DATOS (DBMS).**

Es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de una tarea específica.

El objetivo primordial de un sistema manejador base de datos es proporcionar un contorno que sea a la vez conveniente y eficiente para ser utilizado al extraer, almacenar y manipular información de la base de datos. Todas las peticiones de acceso a la base, se manejan centralizadamente por medio del DBMS, por lo que este paquete funciona como interface entre los usuarios y la base de datos. [12]

Una de las ventajas del DBMS es que puede ser invocado desde programas de aplicación que pertenecen a Sistemas Transaccionales escritos en algún lenguaje de alto nivel, para la creación o actualización de las bases de datos, o bien para efectos de consulta a través de lenguajes propios que tienen las bases de datos o lenguajes de cuarta generación. [10]



Es una colección de programas que permiten a los usuarios crear y mantener una base de datos. El DBMS es por tanto un sistema software de propósito general que facilita los procesos de definición, construcción, y manipulación de base de datos para distintas aplicaciones. A continuación se describen cada uno de estos procesos:

- **Definición:** la definición de una base de datos consiste en especificar los tipos de de datos, las estructuras y restricciones para los datos que se van a almacenar en dicha base de datos.
- **La construcción:** es el proceso de almacenar los datos concretos sobre algún medio de almacenamiento controlado por el DBMS.
- **La manipulación:** incluye funciones tales como consultar la base de datos para recuperar unos datos específicos, actualizar la base de datos para reflejar los cambios ocurridos en el minimundo, y generar informes a partir de los datos. [11]

### 2.7.1. Arquitectura de un SGBD

Una arquitectura para los sistemas de base de datos es la denominada arquitectura de tres esquemas. El objetivo de esta es separar las aplicaciones del usuario y la base de datos física. En esta arquitectura se definen esquemas en los tres siguientes niveles:

- El nivel interno tiene un esquema interno, que describe la estructura física de almacenamiento de la base de datos. El esquema interno emplea un modelo de datos físico y describe todos los detalles para su almacenamiento, así como los caminos de acceso para la base de datos
- El nivel conceptual tiene un esquema conceptual, que describe la estructura de la base de datos completa para una comunidad de usuarios. El esquema conceptual oculta los detalles de las estructuras físicas de almacenamiento y se concentra en describir

entidades, tipos de datos, vínculos, operaciones de los usuarios y restricciones. En este nivel podemos usar un modelo de datos de alto nivel o uno de implementación.

- El nivel externo o de vistas incluye varios esquemas externos o vistas de usuario. Cada esquema externo describe la parte de la base de datos que interesa a un grupo de usuarios determinado, y oculta a ese grupo el resto de la base de datos. [11]

### 2.7.2. Clasificación de los Sistemas de Gestión de Bases de Datos

El principal criterio que suele utilizarse para clasificar los SGBD es el modelo de datos en que se basan. Los modelos de datos empleados con mayor frecuencia en los SGBD comerciales actuales son el relacional, el de red y el jerárquico. A continuación se describen brevemente cada uno de estos:

- **El modelo de datos relacional:** representa una base de datos como una colección de tablas, cada una de las cuales se pueden almacenar en forma de archivo individual. Casi todas las bases de datos relacionales tienen lenguaje de consulta de alto nivel y manejan una forma limitada de vistas de usuarios.
- **El modelo de datos de red:** representan los datos como tipos de registros y también representa un tipo limitado de vínculos 1:N, llamado tipo de conjunto. Donde los tipos de registros aparecen como rectángulos y los tipos de conjuntos como flechas dirigidas rotuladas. Este modelo también tiene un lenguaje de registro por registro asociado que se debe incorporar en un lenguaje de programación anfitrión.
- **El modelo jerárquico:** representan los datos como estructura jerárquica árbol. Cada jerarquía representa varios registros relacionados entre sí. No existe un lenguaje estándar para el

modelo jerárquico aunque, aunque la mayor parte de los SGBD jerárquicos cuentan de registro por registro. [11]

## 2.8. INGENIERÍA DE SOFTWARE.

Es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software", es decir, "permite elaborar consistentemente productos correctos, utilizables y costo-efectivos". [13]

El proceso de ingeniería de software se define como "un conjunto de etapas parcialmente ordenadas con la intención de logra un objetivo, en este caso, la obtención de un producto de software de calidad". [14]

La ingeniería del software es el establecimiento y uso de principios de la ingeniería para obtener económicamente un software confiable y que funcione de modo eficiente en máquinas reales. Esta estratificada en cuatro etapas claves:

- **Un enfoque de calidad:** cualquier enfoque de la ingeniería (incluido el de la ingeniería del software) debe estar sustentado en un compromiso de calidad. Como por ejemplo: La Gestión de Calidad Total, Sigma Seis y otros enfoques similares que fomentan una cultura de mejora continua del proceso, y esta cultura es la que al final conduce al desarrollo de enfoques muy efectivos para la ingeniería del software.
- **El proceso:** el proceso del software forma la base para el control de la gestión de los proyectos de software y establece el contexto en el cual se aplican los métodos técnicos, se generan los productos del trabajo (modelos, documentos, datos, reportes, formatos, etcétera), se establecen los fundamentos, se asegura la calidad, y el cambio se maneja de manera apropiada.

- **Los métodos:** los métodos abarcan un amplio espectro de tareas que incluyen la comunicación, el análisis de requisitos, el modelado del diseño, la construcción del programa, la realización de pruebas y el soporte. Los métodos de la ingeniería del software se basan en un conjunto de principios básicos que gobiernan cada área de la tecnología e incluye actividades de modelado y otras técnicas descriptivas.
- **Las herramientas:** las herramientas de la ingeniería en software proporcionan el soporte automatizado o semiautomatizado para el proceso y los métodos. Cuando las herramientas se integran de forma que la información que cree una de ellas pueda usarla otra, se dice que se ha establecido un sistema para el soporte del desarrollo del software. [15]

### **2.8.1. Proceso Unificado de Desarrollo de Software**

El proceso unificado es un intento encaminado a reunir los mejores rasgos y características de modelos de procesos de software, pero los caracteriza de manera que implementa muchos de los principios del desarrollo ágil de software. Reconoce la importancia de la comunicación con el cliente y los métodos encaminados a describir el punto de vista de un cliente con respecto a un sistema (por ejemplo, el caso de uso).

El proceso unificado enfatiza el importante papel de la arquitectura del software, y ayuda al arquitecto a enfocarse en las metas correctas, como el entendimiento, el ajuste de los cambios futuros y la reutilización. Sugiere un flujo de proceso iterativo e incremental y que proporciona el sentido evolutivo esencial en el desarrollo del software moderno. [15]

### **2.8.2. Fases del Proceso Unificado**

- **Fase de Inicio:** la fase de inicio abarca la comunicación con el cliente y las actividades de planeación. Al colaborar con los clientes y usuarios finales se identifican los requisitos de negocios para el software, se propone una arquitectura aproximada para el sistema, y se desarrolla un plan para la naturaleza iterativa e incremental del sistema subsiguiente. Los requisitos fundamentales de negocios se describen a través de un conjunto preliminar de casos de uso que describen cuales características y funciones son deseables para cada clase importante de usuarios.
- **Fase de Elaboración:** abarca la comunicación con el cliente y las actividades de modelado del modelo genérico del proceso. La elaboración refina y expande los casos de usos preliminares que se desarrollaron como una parte de la fase de inicio; además, expande la representación arquitectónica para incluir cinco visiones diferentes del software: el modelo de caso de uso, el modelo de análisis, el modelo de diseño, el modelo de implementación y el modelo de despliegue.
- **Fase de Construcción:** la fase de construcción desarrolla o adquiere los componentes del software que harán que cada caso de uso sea operativo para los usuarios finales. Lograr esto requiere que los modelos de análisis y diseño iniciados durante la fase de elaboración se completen para reflejar la versión final del incremento del software.
- **Fase de Transición:** el software se entrega a los usuarios finales para realizar pruebas beta, y la retroalimentación del usuario reporta tanto defectos como cambios necesarios. Además el equipo de software crea la información de soporte necesaria (por ejemplo, manuales de usuario, guías de resolución de problemas, procedimientos de instalación) para el lanzamiento.
- **Fase de Producción:** durante esta fase se monitorea el uso subsiguiente del software, se proporciona el soporte para el ambiente

operativo (infraestructura), y se reciben y evalúan los informes de defectos y los requerimientos de cambios.

Es probable que mientras se realizan las fases de construcción, transición, y producción ya se hayan iniciado los trabajos para el siguiente incremento del software. Esto significa que las cinco fases del proceso unificado no suceden en una secuencia, sino en una concurrencia por etapas. [14]

## 2.9. LENGUAJE DE MODELADO UNIFICADO (UML).

UML son las siglas en inglés, (*Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. El Lenguaje de Modelado Unificado define un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación. UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real.

UML ofrece una gran variedad de diagramas en los cuales modelar sistemas, entre los cuales destacan:

- **Diagrama de Caso de Uso:** Muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea.

- **Diagrama de Secuencia:** Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo. Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren.
- **Diagrama de Colaboración:** Un Diagrama de Colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere). Los Diagramas de Colaboración muestran las relaciones entre los roles de los objetos.
- **Diagrama de Clase de Análisis:** Los Diagramas de Clases de Análisis son utilizados por los desarrolladores de software para especificar los requerimientos funcionales, considerando una o varias clases, o subsistemas del sistema a desarrollar.
- **Diagramas de Clases de Diseño:** Los diagramas de clase de diseño representan un conjunto de elementos del modelo que son estáticos, como las clases y sus tipos, sus contenidos y las relaciones que se establecen entre ellos. [15]

## El UML y la Industria del Software

El UML se ha vuelto el estándar de facto (impuesto por la industria y los usuarios) para el modelado de aplicaciones de software. En los últimos años, su popularidad trascendió al desarrollo de software y, en la

actualidad, el UML es utilizado para modelar muchos otros dominios, como por ejemplo el modelado de procesos de negocios.

### **Conceptos básicos sobre UML**

UML son las siglas para Unified Modeling Language, que en castellano quiere decir: Lenguaje de Modelado Unificado.

- **Lenguaje:** el UML es, precisamente, un lenguaje. Lo que implica que éste cuenta con una sintaxis y una semántica. Por lo tanto, al modelar un concepto en UML, existen reglas sobre cómo deben agruparse los elementos del lenguaje y el significado de esta agrupación.
- **Modelado:** el UML es visual. Mediante su sintaxis se modelan distintos aspectos del mundo real, que permiten una mejor interpretación y entendimiento de éste.
- **Unificado:** unifica varias técnicas de modelado en una única.

### **OMG**

La OMG (Object Management Group) es una asociación sin fines de lucro formada por grandes corporaciones, muchas de ellas de la industria del software, como por ejemplo: IBM, Apple Computer, Sun Microsystems Inc y Hewlett-Packard. Esta asociación se encarga de la definición y mantenimiento de estándares para aplicaciones de la industria de la computación. Otro de los estándares definidos por la OMG, además del UML, es CORBA, el cual permite interoperabilidad multiplataforma a nivel de objetos de negocio.

### **Poniendo UML a trabajar**

Un modelo de UML proporciona a menudo apenas una vista de un sistema entre muchas vistas necesitadas para construir o para documentar realmente el sistema completo. Los usuarios nuevos a UML pueden caer en la trampa de intentar modelar todo sobre su sistema con



un solo diagrama y terminar cayendo en la pérdida de la información crítica. O, en el otro extremo, pueden intentar incorporar cada diagrama posible de UML en su modelo, de tal modo que complican demasiado y crean una pesadilla del mantenimiento.

El llegar a ser perito con UML significa entender lo que tiene que ofrecer cada diagrama y sabiendo cuándo aplicarlo. Habrá muchas veces en que un concepto se podría expresar usando cualquier número de diagramas; hay que escoger el diagrama o los diagramas que serán más significativos para la mayoría de los usuarios. [16]

### **Reglas para el UML**

Mientras que UML proporciona un lenguaje común para la captura de funcionalidad e información del diseño, es deliberadamente ampliable permitir la flexibilidad necesaria para modelar diversos dominios. Hay varias reglas a tener presente al usar UML:

- **Casi todo en UML es opcional:** UML proporciona un lenguaje para la captura de información que varía grandemente dependiendo del dominio del problema. En hacer eso, hay a menudo partes de UML que no aplican a un problema particular o pueden no prestar cualquier cosa a la visión particular que se esté intentando representar. Es importante tener presente que no se necesita utilizar cada parte de UML en cada modelo que se crea. Posiblemente más importantemente, no se necesita utilizar cada símbolo disponible para un tipo de diagrama en cada diagrama que se cree. Se debe mostrar solamente lo que ayuda a clarificar el mensaje que se está intentando representar, y dejar lo que no se necesita. Hay ocasionalmente más que una forma para representar la misma información; hay que usar cuál es familiar a la audiencia.

- **Los modelos de UML son raramente completos:** como consecuencia de que todo es opcional, es común que un modelo de UML falten algunos detalles sobre un sistema. El truco es no perder los detalles claves que podrían afectar el diseño del sistema. Saber cuál es un detalle clave contra la información extraña viene con la experiencia; sin embargo, usar un proceso iterativo y la nueva revisión del modelo ayuda a dejar sólo lo que necesita estar allí. [16]

## **2.10. MÉTRICAS O INDICADORES DE SOFTWARE**

Una métrica de software es una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Un indicador es una métrica o una combinación de métricas que proporcionan una visión profunda del proceso de software o del producto en sí. Las métricas de software relacionan de alguna forma las medidas individuales sobre algún aspecto (por ejemplo: el número medio de errores encontrados por revisión o el número medio de errores encontrados por persona y hora en revisiones)

Un indicador proporciona una visión profunda que permite al gestor de proyectos o a los ingenieros de software ajustar el producto, el proyecto o el proceso para que las cosas salgan mejor. [17]

### **2.10.1. Clasificación Básica de las Métricas de Software**

Existen innumerables métricas con propósitos diferentes que reflejan o describen la conducta del software, estas pueden medir entre otros aspectos la competencia, calidad, desempeño y la complejidad del software contribuyendo a establecer de una manera sistemática y objetiva una visión interna del trabajo mejorando así la calidad del producto.

A continuación se muestra la clasificación de las mismas:

**Métricas de Complejidad:** Son todas las métricas de software que definen de una u otra forma la medición de la complejidad; Tales como volumen, tamaño, anidaciones, costo (estimación) y configuración. Estas son los puntos críticos de la concepción, viabilidad, análisis, y diseño de software.

**Métricas de Calidad:** Son todas las métricas de software que definen de una u otra forma la calidad del software; tales como exactitud, estructuración o modularidad, pruebas, mantenimiento, reusabilidad, entre otras. Estas son los puntos críticos en el diseño, codificación, pruebas y mantenimiento.

**Métricas de Competencia:** Son todas las métricas que intentan valorar o medir las actividades de productividad de los programadores o practicantes con respecto a su certeza, rapidez, eficiencia y competencia.

**Métricas de Desempeño:** Corresponden a las métricas que miden la conducta de módulos y sistemas de un software, bajo la supervisión del sistema operativo o hardware. Generalmente tienen que ver con la eficiencia de ejecución, tiempo, almacenamiento, complejidad de algoritmos computacionales, etc.

**Métricas Estilizadas:** Son las métricas de experimentación y de preferencia; Por ejemplo: estilo de código, las convenciones denominando de datos, las limitaciones, etc. Pero estas no se deben confundir con las métricas de calidad o complejidad.

**Variedad de Métricas:** Tales como portabilidad, facilidad de localización, consistencia, etcétera.

Estas clasificaciones de métricas fortalecen la idea, de que más de una métrica puede ser deseable para valorar la complejidad y la calidad

del software, teniendo en cuenta que para ello es necesario medir los atributos del software. [18].

### **2.11. MODELO DE ESTIMACIÓN DE COSTOS (APLICADO AL PROCESO DE DESARROLLO DE SOFTWARE).**

Todo proyecto de ingeniería de software debe partir con un buen plan, pero lamentablemente, la planificación es una tarea nada trivial. Uno de los aspectos que dificultan la labor de administradores y jefes de proyecto en torno a la planificación es la difícil tarea de realizar una estimación de costos y plazos realista.

La estimación es más arte que ciencia; también es parte de la etapa de la planificación y algunas actividades de la ingeniería. La diferencia en la estimación de costos entre ingeniería de software y otras disciplinas es que en ingeniería de software lo principal para las personas es el costo; y en otras disciplinas el costo de las cosas materiales depende de la actividad.

Existen técnicas para la estimación de costos, pero para ello se requiere experiencia, acceso a una buena información histórica y coraje para confiar en medidas cuantitativas cuando todo lo que existe son datos cualitativos.

El manejador de costo principal para un proyecto de desarrollo de software es sin duda el tamaño del producto. La medida del tamaño debe ser tal que esté en relación directa con el esfuerzo de desarrollo, por lo que las métricas de tamaño tratan de considerar todos los aspectos que influyen en el costo, como tecnología, tipos de recursos y complejidad. [19]

### **2.12. MODELO DE ESTIMACIÓN EMPÍRICA (APLICADO AL PROCESO DE DESARROLLO DE SOFTWARE).**

Un modelo empírico de estimación para software puede utilizar fórmulas derivadas empíricamente para predecir el esfuerzo como una función del número de líneas de código (LDC) o los llamados “*Puntos de Funcionalidad*” (PF).

Los datos empíricos que soportan la mayoría de los modelos de estimación se obtienen de una muestra limitada de proyectos. Es por eso que estos modelos de estimación no son adecuados para todas clases de software y en todos los entornos de desarrollo. Por consiguiente, los resultados obtenidos de dichos modelos se deben utilizar con prudencia.

#### **[20]**

Barry Boehm en su libro sobre “Economía de la Ingeniería del Software”, menciona una escala de modelos de estimación de software con el nombre de COCOMO, por COonstrucive COst MOdel (MOdelo COnstructivo de COsto). La escala de modelos de Boehm incluye:

- *Modelo 1.* El modelo COCOMO *básico* calcula el esfuerzo (y el costo) del desarrollo de software en función del tamaño del programa, expresado en las líneas estimadas de código (LDC).
- *Modelo 2,* El modelo COCOMO *intermedio* calcula el esfuerzo del desarrollo de software en función del tamaño del programa y de un conjunto de “conductores de costo” que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.
- *Modelo 3,* El modelo COCOMO *avanzado* incorpora todas las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de costo en cada fase (análisis, diseño, etc.) del transcurso de ingeniería del software.

### **2.13. PROGRAMACIÓN ORIENTADA A OBJETOS**

La programación orientada a objetos es un conjunto completo de conceptos e ideas. Es una manera de pensar en el problema al que va dirigido un programa de ordenador y de enfrentarlo de modo más intuitivo e incluso más productivo. En un lenguaje orientado a objetos verdadero, toda entidad del dominio del problema se expresa a través del concepto de objetos. Los programas escritos para simular los objetos del mundo real para el dominio del problema son mucho más fáciles de diseñar y escribir porque permiten pensar de un modo más natural.

Por definición, los objetos comprenden datos y métodos que trabajan con esos datos. Una clase es un diseño para un determinado conjunto de funcionalidad, y un objeto creado tomando como base una determinada clase tiene toda la funcionalidad de esa clase a partir de la que se ha construido. Un objeto es una instancia o ejemplar de una clase. [21]

#### **2.13.1. Principios de los lenguajes orientados a objetos**

Según Bjarne Stroustrup, autor del lenguaje de programación C++, para que un lenguaje se llame a sí mismo orientado a objetos debe soportar tres conceptos: objetos, clases y herencia. Sin embargo, ha llegado a pensarse más comúnmente que los lenguajes orientados a objetos son lenguajes contruidos sobre el trípode encapsulación, herencia y polimorfismo. La razón de este cambio de filosofía es que con el paso de los años, los desarrolladores de software han llegado a darse cuenta que la encapsulación y el polimorfismo son partes tan integrantes de la construcción de sistemas orientados a objetos como la clase y la herencia. [20]

#### **2.13.2. Encapsulación**

Habilidad de un objeto para esconder sus datos y métodos internos y de presentar una interfaz que hace, hablando desde el punto de vista del

programa, accesibles las partes importantes del objeto. Los procedimientos internos sobre cómo lleva a cabo un objeto su trabajo no son importantes mientras que ese objeto pueda desempeñar su trabajo. [21]

### **2.13.3. Herencia**

Está relacionada con la habilidad del programador para especificar que una clase tiene una relación “especie de” con otra clase. A través de la herencia, se puede crear (o derivar) una nueva clase que esté basada en una clase ya existente. Entonces se puede modificar la clase de la manera que se quiera y crear objetos nuevos del tipo derivado. Esta habilidad es la esencia de la creación de la jerarquía de clases. Una clase derivada es la nueva clase que se está creando y la clase base es desde la que se deriva la nueva clase. La clase nueva derivada hereda todos los miembros de la clase base, para así posibilitar la reutilización del trabajo anterior. [21]

### **2.13.4. Polimorfismo**

Funcionalidad que permite al código antiguo invocar código nuevo. Este es probablemente el mayor beneficio de la programación orientada a objetos, porque permite extender o mejorar un sistema sin romper o modificar el código existente. [21]

El polimorfismo proporciona al menos dos beneficios. Primero permite la capacidad de agrupar objetos que tienen una clase base en común y tratarlos consistentemente. Y la segunda ventaja es la que ya se ha mencionado: el código antiguo puede utilizar código nuevo.

## **CAPÍTULO 3**

### **3.1. INTRODUCCIÓN**

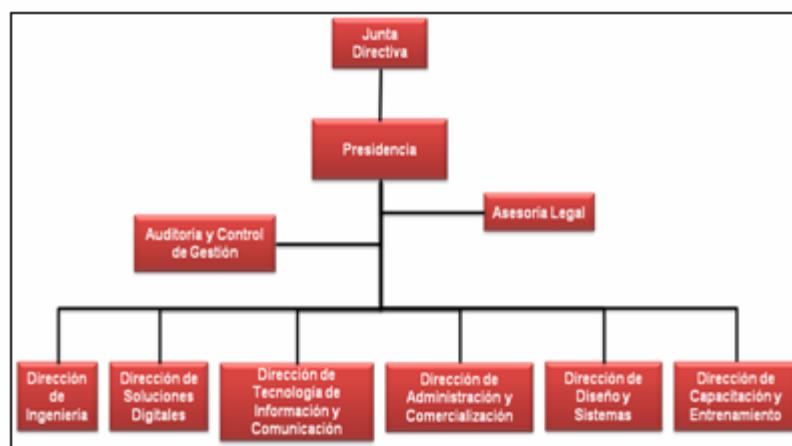
Cuando se pretende dar resolución a una situación problema es necesario previamente conocer de manera crítica el contexto de la misma. Solo de esta manera es posible entender las formas, sutiles o evidentes, en las que se verá afectado el sistema por los cambios que se puedan implementar, permitiendo así efectivamente predecir los pasos a tomar para mejorar la situación de la forma requerida. Esta búsqueda de un conocimiento global no viene sin dificultades anexas, típicas del entorno específico de estudio, las cuales, a su vez, se ven aumentadas exponencialmente al involucrase el impredecible e incontrolable factor humano. Aún más complejo que el ser humano, como ente actor y reactor ante un medio, es la tela de interacciones que llega a desarrollar con sus semejantes dentro de los sistemas en los que coexisten, los llamados sistemas humanos.

Esa es la realidad dentro de las organizaciones como la que se estudia en este caso, y debe comprenderse cabalmente aún más allá del problema planteado si se espera resolver con éxito el mismo y no resultar por completo ineficaces, aunque no lo sea así en apariencia. Aceptado, las organizaciones humanas con un objetivo claro ofrecen facilidades para el entendimiento de sus labores internas, al menos tienden un poco menos al caos, pero dar esto por sentado resulta en un error que se debe tener cuidado al evitar. Así, en este capítulo se detalla la búsqueda de esa comprensión, perfecta solo en ideal, que se presenta completamente necesaria para la correcta conceptualización del sistema a diseñar.

### **3.2. ESTRUCTURA ORGANIZATIVA DEL SISTEMA OBJETO DE ESTUDIO**



*Saetha Design & Systems* presenta una organización jerárquica si se quiere “tradicional” en este tipo de empresas, con los eslabones más altos de la cadena de mando bien establecidos y diferenciados de las ramas operativas y comerciales. En realidad, la compañía es parte de la iniciativa “*Empresas Saetha*” (Ver figura 3.1), la cual comprende una amplia variedad de servicios en el área de la ingeniería y comunicación, y funciona como una unidad de negocio independiente. No es de entenderse debido a esto como una compañía completamente apartada pues, de hecho, comparte el mismo liderazgo con sus homólogas de otros campos. Su atributo de independencia viene dado por la necesidad que tiene de producir ingresos suficientes para justificar su existencia dentro de la iniciativa sin resultar nunca una carga para las otras unidades de negocio, y su organización interna única diseñada para alcanzar sus objetivos estratégicos. De este modo, el sistema presenta una identidad propia con límites claramente marcados que simplifican el diagnóstico, eliminando las complicaciones inherentes al estudio de las interacciones entre las diferentes unidades. A continuación se muestran las características que definen a dicha unidad extraídas de su documentación interna oficial.



**Figura 3.1.** Organigrama Estructural de *Empresas Saetha*

### **3.2.1. Misión**

“**Saetha** es una organización multidisciplinaria que aporta soluciones tecnológicas y de negocios eficientes e innovadoras orientadas a satisfacer las necesidades del mercado, maximizando la rentabilidad y el valor de sus clientes, mediante el uso de herramientas adecuadas y un recurso humano altamente capacitado y motivado. Favorecer el desarrollo y la calidad de vida de la sociedad es el objetivo principal que se persigue”.

### **3.2.2. Visión**

“Convertirnos en una organización líder en innovación tecnológica y negocios en el mercado nacional, basados en el aprovechamiento continuo de nuestra capacidad creativa y en la generación de conocimiento, proyectándonos de forma competitiva a nivel mundial con presencia efectiva en múltiples países, contribuyendo así al desarrollo de nuestros clientes y al crecimiento científico, económico y cultural de la sociedad”.

### **3.2.3. Filosofía de Servicio de la Empresa**

- Elaborar productos y servicios innovadores y asertivos que satisfagan las necesidades del cliente.
- Ofrecer a nuestros clientes una costo-efectiva solución tecnológica.
- Contribuir con el mejoramiento profesional y económico de la sociedad.
- Transformar los desafíos en oportunidades de mejora del negocio de nuestros clientes.
- Ofrecer una alta calidad y confiabilidad en todos nuestros productos y servicios así como un excelente soporte técnico.

- Establecer una relación profesional, ética y agradable con nuestros clientes, manteniendo firmes políticas de transparencia, rectitud y buena fe.
- Utilizar recursos artísticos en la elaboración de nuestras soluciones.
- Trabajar con una enérgica vocación de servicio orientada al cumplimiento de las metas establecidas con nuestros clientes, compartiendo sus intereses y velando por su bienestar, para garantizar su continua satisfacción y lealtad.

#### **3.2.4. Descripción y Funciones Principales**

Dentro del mapa organizacional de la empresa (Ver figura 3.2) cada oficina o dependencia existe con un propósito previamente definido, lo cual genera un marco de trabajo para todos los empleados que laboran en ella. Este propósito encuentra su aplicación práctica en la definición de las funciones exactas que se realizan dentro de cada una, de manera abstracta al resto de la organización.

##### **3.2.4.1. Presidencia**

Esta oficina es la que carga con la mayor responsabilidad de la iniciativa, analizando efectivamente los indicadores de gestión generados por todas las vertientes de la misma con el objeto de desarrollar y monitorear la aplicación de la planificación estratégica.

##### **3.2.4.2. Dirección General**

La Dirección se ocupa de mantener a la empresa como un caso de negocio viable mediante el continuo control de la productividad. Participa activamente en la validación de las posibles estrategias a implementar y desarrolla planes de acción acorde a la persecución de estas.

##### **3.2.4.3. Gerencia de Desarrollo de Software**

La Gerencia de Desarrollo de Software depende directamente de la Dirección y es su misión producir el establecimiento táctico necesario para la ejecución de los planes de acción que hayan sido desarrollados en los niveles gerenciales, y requieran la construcción de nuevas aplicaciones de software. Sobre ella descansa la responsabilidad de asignar proyectos y recursos a las coordinaciones de área que la conforman, así como verificar la correcta finalización de los mismos.

#### **3.2.4.4. Coordinación de Desarrollo de Software**

Esta área de la empresa se dedica de manera operacional a la ejecución real de los proyectos de software. Asigna las actividades asociadas que se desprenden de estos últimos a su personal, ya sea individualmente o en grupos de trabajos según sus capacidades. También supervisa activamente su realización, asegurando la eliminación de las dificultades asociadas a cada una.

#### **3.2.4.5. Coordinación de Soluciones Web**

Esta coordinación, al igual que su equivalente de desarrollo de software, debe asignar el personal y coordinar en todo momento las actividades necesarias para la realización de los proyectos entrantes.

#### **3.2.4.6. Análisis de Proyectos**

La oficina de Análisis de Proyectos existe en respuesta a la necesidad de la empresa de evaluar y mejorar la productividad a través de la observación crítica de las metodologías aplicadas en los procesos internos y la constante investigación tecnológica. Como dependencia directa de la Gerencia de Desarrollo de Software, actúa de la mano con las coordinaciones para la implementación de mejoras en los procesos, estructuras organizacionales, e implementación de sistemas facilitadores del flujo de trabajo. Adicionalmente, realiza labores de levantamiento, conceptualización y diseño para proyectos externos que así lo exijan por su complejidad o carácter innovador.

#### **3.2.4.7. Gerencia de Soluciones Gráficas y Multimedia**

Esta dependencia táctica de la Dirección gestiona todos los planes de acción de la compañía que hagan referencia a la creación e implementación de recursos gráficos como herramienta de presentación y manejo de información, tanto a nivel externo como interno. Pone en marcha los proyectos que acarreen estos planes pasando la documentación necesaria a la coordinación. Debido a su naturaleza, esta oficina se ve normalmente muy ligada a los esfuerzos mercadotécnicos de toda la iniciativa Saetha.

#### **3.2.4.8. Coordinación de Soluciones Gráficas y Multimedia**

Esta coordinación funciona a nivel operacional tal como aquellas encontradas en el área de desarrollo de software. Agrupa a los diferentes elementos técnicos y artísticos que dependen de ella, gestionando sus actividades en función de la ejecución de los proyectos individuales entrantes. Como las otras, tiene la responsabilidad de asegurar la ejecución de cada actividad individual dentro del alcance del proyecto que se esté trabajando, evitando cualquier posible retraso.



Figura 3.2. Estructura Organizacional de *Saetha Design & Systems*.

### 3.3. ANÁLISIS DEL SISTEMA ACTUAL

Dentro de Saetha D&S, como en cualquier empresa, existen dos tipos básicos de proyecto: interno y externo. La principal diferencia entre ellos a nivel operativo es la fuente de los mismos, siendo los internos generados normalmente por la oficina de análisis de proyectos mientras que, por su parte, los externos surgen directamente de la actividad de ventas y los intercambios comerciales que la compañía negocia con otras organizaciones. Para efectos de producción, estas diferencias no juegan un papel decisivo ya que la empresa no posee esquemas que dependan de ellas en cuanto al manejo de recursos humanos o financieros.

Una vez un proyecto se encuentra en estatus activo (la orden de inicio de actividades ha sido dada por el gerente de desarrollo) recae sobre los coordinadores la responsabilidad de asignar las actividades a los grupos de trabajo respectivos. Posteriormente, los empleados

proceden a ejecutar las actividades que les hayan sido asignadas en el orden definido, aunque es sobreentendido que sus prioridades pueden ser reajustadas por los coordinadores de ser necesario. De esta manera las tareas asociadas a un proyecto son distribuidas entre varios empleados que pueden o no trabajar juntos según el proceso lo requiera.

Cuando se hace necesaria la intervención del cliente como emisor de información relevante al proyecto, o tan solo una opinión crítica, es común observar que el contacto sea realizado directamente por la parte interesada (empleado) de no estar disponible un representante de la empresa con la autoridad necesaria para dicho trámite. Si lo que se requiere del cliente es relativamente importante y no puede ser procurado de manera inmediata – otro caso común – se le solicita a éste que satisfaga el requerimiento a la brevedad posible; durante este lapso de tiempo los empleados se dedican a otros proyectos activos, sin embargo, no existe un cambio de estatus formal para aquel que quedó “en espera” ni se genera documentación al respecto. Tampoco se mide el tiempo que el proyecto pasa efectivamente “en ejecución”.

Dado que el proyecto posea ya todos los insumos necesarios, las actividades son completadas y el producto final entregado, cerrando efectivamente la transacción de la empresa con su cliente.

En la etapa inicial de la vida del proyecto, esa en la que se asignan las actividades, es importante para la empresa, sobre todo a nivel táctico, estimar de alguna forma el tiempo que un empleado necesita para finalizar una tarea dada. Esto se hace en función de poder planificar de una manera más eficiente el orden en que se debe atacar cada proyecto y así maximizar la productividad. Dicha estimación nace del conocimiento de la dificultad que implica un proyecto y las capacidades específicas de las personas involucradas. Así, las estimaciones pueden ser muy variables debido a que se basan en razonamientos casi por completo

subjetivos. En ocasiones, el coordinador no está en capacidad de evaluar por su cuenta el tiempo de finalización de una actividad y esto acarrea que la estimación sea realizada por el empleado asignado a la misma. En cualquiera de los casos, el tiempo estimado de culminación es el resultado de recuerdos de casos de éxito y datos asumidos sobre los requisitos técnicos del proyecto – a pesar de desempeñarse aceptablemente en la mayoría de los casos, ¡está lejos de representar un conocimiento científico real!

Por otro lado, en las etapas finales, se programa una auditoría al resultado del proceso de desarrollo del software como parte del control de calidad ejercido sobre él. **Esta auditoría está orientada por completo al producto, nunca al proceso como tal**, y es completamente independiente de otras realizadas en proyectos alternos. Igualmente, los datos recopilados de auditorías pasadas no juegan ningún rol en las presentes; son normalmente organizados en función de la corrección inmediata de los errores encontrados y no tienen ninguna durabilidad en el tiempo.

Finalmente superado este paso el software está técnicamente listo para ser implementado, pero aún debe superar una última evaluación por parte del gerente de desarrollo. Se da muchas veces el caso en el que debido a la falta de estandarización de la que sufren los criterios utilizados, la tarea de evaluar termina siendo del director de la unidad o inclusive el presidente de la empresa. No obstante, la tendencia es hacia la descentralización en este tipo de decisiones. La evaluación de la relación **Calidad/Tiempo de Ejecución** es ejecutada por los mismos entes y es aún más subjetiva que la descrita anteriormente. Normalmente es ignorada a menos que exista un retraso considerable en el proyecto o la calidad del resultado sea notablemente deficiente, casos que rara vez



se dan y, como es de esperar, suelen tomar por sorpresa a la directiva táctica inmediata.

A través de la combinación de estas actividades la empresa cumple con sus compromisos de una manera bastante satisfactoria, si no necesariamente eficiente, que deja una buena impresión con todos sus clientes y asociados.

### **3.4. ANÁLISIS DE LA PROBLEMÁTICA**

La falta de organización que se evidencia en la descripción de los procesos asociados al control y a la evaluación del trabajo trae consecuencias que no son fácilmente observables ya que se diluyen en una sensación de “*eficiencia suficiente*”. Una de estas consecuencias, fácilmente la más importante, se da al momento de iniciarse un nuevo proyecto.

Cuando el proyecto es facturado (cotizado), se hace con el costo estimado de las horas totales de ejecución, más una utilidad esperada, por supuesto, basada como se ha dicho en los casos de éxito anteriores que la empresa tiene en su haber. Al ser asignadas las actividades posteriormente, se estiman tiempos de trabajo que no están relacionados con esta facturación, sino con la administración del tiempo de trabajo de los empleados y su distribución en los diferentes proyectos. Ya que no se mide la duración del trabajo real, se posibilita una diferencia entre el estimado y la realidad que permanece oculta a menos que el proyecto exceda en ejecución el tiempo facturado total por causa de su acumulación. En consecuencia, **la empresa no sabe en ningún momento a ciencia cierta si está obteniendo la utilidad deseada** pues no mide cuánto tiempo ha invertido realmente en el proyecto. Puede darse el caso en que ignoren incluso si están perdiendo dinero en una inversión.

A medida que aumenta el volumen de proyectos el problema descrito lo hace también proporcionalmente pero, irónicamente, se hace más difícil de localizar debido al ingreso de más personal en el nivel operacional. Simplemente, no puede saberse de quién es la responsabilidad del retraso observado, si es que es fácilmente observable, sin invertir para ello tiempo y recursos.

Todo esto es resultado directo de las deficientes prácticas en la estimación de tiempos de desarrollo y la falta de mecanismos de control adecuados en las tareas asociadas al proyecto. El problema se presenta como un ente recursivo, produciendo más de sí mismo con el paso del tiempo. **La mala estimación y control produce un vacío de información que dificulta las estimaciones futuras, repitiendo el esquema.**

Otro problema que afecta bastante a la organización viene dado por la falta de estandarización de los controles de calidad, tanto de los productos como del proceso de desarrollo del que nacen. La necesidad de hacer que todos los productos pasen por una revisión hecha por personal con criterio, descubriéndose muchas veces errores de forma y fondo, conduce a pérdidas de tiempo que a menudo obstaculizan el funcionamiento regular de la empresa. De hecho, esto genera aún otra dificultad también difícil de observar a simple vista.

La compañía no sabe si está mejorando como desarrolladora de software. La falta de una recopilación sistemática de datos pasados en cuanto a los ya discutidos tiempos reales de ejecución y a la calidad del software desarrollado – “*al gerente le gustó*” no es una medida científica – niegan rotundamente la posibilidad de localizar, analizar, y mejorar sobre los errores que afectaron a los trabajos anteriores. **El crecimiento sin optimización es una fórmula probada para el fracaso.**

## **CAPÍTULO 4**

### **4.1. INTRODUCCIÓN**

Así como es primordial entender el contexto de la problemática presente en el sistema y las variables que la afectan y sobre las cuales ella actúa en cambio, es también de suma importancia tener una amplia conciencia de lo que se requiere para darla por resuelta. En cuanto a los sistemas de información (SI) como solución viable, conocer estos llamados “*requerimientos*” involucra concebir al sistema dentro del contexto previamente estudiado, imaginar sus posibles intercambios con los entes ya conocidos y vislumbrar las maneras en que a través de ellos podrían llevarse a término las dificultades identificadas. Es este acaso el paso crítico de su desarrollo, ¡*comprender el Deber Ser del sistema!*

En el capítulo presentado a continuación se identifican, analizan y diagraman esas, por ahora, muy abstractas características que eventualmente definirán a la solución tecnológica diseñada.

### **4.2. ANÁLISIS DE LOS REQUERIMIENTOS DEL SISTEMA**

Un sistema de información es esencialmente un esquema de transmisión de datos organizados en función de una tarea concreta, de una entidad emisora a una receptora – muchas veces la misma – en un tiempo específico. Se hace entonces evidente que su composición interna es completamente dependiente de esos usuarios con los que debe ser capaz de interactuar, de esa información que debe ser capaz de manejar, y de la manera y momento en que la debe ser presentada de vuelta al entorno.

De este modo, entender a los usuarios y lo que ellos proveerán y demandarán del sistema dirige a la conceptualización precisa de lo que debe hacer éste para cumplir con la tarea, e incluso cómo puede estar capacitado para hacerlo.

La importancia de los usuarios es incuestionable; en su estudio y posterior análisis han sido necesarias una serie de reuniones con la directiva de la empresa para identificarlos y definir las labores que se espera puedan realizar por medio del software a diseñar.

Las entrevistas no estructuradas realizadas revelaron que eran los pasos a seguir desde el inicio de un proyecto hasta su finalización los que debían tomarse principalmente en cuenta para la automatización, así como el manejo posterior de la información recabada. Esto incluye la estimación de tiempo de desarrollo del proyecto, la asignación de actividades, el control de los tiempos de desarrollo y la evaluación de los resultados; todas actividades problemáticas para la empresa.

#### **4.2.1. Requerimientos Funcionales**

Como se ha dicho, el sistema diseñado debe estar en capacidad de ejecutar una cierta cantidad de acciones para satisfacer las necesidades de los usuarios que lo utilizarán. Estas "*funciones*" a desarrollar son definidas a continuación:

- El sistema debe contar con una aplicación de control de usuarios que garantice la seguridad de los datos y su usabilidad.
- Deben poder almacenarse para su posterior asignación en los procesos todos los elementos humanos del área de desarrollo.
- El sistema debe contar con una aplicación de configuración que permita alterar los datos relativos al cálculo de los costos operacionales.
- Deben poder asignarse a empleados o grupos de trabajo específicos actividades relacionadas con los proyectos.
- Se debe contar con una interfaz dirigida al empleado que permita declarar o medir de alguna forma el tiempo real de desarrollo.

- El sistema debe estar en capacidad de evaluar empíricamente si el resultado del trabajo – individual, grupal o total – es satisfactorio y servir como guía para las bonificaciones adicionales de los empleados. Dicha evaluación debe ser auditable.
- Se debe permitir estimar tiempos de desarrollo futuros en función de los datos de proyectos de características similares culminados previamente.
- El sistema debe presentar reportes claros sobre los tiempos de trabajo de los proyectos y sus costos asociados.

#### **4.2.2. Especificación de los Requerimientos**

Los requerimientos listados anteriormente, aunque engloban por completo las funcionalidades macro del sistema, reflejan muy poco de él a un nivel técnico. Para aumentar la veracidad de los mismos, y llevarlos al punto en el que pueden resultar verdaderamente útiles para el diseño del software, se expresaron en diagramas utilizando la herramienta del UML. Concretamente, se utilizaron en esta etapa los diagramas de Casos de Uso, Clases de Análisis y de Colaboración. El primero se empleó debido a las facilidades que ofrece para correlacionar cada requisito específico del sistema con un escenario de uso e identificar nuevos que se deriven de su cumplimiento, asegurando que no existan redundancias (o faltantes) en el diseño. El segundo y tercero, por su parte, se eligieron por la necesidad de convertir esos escenarios en clases definidas (¡requeridas!) y las diferentes interacciones internas que se deben considerar.

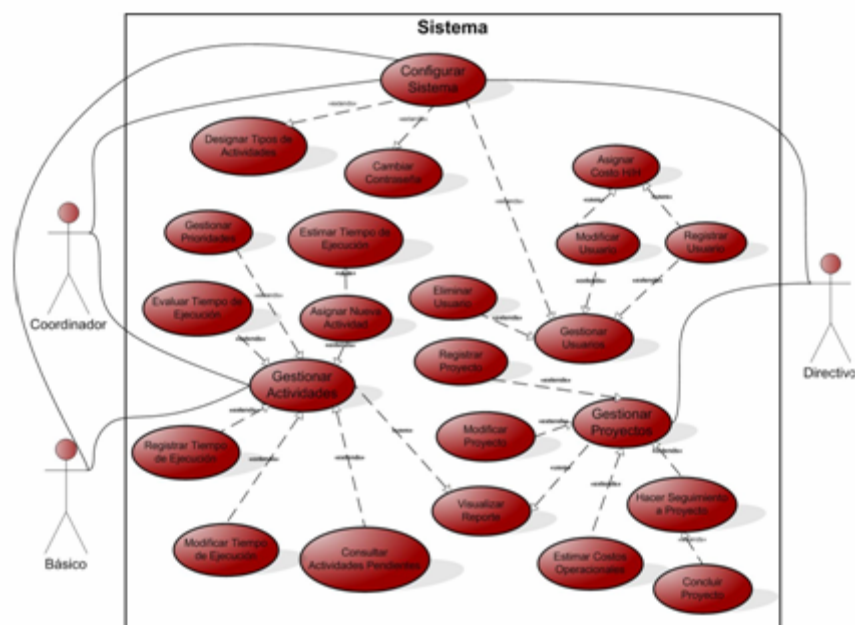
En general, los diagramas mostrados a continuación permiten ampliar los requerimientos de los usuarios y obtener requerimientos técnicos a cambio.

##### **4.2.2.1. Diagramación de los Casos de Uso del Sistema**

Para determinar los casos de uso se debieron primero identificar a los actores, es decir, las entidades que se beneficiarían de las funcionalidades del sistema. A continuación la lista:

- A. Directivo:** Este tipo de usuario es el que debe ser capaz de ejecutar las labores de control de usuarios y dar la voz de inicio y finalización de los proyectos. Para ello está en constante supervisión de los estatus variantes de los mismos. También es el encargado de revisar el resultado del trabajo total, evaluando la productividad y la calidad dadas las herramientas provistas por el sistema. En general, este tipo de usuario tendrá la responsabilidad de estimar las horas totales máximas de realización de un proyecto, en función de las horas que fueron efectivamente facturadas.
- B. Coordinador:** El usuario coordinador es aquel que asigna a los equipos de trabajo y supervisa las actividades que vienen dadas por cada etapa del proyecto. Administra los recursos humanos en base a las estimaciones efectuadas por el sistema.
- C. Básico:** Los empleados de la coordinación son aquellos usuarios básicos cuyo rol dentro del sistema es reportar sus actividades diarias para la medición del tiempo efectivo de trabajo del proyecto. A través del sistema, podrá consultar los datos de las tareas que deben ser completadas. Los usuarios de niveles superiores poseen implícitamente las capacidades de aquellos de nivel inferior.

Una vez descritos claramente los actores se procedió a la representación de los casos de uso del sistema como se muestra en la **figura 4.1**.



**Figura 4.1.** Diagrama de Casos de Uso del Sistema

Lo mostrado en el diagrama de Casos de Uso es un resumen visual completo de las reacciones predefinidas del sistema ante las variadas intervenciones de los usuarios, con el fin de dar respuesta a los requerimientos que estos poseen. Para los usuarios directivos el sistema es una herramienta de control de proyectos y de toma de decisiones gerenciales, presentando las opciones pertinentes al caso. La plataforma de control macro de proyectos le permite al usuario registrar nuevos proyectos – y sus datos asociados –, modificarlos sobre la marcha y verificar el estatus y el porcentaje de avance de los mismos. Adicionalmente, se presenta la importante funcionalidad de estimación de costos operacionales a través del caso correspondiente, en la cual el usuario podrá variar las posibles asignaciones de actividades para visualizar aproximaciones de diferentes escenarios financieros.

Al directivo se le presenta también la posibilidad de crear otras cuentas de usuario de cualquier tipo dentro del entorno del sistema,

asociando datos relevantes a su funcionamiento como el costo de una hora/hombre de ese individuo para la empresa

A diferencia de la experiencia del usuario directivo en el sistema, para el coordinador la interacción se centra en las actividades en lugar de en los proyectos que las generan. Éste es informado de las tareas asociadas a los proyectos que se deben completar y procede a asignarlas a través del sistema a los individuos o grupos de trabajo. Por su parte, el sistema presenta al usuario coordinador la necesidad de especificar, previamente a cualquier asignación, los tipos de actividades que se asociarán a una clase de proyecto específico para posteriormente ofrecerlas como opciones. En esta labor el coordinador está obligado a hacer uso de su conocimiento en las tareas que se asignarán, al igual que al momento de evaluar el desempeño de los empleados comparando los tiempos reales con los estimados de una manera empíricamente subjetiva.

El usuario básico, representante de los empleados de producción, encuentra en el sistema a un gestor de trabajo, pudiendo consultar sus actividades pendientes para cualquier momento que especifique, tal y como fueron planeadas por el coordinador. Una vez terminadas las tareas el usuario puede registrar los tiempos de desarrollo, así como las posibles eventualidades que sucediesen. En caso de errores, por supuesto, el sistema ofrece la funcionalidad de modificar sobre los registros hechos previamente, con algunas restricciones de tiempo que garanticen la seguridad de la información.

A través de estos requerimientos se identificaron los casos fundamentales del sistema: **Gestionar Proyectos**, **Gestionar Actividades**, **Configurar Sistema** y la funcionalidad incluida de **Visualizar Reportes**. A continuación la documentación de cada uno:



- **Gestionar Proyectos**

**Actores:** Usuario Directivo

**Descripción:** El usuario podrá crear, modificar y concluir los proyectos de la empresa. También estará en capacidad de estimar costos operacionales en función de los recursos humanos disponibles

**Flujo de eventos:**

- El usuario selecciona la opción "*Gestión de Proyectos*"
- El sistema presenta un listado de los proyectos registrados y las opciones de creación, modificación, supervisión o estimación
- El usuario elige la opción que satisfaga la necesidad presente
- El sistema presenta un formulario de interacción
- El usuario completa el formulario y guarda los cambios
- El sistema presenta la respuesta al usuario

- **Gestionar Actividades**

**Actores:** Usuarios Coordinador y Básico

**Descripción:** El usuario Coordinador podrá asignar nuevas actividades al personal, dependiendo de cuáles proyectos se encuentren en estatus activo; esto involucrará una estimación de tiempo de ejecución. Así mismo, podrá priorizar actividades por encima de otras según lo considere necesario. Posteriormente evaluará el desempeño real en contraste con la estimación realizada.

El usuario Básico consultará y luego registrará los tiempos de ejecución de sus actividades diarias, avanzando el flujo de trabajo de los proyectos.

**Flujo de eventos:***Coordinador:*

- El usuario selecciona la opción “*Gestión de actividades*”
- El sistema presenta un listado de los proyectos activos y las opciones de asignación de actividades y evaluación del desempeño
- El usuario elige la opción que satisfaga la necesidad presente
- El sistema presenta un formulario de interacción
- El usuario completa el formulario y guarda los cambios
- El sistema presenta la respuesta al usuario

*Básico:*

- El usuario selecciona la opción “*Gestión de actividades*”
- El sistema presenta un listado de las actividades pendientes y realizadas según la fecha de consulta y presenta la opción de registro o modificación.
- El usuario elige la opción que satisfaga la necesidad presente
- El sistema presenta un formulario de interacción
- El usuario completa el formulario y guarda los cambios
- El sistema presenta la respuesta al usuario

**• Configurar Sistema**

**Actores:** Usuarios Directivo, Coordinador y Básico

**Descripción:** El usuario coordinador podrá determinar los tipos de actividades que se puedan asociar a un proyecto, y las tareas relativas a ese tipo de actividad. El Directivo podrá agregar nuevas cuentas de usuario de cualquier tipo y gestionar las variables de conocimiento de cada empleado. Todos los usuarios podrán modificar los datos propios a su acceso, tal como su contraseña y nombre de usuario

**Flujo de eventos:**

*Directivo:*

- El usuario selecciona la opción "*Gestionar Usuarios*"
- El sistema presenta un listado de los usuarios del sistema y las opciones de creación, modificación y eliminación
- El usuario elige la opción que satisfaga la necesidad presente
- El sistema presenta un formulario de interacción
- El usuario completa el formulario y guarda los cambios
- El sistema presenta la respuesta al usuario

*Coordinador:*

- El usuario selecciona la opción "*Gestionar Tipos de Actividades*"
- El sistema presenta un listado de los tipos de actividades y las diferentes tareas que han sido asociadas a cada una. También se presentan las opciones de creación, modificación y eliminación de actividades y tareas.
- El usuario elige la opción que satisfaga la necesidad presente
- El sistema presenta un formulario de interacción
- El usuario completa el formulario y guarda los cambios

- El sistema presenta la respuesta al usuario

*Básico:*

- El usuario selecciona la opción “*Gestionar Datos de Usuario*”
- El sistema demanda la contraseña de acceso actual y otorga la posibilidad de establecer una nueva
- El usuario completa el formulario y guarda los cambios
- El sistema presenta la respuesta al usuario

#### **4.2.2.2. Determinación del Diagrama de Clases de Análisis**

El siguiente diagrama permite la identificación de las clases conceptuales preliminares necesarias para posibilitar las funcionalidades descritas en los Casos de Uso. Se extrae del análisis de la estructura que se mostraría necesaria para proveer al usuario con las interfaces de las que requiere en los casos de uso, y el correcto procesamiento y almacenamiento de los datos.

Partiendo de este principio, se identificaron tres casos de uso fundamentales en los cuales se engloba el funcionamiento total del sistema; estos son la gestión de proyectos, la gestión de actividades (personales o generales dependiendo del tipo de usuario) y la configuración del sistema. Suplementariamente, se hizo presente la necesidad de establecer las facilidades de reporte del sistema y las entidades necesarias para su ejercicio. Respectivamente, el resultado de estos análisis se encuentra expuesto a continuación:

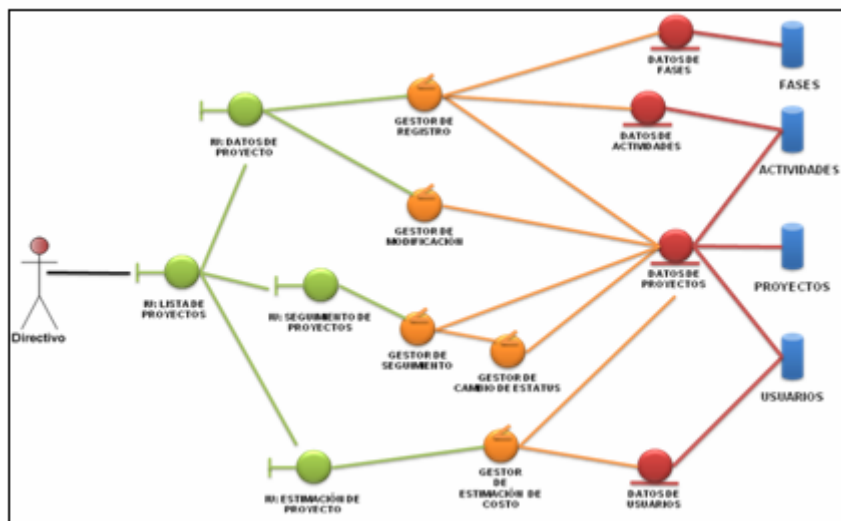
La funcionalidad principal del sistema para la toma de decisiones gerenciales viene dada por la gestión activa de proyectos; en tal sentido, se plantean las operaciones de registro, modificación y seguimiento de estatus de los proyectos para el usuario directivo, además de poder éste

en cualquier momento estimar costos relacionados con proyectos de realización futura. La **figura 4.2** ilustra el diagrama de clase de análisis de este caso.

A nivel de control operacional, el módulo básico del sistema es la gestión de actividades ejecutada por el coordinador y el control de las actividades propias ejercido por el empleado o usuario básico. Igualmente, se establecen las funciones de registro y modificación de los datos asociados como base del trabajo. Para el coordinador, por supuesto, existe la responsabilidad de la asignación de las actividades – con estimaciones asociadas – que debe realizar periódicamente, así como la evaluación posterior al resultado. La **figura 4.3** resume todos estos aspectos.

Resulta importante para un sistema como este, de datos fácilmente variables, tener una aplicación de configuración capaz de mantenerlo adaptable a las situaciones venideras. Los tipos de usuario han resultado en diferentes niveles de acceso a esta configuración, cada uno de acuerdo a sus necesidades específicas. Así, por ejemplo, el directivo está en capacidad de agregar o eliminar usuarios del sistema, mientras que el básico solo puede cambiar su propia contraseña de acceso, si cree que la previa ha sido comprometida. En la **figura 4.4** se muestra la consecuencia directa del análisis de este aspecto.

Finalmente, como se mencionó, la visualización de reportes de cualquier tipo crea la necesidad de entidades determinadas a su conformación a partir de los datos almacenados; estas se resumen en la **figura 4.5**.



Fuente: Realización propia

Figura 4.2. Diagrama de Clases de Análisis del caso "Gestionar Proyectos"

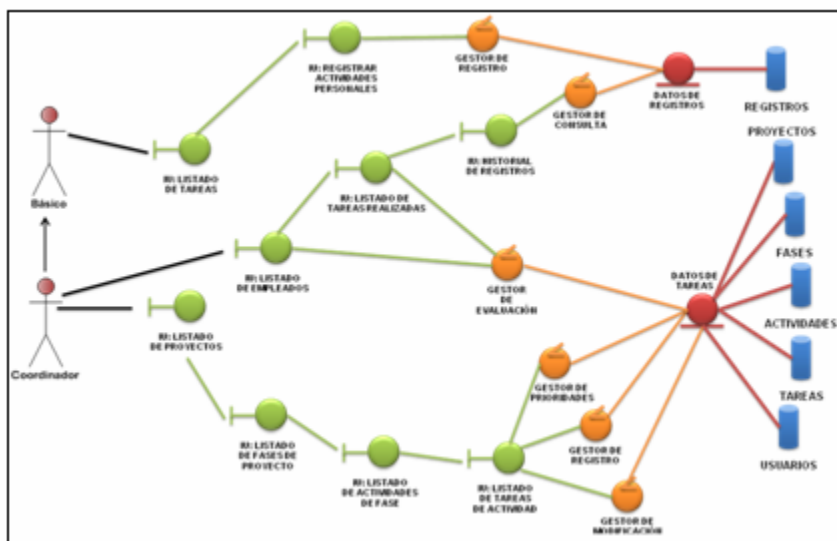
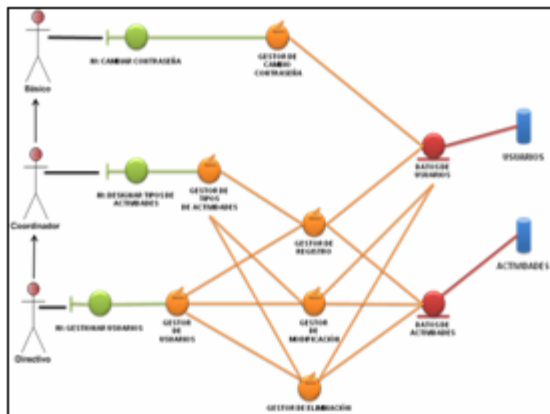


Figura 4.3. Diagrama de Clases de Análisis del caso "Gestionar Actividades"



Fuente: Ploteación propia

Figura 4.4. Diagrama de Clases de Análisis del caso "Configurar Sistema"



Fuente: Ploteación propia

Figura 4.6. Diagrama de Clases de Análisis del caso "Visualizar Reportes"

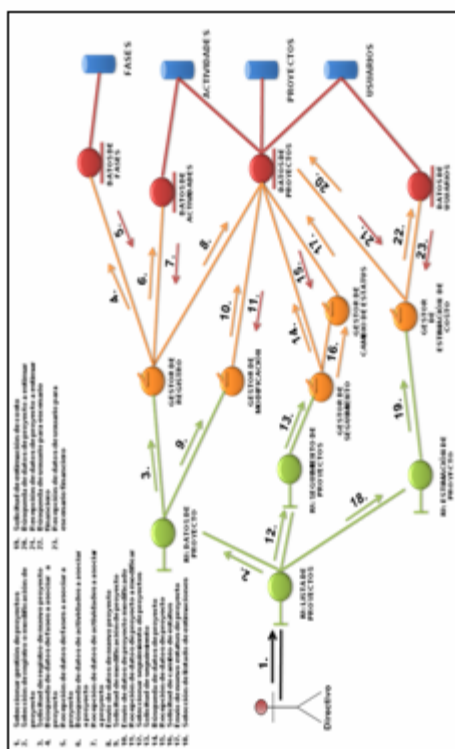
#### 4.2.2.3. Determinación del Diagrama de Colaboración del Sistema

Partiendo del diagrama de clases de análisis expuesto previamente, se procedió a detectar las interacciones entre las entidades que fueron identificadas por medio de un diagrama de colaboración. Estas reciprocidades básicas indican claramente de qué manera se deben comunicar las diferentes partes del sistema para cumplir con las tareas ligadas a cada caso de uso.

En las figuras mostradas a continuación se puede apreciar precisamente este hecho en, quedando en ellas bien definido cada intercambio con un número y una flecha directiva, así como una breve explicación de cada una:

La **figura 4.6** resume la comunicación entre objetos en el caso de uso de *gestión de proyectos*. El usuario es el activador del caso, seleccionando la opción de gestión de proyectos de la interfaz principal, solicitando esta última al gestor correspondiente la activación de la funcionalidad que el usuario elija de entre las opciones presentadas. Cada elección afecta a una entidad de datos persistentes, almacenándose estos cambios posteriormente en los almacenes de la base de datos del sistema. Estas alteraciones crearán, eliminarán o modificarán registros según sean causadas por los diferentes gestores del caso.

El mismo patrón se repite subsecuentemente en las **figuras 4.7**, **4.8** y **4.9** que corresponden a los casos *Gestionar Actividades*, *Configurar Sistema* y *Visualizar Reportes* respectivamente, con las variaciones típicas de las necesidades específicas del caso.



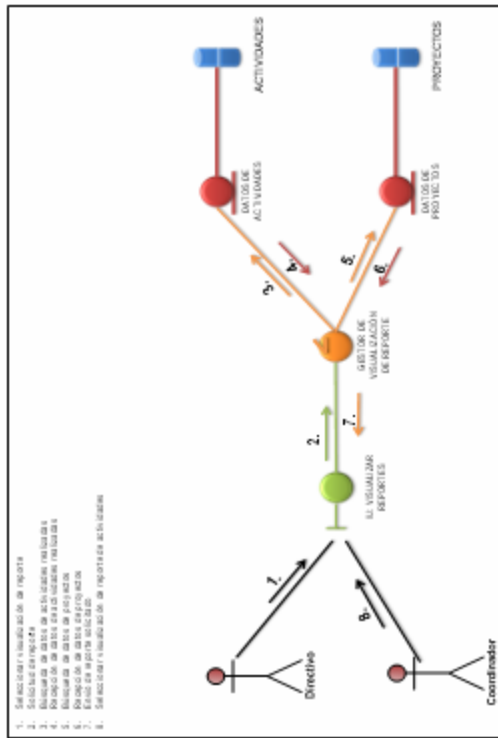
Fuente: Realización propia

Figura 4.6. Diagrama de Colaboración del caso "Gestionar Proyecto"









## CAPÍTULO 5

### 5.1. INTRODUCCIÓN

A medida que ha avanzado el estudio, la idea inicial abstracta de la “*solución tecnológica*” ha ido tomando una forma más estructurada, real, programable y aplicable a los problemas reales de la organización; aún así, sigue estando lejos de representar un modelo computacional capaz de solventar la situación presentada. Partiendo de las conclusiones alcanzadas en los análisis del apartado previo, el siguiente capítulo se centra en dar los pasos necesarios para esa transformación final de concepto a funcionalidad que se ha venido gestando desde el mismísimo planteamiento del problema. Para ello, se delinea la estructura de un software, a partir de las clases previstas, que tenga la capacidad de satisfacer los casos de uso que se hicieron evidentes durante el análisis. Así mismo, se diseña un modelo de base de datos optimizado para el desempeño ideal del sistema propuesto, basado en los espacios necesarios para el almacenamiento de la información que se deberá manejar continuamente.

### 5.2. DISEÑO DE LA ESTRUCTURA DEL SOFTWARE

Todo software que pretenda ser construido de una manera eficiente debe poseer, desde su concepción, un cuerpo, una arquitectura definida que dictamine el orden en el cual deberá organizarse su codificación interna. Sin ello, la labor de evitar la redundancia en el código, o siquiera la correcta integración del mismo, resulta en un problema de generación progresiva.

La arquitectura a desarrollar debe, por estas causas, estar orientada a la *organización* de las diferentes funcionalidades del programa de una manera lo suficientemente específica y, al mismo tiempo, abstracta para garantizar tanto la libertad al momento de programar como

la integridad de la línea base de diseño de la aplicación. El paradigma orientado a objetos (y clases) se presenta ideal para el sistema conceptualizado, permitiendo una segmentación clara de las funciones del mismo, así como de sus relaciones internas.

Una vez definido el esquema a utilizar, se procedió al desarrollo de *paquetes de clases de diseño* que englobarán a todas las clases de análisis establecidas previamente.

### **5.2.1. Diagrama de Paquetes**

El presente diagrama muestra las clases de diseño definidas para el sistema, todas ellas agrupadas en función de paquetes asociados a la función que cumplen sus elementos internos dentro de la estructura del software.

#### **5.2.1.1. Paquete SISTEMA**

El paquete *SISTEMA*, mostrado en la **figura 5.1**, representa la totalidad del software diseñado y está formado por cinco sub-paquetes distribuidos en dos capas: *capa de negocio* y de *presentación*. La *capa de negocio* contiene los cuatro paquetes que organizan la funcionalidad lógica de todos los elementos: *Seguridad*, *Estructura*, *Servicios* y *Configuración*. La *capa de presentación* contiene un único paquete, *Interfaz*. A continuación una breve explicación de cada sub-paquete de SISTEMA.

- En el paquete Seguridad se encuentra la lógica necesaria para la administración de los usuarios y roles que accederán al sistema.
- El paquete Estructura define las clases que organizarán al sistema en sus diferentes secciones
- En el paquete Configuración se encontrarán todas las clases que definen a los parámetros necesarios para el funcionamiento del

sistema, tales como las fases estándar definidas de los proyectos y sus actividades y tareas asociadas.

- Dentro de Servicios se colocarán todas las clases encargadas de llevar el registro continuo de la operación del sistema, en función de los proyectos entrantes y todos los procesos subsecuentes.

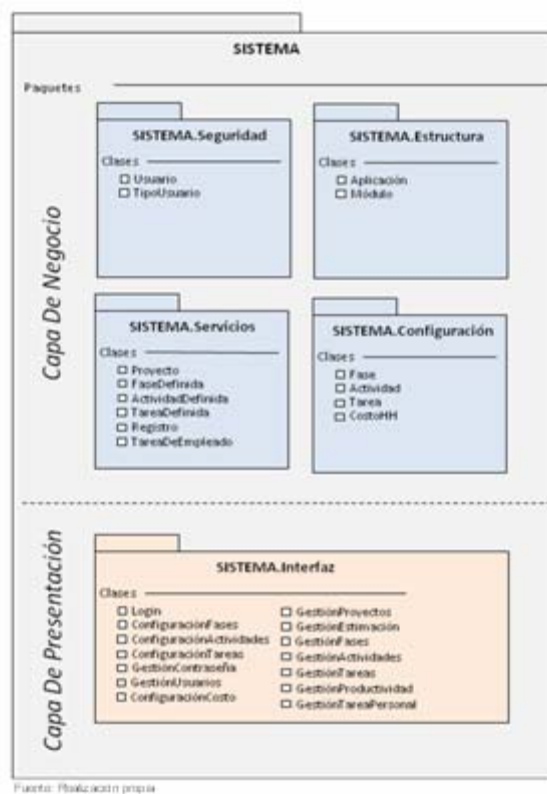
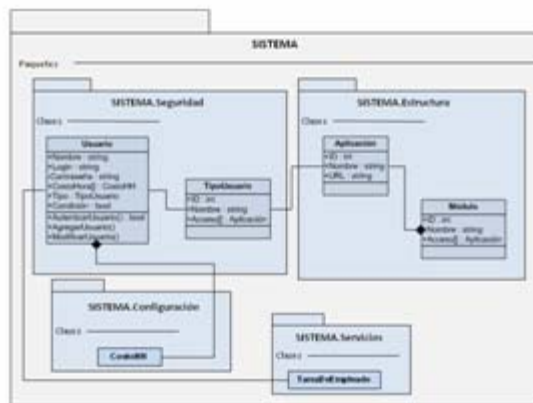


Figura 5.1. Diagrama de Paquetes SISTEMA

### 5.2.1.2. Clases de los Paquetes SISTEMA.Seguridad y SISTEMA.Estructura

Dentro de estos paquetes (**Figura 5.2**) se definen las clases *Usuario* y *TipoUsuario*, *Aplicación* y *Módulo*, las cuales describen la colección de usuarios registrados en el sistema, y las aplicaciones a las que cada uno de ellos podrá tener acceso, dependiendo del tipo de usuario que sean. De manera análoga, cada usuario deberá contener un conjunto de costos asociados a sus horas de trabajo efectivo en tareas, datos que quedan establecidos en sus relaciones con las clases *CostoHH* y *TareaDeEmpleado*. Los procesos básicos de registro de nuevos usuarios y validación de accesos se definen en los métodos de la clase *Usuario*.



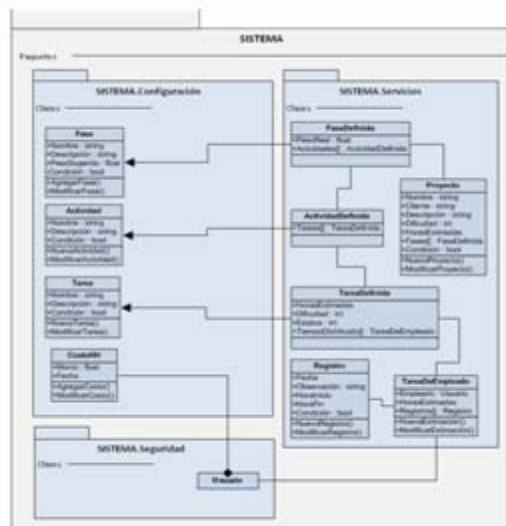
Fuente: Realización propia

**Figura 5.2.** Diagrama de Clases de Diseño para los paquetes SISTEMA.Seguridad y SISTEMA.Estructura

### 5.2.1.3. Clases de los Paquetes SISTEMA.Configuración y SISTEMA.Servicios

Las clases *Fase*, *Actividad*, *Tarea* y *CostoHH* del paquete SISTEMA.Configuración representan el centro del funcionamiento del sistema y especifican todos los parámetros iniciales de configuración del mismo, y su utilización a medida que se dan los casos de uso de las aplicaciones, por medio de los métodos básicos correspondientes. Las clases *FaseDefinida*, *ActividadDefinida*, *TareaDefinida*, por su parte, restringen a las clases generales de las cuales heredan al marco de un

proyecto específico, personificado en la clase *Proyecto*. La **figura 5.3** muestra las relaciones existentes entre las clases en detalle.



Fuente: Realización propia

**Figura 5.3.** Diagrama de Clases de Diseño para los paquetes SISTEMA.Configuración y SISTEMA.Servicios

*TareaDeEmpleado* y *Registro*, son clases que encuentran su definición en el usuario, y sirven de enlace entre los proyectos activos del sistema y los empleados que efectivamente se encontrarán trabajando en ellos.

#### 5.2.1.4. Clases del Paquete SISTEMA.Interfaz

Este paquete agrupa a todas las clases dedicadas a la *presentación* de los datos procesados al usuario por pantalla. Cada una de ellas define un conjunto de una o más interfaces dedicadas a la puesta en servicio de todos los métodos existentes en la capa de negocios del sistema. En la **figura 5.4** se expone la forma en que las interfaces toman y almacenan información de las clases internas, y brindan los accesos necesarios al usuario. De manera implícita, todas las clases de este paquete están relacionadas con todas las clases del paquete SISTEMA.Estructura,





acarrearía su utilización en el sistema; resulta particularmente importante evitar la sobrecarga por causa de la necesidad de actualización o sincronización periódica de registros.

Considerando todos estos aspectos, se tomó la decisión de recurrir al Modelo Relacional para la construcción de bases de datos, debido a que su buena documentación y simplicidad de uso se muestran óptimas para este tipo de desarrollos. No obstante, para eliminar subjetividades innecesarias en el diseño se optó, en primera instancia, por modelar el Dominio del sistema utilizando las herramientas que el UML ofrece para tal fin. Sin embargo, esta aproximación no permitió visualizar de una forma directa el estado óptimo de la base de datos, prestándose el resultado aún a diferentes interpretaciones sobre cuál era el mejor esquema a aplicar.

Ante esta circunstancia, se recurrió al diseño, previamente a cualquier aproximación a la base de datos, de un eslabón confiable entre las entidades abstractas, identificadas durante el proceso, y una base de datos relacional estructurada. Partiendo del modelo de dominio tal y como es utilizado en UML, se **creó** un nuevo diagrama más direccionado hacia la creación de tablas relacionables. Esta nueva metodología está basada gráficamente en la manera en la que se forman estructuras complejas en la materia a nivel atómico, y ha sido por tanto llamada *Estructura de Dominio del Sistema*.

### **5.3.1. Estructura de Dominio del Sistema**

El diagrama divide a cada entidad en núcleos de interés del sistema, con componentes adjuntos necesarios “orbitando” alrededor de ellos. Cada núcleo es poseedor de tres orbitales, o capas, que representan características claves de los elementos que por ellas

transitan. Esta manera de leyenda del diagrama se encuentra representada en la **figura 5.5**, presentada a continuación:

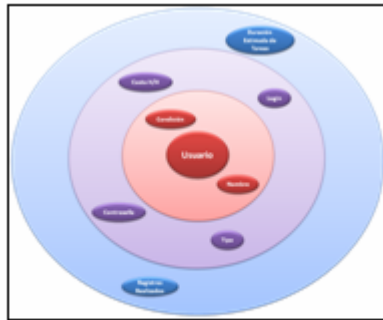


**Figura 5.5.** Leyenda de Estructura de Dominio del Sistema

De este modo, se presentan las capas que contendrán elementos asociados al núcleo, donde aquellos que se encuentren en la capa interna (roja) serán posicionados en una tabla de datos de variación poco frecuente, que definen al núcleo y son fácilmente localizados por medio de una sola clave, la clave del núcleo que orbitan.

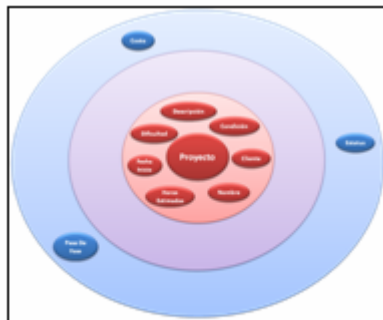
La tercera capa (azul) es la que identifica aquellos elementos asociados al núcleo que deben estar variando continuamente durante el funcionamiento del sistema, y solo pueden ser identificados a través de la utilización de dos o más claves relacionales. La capa intermedia (morada) contiene a aquellos elementos definitorios cuyas variaciones hay que tomar en cuenta tanto en el diseño de la base de datos como en las interfaces del sistema. Pueden necesitar de una o varias claves dependiendo del elemento.

En el Diagrama de Clases de Diseño, y análisis adjuntos, se identificaron los siguientes elementos claves: *usuarios*, *proyectos*, *fases*, *actividades*, *tareas*, *registros*, *estatus*, *aplicaciones* y *módulos*. Cada uno de ellos fue convertido en una Estructura de Dominio siguiendo las especificaciones expuestas previamente. Los resultados son expuestos de la **figuras 5.6** a la **5.14**, respectivamente.



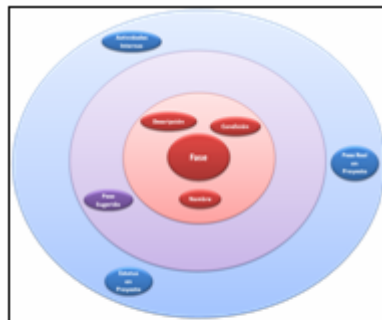
Fuente: Realización propia

**Figura 6.6. Estructura del Dominio del Usuario**



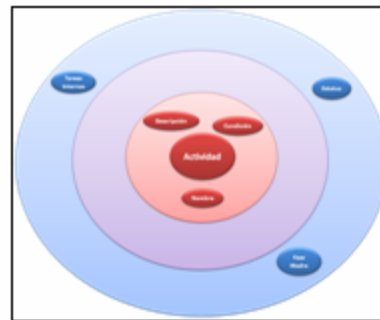
Fuente: Realización propia

**Figura 6.7. Estructura del Dominio del Proyecto**



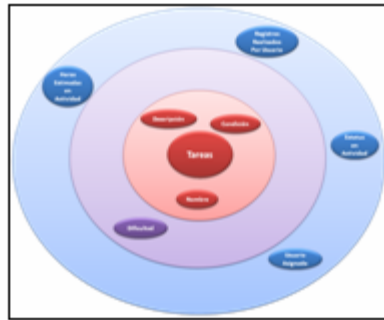
Fuente: Realización propia

**Figura 6.8. Estructura del Dominio de la Fase**



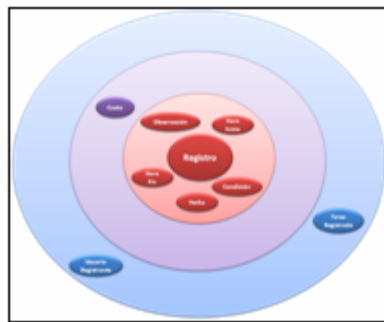
Fuente: Realización propia

**Figura 6.9. Estructura del Dominio de la Actividad**



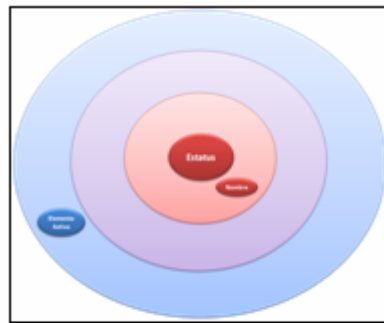
Fuente: Realización propia

**Figura 6.10. Estructura del Dominio de la Tarea**



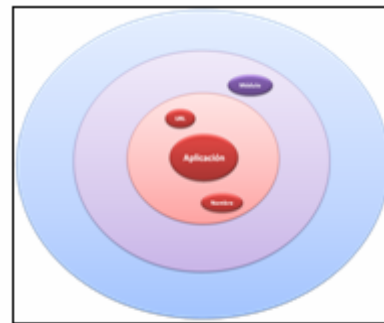
Fuente: Realización propia

**Figura 6.11. Estructura del Dominio del Registro**



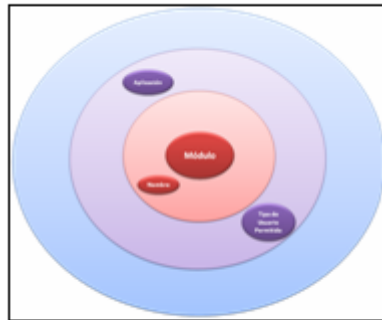
Fuente: Realización propia

**Figura 6.12. Estructura del Dominio del Estado**



Fuente: Realización propia

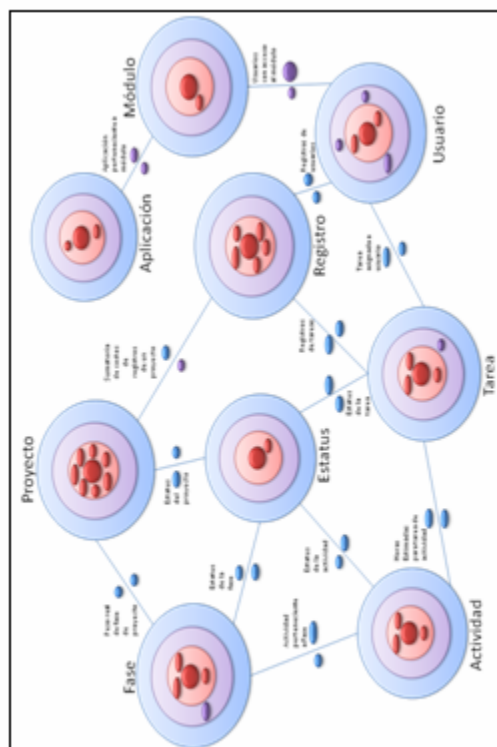
**Figura 6.13. Estructura del Dominio de la Aplicación**



Fuente: Realización propia

Figura 5.14. Estructura del Dominio del Módulo

Una vez modelado cada uno de los elementos, y formulada una idea bastante precisa de la estructura correcta de las tablas de la base de datos, se procedió a relacionar cada estructura en el carácter global del sistema en la **figura 5.15**.



Fuente: Realización propia

Figura 5.15. Estructura del Dominio del Sistema

### 5.3.2. Diseño del Modelo Relacional de la Base de Datos

Partiendo de los resultados obtenidos en la sección anterior se crearon al menos tres tablas por elemento, cada una asociada a una capa orbital. Posteriormente, estas fueron sujetas al proceso de normalización creando las claves necesarias para la obtención de las relaciones que fueron manifestadas en la estructura.

La base de datos terminó siendo conformada en esta etapa de diseño por diecinueve (19) tablas relacionadas por identificadores o “*campos clave*”. A continuación se listó una descripción un poco más precisa de cada una de estas tablas, detallando el tipo de dato que se almacenará en cada una de sus celdas y la longitud máxima permitida de caracteres.

Finalmente, se construyó el Modelo Relacional de la Base de Datos del Sistema mostrado en la **figura 5.16**. En tal modelo, cada tabla es representada por un rectángulo relacionado a otros por medio de líneas que representan la cardinalidad de la relación. (1 → N) o (1 – 1).

#### 5.3.2.1 Control de Usuarios, Seguridad, Configuración y Estructura

**Tabla 5.1. Dat\_Usuario (Datos de usuarios)**

CAMPO	TIPO DE DATO	DESCRIPCIÓN
<b>Id_Usuario</b>	Long Int	Clave principal
Nombre	String (50)	Nombre del usuario
Id_TipoUsu	Long Int	Clave foránea del tipo de usuario
Condición	boolean	Condición activa o inactiva del elemento

**Tabla 5.2. TipoUsu (Datos de tipos de usuarios)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_TipoUsu</b>	Int	Clave principal
Nombre	String (50)	Nombre del usuario

**Tabla 5.3. Log\_Usuario (Datos de validación de cuentas de usuarios)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_Usuario</b>	Long Int	Clave principal
Login	String (50)	Nombre de acceso del usuario
Contraseña	String (15)	Contraseña de usuario (campo encriptado)

**Tabla 5.4. Costo\_HH (Historial de costos asociados a empleado)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_Usuario</b>	Long Int	Clave principal compuesta
<b>Fecha</b>	Date/Hour	Fecha de entrada en vigencia del costo asociado (clave principal compuesta)
Costo	Float	Costo de la hora/Hombre del empleado



**Tabla 5.5. Dat\_Estatus (Datos de estatus posibles para tareas)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_Estatus</b>	Int	Clave principal
Nombre	String (15)	Denominación del estatus a mostrar por pantalla

**Tabla 5.6. Dat\_Mod (Datos de módulos del sistema)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_Mod</b>	Int	Clave principal
Nombre	String (25)	Denominación del módulo a mostrar por pantalla

**Tabla 5.7. Dat\_Apli (Datos de aplicaciones del sistema)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_Apli</b>	Int	Clave principal
Nombre	String (25)	Denominación de la aplicación a mostrar por pantalla

**Tabla 5.8. Apli\_Mod (Estructura de aplicaciones por módulo)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_Apli</b>	Int	Clave principal
Id_Mod	Int	Clave foránea del módulo asociado

**Tabla 5.9. Acces\_Apli (Relación entre usuarios y aplicaciones permitidas)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_TipoUsu</b>	Long Int	Identificador del tipo de usuario (clave principal compuesta)
<b>Id_Apli</b>	Int	Identificador de la aplicación permitida (clave principal compuesta)

### 5.3.2.2 Definición de Entidades

**Tabla 5.10. Dat\_Proj (Datos de proyectos)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_Proyecto</b>	Long Int	Clave principal
Nombre	String (50)	Nombre del proyecto
Cliente	String (25)	Nombre del cliente
Descripción	String (250)	Descripción breve del proyecto, aspectos clave a considerar, etc.
Dificultad	Int	Dificultad asociada al proyecto (valor entre 1 y 5)
Fecha_Ini	Date/Hour	Fecha de registro del proyecto en el sistema (guardado automático)
Horas_Est	Float	Cantidad de horas facturadas para realización del proyecto
Condición	Boolean	Condición activa o inactiva del elemento

**Tabla 5.11. Dat\_Fase (Datos de fases)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_Fase</b>	Long Int	Clave principal
Nombre	String (50)	Nombre de la fase
Descripción	String (250)	Descripción breve de la fase, aspectos clave a considerar, etc.
Peso_Sugerido	Float	Peso estándar de la fase independientemente del proyecto
Condición	Boolean	Condición activa o inactiva del elemento

**Tabla 5.12. Dat\_Activ (Datos de actividades)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_Activ</b>	Long Int	Clave principal
Nombre	String (50)	Nombre de la actividad
Descripción	String (250)	Descripción breve de la actividad, aspectos clave a considerar, etc.
Condición	Boolean	Condición activa o inactiva del elemento

**Tabla 5.13. Dat\_Tarea (Datos de tareas)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_Tareas</b>	Long Int	Clave principal
Nombre	String (50)	Nombre de la tarea
Descripción	String (250)	Descripción breve de la tarea, aspectos clave a considerar, etc.
Condición	Boolean	Condición activa o inactiva del elemento

**Tabla 5.14. Dat\_Registro (Datos de registros)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_Registro</b>	Long Int	Clave principal
Fecha	Date/Hour	Fecha de registro del elemento
Observación	String (150)	Observación breve de aspectos descriptivos a destacar en la tarea realizada
Hora_Ini	Date/Hour	Hora marcada como inicio de realización de tarea
Hora_Fin	Date/Hour	Hora marcada como final de realización de tarea
Condición	Boolean	Condición activa o inactiva del elemento

### 5.3.2.3 Relaciones entre Entidades

**Tabla 5.15. Tarea\_Asig (Tareas que han sido asignadas a un usuario)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_Usuario</b>	Long Int	Identificador del usuario (clave principal compuesta)
<b>Id_Proyecto</b>	Long Int	Identificador del proyecto (clave principal compuesta)
<b>Id_Fase</b>	Long Int	Identificador de la fase (clave principal compuesta)
<b>Id_Activ</b>	Long Int	Identificador de la actividad (clave principal compuesta)
<b>Id_Tarea</b>	Long Int	Identificador de la tarea (clave principal compuesta)
Horas_Est	Float	Cantidad de horas estimadas de trabajo en la tarea

**Tabla 5.16. Reg\_Usuario (Registros hechos por usuarios)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_Registro</b>	Long Int	Clave principal
Id_Usuario	Long Int	Clave foránea del usuario
Id_Proyecto	Long Int	Clave foránea del proyecto
Id_Fase	Long Int	Clave foránea de la fase
Id_Activ	Long Int	Clave foránea de la actividad
Id_Tarea	Long Int	Clave foránea de la tarea

**Tabla 5.17. Fase\_Proj (Fases declaradas dentro de un proyecto específico)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_Proyecto</b>	Long Int	Clave principal compuesta
<b>Id_Fase</b>	Long Int	Clave principal compuesta
<b>Peso_Real</b>	Float	Peso de la fase dentro del proyecto

**Tabla 5.18. Activ\_Fase (Actividades declaradas dentro de una fase específica)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_Proyecto</b>	Long Int	Clave principal compuesta
<b>Id_Fase</b>	Long Int	Clave principal compuesta
<b>Id_Activ</b>	Long Int	Clave principal compuesta

**Tabla 5.19. Tarea\_Activ (Tareas declaradas dentro de una actividad específica)**

<b>CAMPO</b>	<b>TIPO DE DATO</b>	<b>DESCRIPCIÓN</b>
<b>Id_Proyecto</b>	Long Int	Clave principal compuesta
<b>Id_Fase</b>	Long Int	Clave principal compuesta
<b>Id_Activ</b>	Long Int	Clave principal compuesta
<b>Id_Tarea</b>	Long Int	Clave principal compuesta
Horas_Est	Float	Cantidad de horas estimadas de trabajo en la tarea
Dificultad	Int	Dificultad asociada a la tarea (valor entre 1 y 5)
Id_Estatus	Int	Clave foránea del estatus





En los sistemas que exigen madurez organizacional para su implementación, y un cierto grado de rigidez en los procesos, es vital la correcta administración de una estrategia activa de manejo de la resistencia al cambio. Aunque la mayor parte de este proceso escapa al alcance establecido de este trabajo de grado, existe en la interfaz un elemento aprovechable en esta lucha por atraer a los nuevos usuarios al uso adecuado del sistema como recurso tecnológico de la empresa. Para ello, aunado al diseño conceptual de los elementos que serán mostrados en pantalla simultáneamente, se solicitaron los servicios de uno de los diseñadores del área de la **Coordinación de Soluciones Gráficas y Multimedia** a la directiva.

Sumados el aporte artístico del diseñador gráfico y el concepto de interfaz creado previamente, se procedió a dar forma final a las interfaces de usuario del sistema, y a bautizarlo como **SGP (*Sistema de Gestión de Proyectos*)**.

#### **5.4.1. Pantalla de Presentación del Sistema**

Esta pantalla sirve de entrada al sistema y confirma al usuario que ha ingresado la dirección correcta en el navegador. Así mismo, presenta el panel de login para validar al usuario y definir qué aplicaciones deben ser mostradas.



**Figura 5.17.** Pantalla de Presentación y Acceso al Sistema

#### **5.4.2. Pantalla de Bienvenida**

En esta interfaz se muestran las aplicaciones para las cuales el usuario posee un permiso activo, de acuerdo a su tipo, a modo de iconos en la parte superior de la pantalla.



**Figura 5.18.** Pantalla de Bienvenida

#### **5.4.3. Módulo de Seguridad**

Este módulo presentará al usuario todas las aplicaciones relacionadas con el acceso a las aplicaciones.

##### **5.4.3.1. Cambio de Contraseña**

Esta sencilla aplicación permite que el usuario modifique su contraseña de acceso al sistema, provisto que conozca la contraseña previa.



Fuente: Plataforma propia

**Figura 5.19.** Cambio de Contraseña

#### 5.4.3.2. Control de Usuarios

En esta interfaz el usuario directivo u administrador del sistema podrá crear nuevos usuarios a través de la tarea *Agregar Nuevo Usuario*, o modificar los datos de uno ya existente presionando sobre su nombre.

Al agregarse un nuevo usuario el sistema solo pedirá ingresar un login para el mismo, estableciéndose su contraseña igual al login por defecto.



Fuente: Plataforma propia

**Figura 5.20.** Control de Usuarios



**Figura 5.21.** Registro de Nuevo Usuario

#### **5.4.4. Módulo de Configuración**

Este módulo presentará al usuario todas las aplicaciones relacionadas con la definición de los parámetros iniciales necesarios para el funcionamiento del sistema, tales como fases, actividades y tareas predeterminadas.

##### **5.4.4.1. Fases de Proyecto**

En esta interfaz el usuario podrá definir las fases estándar de los proyectos de desarrollo de software de la empresa, así como su peso dentro del proceso. Se podrá agregar más fases o simplemente modificar los datos de las ya existentes.

Nombre	Descripción	Actividad Definida	Porcentaje Base	Completado
Análisis	Definición de requisitos	Definición de requisitos	10%	10%
Desarrollo	Desarrollo de software	Desarrollo de software	40%	40%
Pruebas	Pruebas de software	Pruebas de software	30%	30%
Mantenimiento	Mantenimiento de software	Mantenimiento de software	20%	20%

Fuente: Fideusoft (en proceso)

Figura 6.22. Fases de Proyecto

Fuente: Fideusoft (en proceso)

Figura 6.23. Modificación de Porcentajes Estándar de Fases

Fuente: Fideusoft (en proceso)

Figura 6.24. Registro de Nueva Fase Estándar

#### 5.4.4.2. Actividades de Fase

En esta interfaz el usuario podrá definir las actividades estándar inherentes a las fases de los proyectos de desarrollo de software. En esta etapa el peso de cada fase es distribuido equitativamente entre todas las actividades registradas. Esto se hizo para evitar el *micromanejo* excesivo del peso de cada actividad dentro del proyecto, ya que pueden llegar a ser muchas.



Nombre	Descripción	Porcentaje Sucesos	Condición
Actividad 1	Descripción 1	10%	Activo
Actividad 2	Descripción 2	40%	Activo
Actividad 3	Descripción 3	50%	Activo
Actividad 4	Descripción 4	1%	Activo
Actividad 5	Descripción 5	1%	Activo

Fuente: Plataforma propia

Figura 6.26. Actividades de Fase



Fuente: Plataforma propia

Figura 6.26. Registro de Nueva Actividad Estándar

#### 5.4.4.3. Tareas de Actividad

En esta interfaz el usuario podrá definir las tareas estándar que llevan a término una actividad cualquiera de un proyecto. Estas tareas serán las que serán asignadas a los equipos de trabajo de la empresa. El usuario

podrá crear nuevas tareas y modificar los datos de las existentes.



**Figura 5.27.** Tareas de Fase

#### 5.4.4.4. Costo de Hora Hombre

En esta interfaz el usuario podrá definir el costo que tiene para la empresa una hora de trabajo de cualquiera de los usuarios registrados, número en función de cual se calcularán los costos reales de realización de los proyectos. Cada registro del usuario será medido en horas entre inicio y fin, y éstas horas, a su vez, serán multiplicadas por el costo aquí establecido. Adicionalmente, el usuario podrá visualizar el historial de costos de un empleado cualquiera.



**Figura 5.28.** Gestión de Costo de Hora Hombre



The screenshot shows the SGP system interface. At the top, there is a navigation bar with the SGP logo and several icons. Below the navigation bar, there is a header section with the text 'Compañía y Sistema de Costos de Hora Hombre y Registro de Costos de Hora Hombre'. The main content area is titled 'Registro de Costos de Empleo' and contains a form with a 'Fecha de Pago' field and a 'Registrar' button. Below the form, there is a table with two columns: 'COSTO HORA' and 'FECHA DE ENTRADA EN VIGENCIA'. The table has three rows of data. At the bottom of the page, there is a link that says 'REGISTRAR NUEVO COSTO'.

Figura 5.28. Muestra de Costos de "Empleato"

The screenshot shows the SGP system interface. At the top, there is a navigation bar with the SGP logo and several icons. Below the navigation bar, there is a header section with the text 'Compañía y Sistema de Costos de Hora Hombre y Registro de Costos de Hora Hombre'. The main content area is titled 'Registro de Nuevo Costo de Hora Hombre' and contains a form with 'Costo Hora' and 'Fecha de Entrada en Vigencia' fields, and 'Aceptar' and 'Cancelar' buttons.

Figura 5.30. Registro de Nuevo Costo de Hora Hombre

#### 5.4.5. Módulo de Gestión de Proyectos

Este módulo presentará al usuario todas las aplicaciones relacionadas con el control macro de los proyectos activos en la empresa en un momento dado, así como las capacidades para estimar costos para posibles proyectos ficticios en función de la data recopilada.

##### 5.4.5.1. Listado de Proyectos

En esta interfaz el usuario podrá visualizar los proyectos que han sido registrados en el sistema, agregar nuevos proyectos y acceder a la información interna de cada uno de ellos, entre ella el estatus. El estatus del proyecto es completamente dinámico y será calculado en función de los estatus de cada tarea interna a él. Hay solo dos estatus predefinidos del proyecto que son independientes de sus tareas internas *No iniciado* y *Ficticio*, correspondiendo a los proyectos registrados en los que aún no se

ha trabajado y a aquellos que solo han sido estimados, mas no registrados. Los detalles de seguimiento de un proyecto podrán visualizarse presionando sobre el estatus.

Al agregar un nuevo proyecto el usuario tendrá que completar un formulario diseñado para establecer el patrón de cálculo principal de la eficiencia del equipo de desarrollo de la empresa, la dificultad. Cada proyecto dentro del ámbito del sistema poseerá una dificultad asociada, la cual será establecida automáticamente a través del esquema de *macro manejadores de dificultad*, cuyos detalles de diseño se establecen en las tablas 5.20, 5.21 y 5.22.

**Tabla 5.20. Macro Manejadores de Dificultad (Parte 1)**

<b>Las herramientas tecnológicas a utilizar son:</b>	<b>Puntuación Equivalente</b>
Nuevas y poco documentadas	5
Nuevas y bien documentadas	4
De uso poco frecuente por la empresa	3
De uso frecuente por la empresa	2
Uso estándar	1

**Tabla 5.21. Macro Manejadores de Dificultad (Parte 2)**

<b>El proyecto tiene una naturaleza:</b>	<b>Puntuación Equivalente</b>
Innovadora y relativamente complicada	5
Innovadora y relativamente sencilla	4

<b>El proyecto tiene una naturaleza:</b>	<b>Puntuación Equivalente</b>
Poco explorada previamente	3
Altamente explorada previamente	2
Completamente paquetizada	1

**Tabla 5.22. Matriz de Cálculo de Dificultad por Macro Manejadores**

<b>Naturaleza del Proyecto</b> >	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Dominio de Herramientas</b>  <b>v</b>
	A	A	B	C	C	<b>5</b>
	A	B	B	C	C	<b>4</b>
	B	B	C	D	D	<b>3</b>
	C	C	D	D	E	<b>2</b>
	C	C	D	E	E	<b>1</b>

La dificultad resultante (siendo A lo más complejo y E los más sencillo) le permitirá al usuario facturar las horas de una manera más eficiente y productiva.

Esta misma interfaz exigirá que el usuario asocie fases al proyecto que está registrando, cada una con un peso dentro del mismo, para lo cual el usuario tendrá como plantilla rápida las fases predefinidas en las aplicaciones de configuración. La misma situación se repite a nivel de las actividades pertenecientes a cada fase.



Fuente: Plataforma propia

Figura 6.31. Dashboard de Proyectos



Fuente: Plataforma propia

Figura 6.32. Planificación de un Nuevo Proyecto



Fuente: Plataforma propia

Figura 6.33. Seguimiento de Proyectos

### 5.4.5.2. Estimación de Costos de Proyectos

En esta interfaz el usuario podrá visualizar los proyectos ficticios cuyos costos han sido estimados previamente, visualizar los datos de cada uno y realizar nuevas estimaciones. Crear una nueva estimación es igual a registrar un nuevo proyecto, a excepción de que el usuario tendrá que especificar un mayor nivel de detalle. A diferencia del registro de proyectos reales, en las estimaciones el usuario especificará las tareas internas de cada actividad, así como el equipo de trabajo que desea involucrar en la misma. Como resultado, la interfaz presentará al usuario un monto en Bs.F. calculado a partir del tiempo que han tardado los empleados en realizar labores similares en proyectos con la misma dificultad.



Fuente: Plataforma propia

Figura 6.34. Usando Estimaciones



Fuente: Publicación propia

Figura 5.26. Nueva Etimación

#### 5.4.6. Módulo de Control de Actividades

Este módulo presentará al usuario todas las aplicaciones relacionadas con el manejo táctico de las actividades relacionadas con los proyectos registrados en el sistema, permitiendo asignaciones y evaluaciones de desempeño automáticas, basadas en datos empíricos.

##### 5.4.6.1. Listado de Proyectos en Proceso

En esta interfaz el usuario podrá visualizar los proyectos que han sido registrados en el sistema, cada uno con datos actualizados relativos a su avance y estatus.

La duración total del proyecto en mostrada en función de las horas facturadas, en una relación de 40 horas por semana. El tiempo consumido es una resta entre esta duración total y los tiempos sumados por medio de los registros de los empleados. A partir de los estatus de las fases del proyecto, se muestra igualmente un porcentaje de completación que puede ser rojo o verde dependiendo de si es mayor o menor al tiempo consumido, respectivamente.

Presionar sobre alguno de los proyectos traslada al usuario al listado de fases.



Fuente: Plataforma propia

Figura 5.36. Lista de Proyectos en Proceso

#### 5.4.6.2. Listado de Fases del Proyecto

En esta interfaz el usuario podrá visualizar las fases que se definieron para el proyecto que eligió previamente. Se muestran datos interesantes para la toma de decisiones tales como el número de actividades internas a la fase, las horas estimadas Vs. las invertidas, el porcentaje de completación basado en el número de actividades internas completadas y la fecha de inicio. Presionar sobre cualquier fase traslada al usuario al listado de actividades.

Nombre	Nº de Acti.	Horas Est.	Horas Inv.	% Completado	Fecha Inicio
Investigación	2	20	10	50%	2020/01/01
Diseño	4	40	30	75%	2020/01/15
Desarrollo	2	20	5	25%	2020/02/01
Pruebas	2	20	0	0%	2020/02/15
Implementación	0	0	0	0%	2020/03/01

Fuente: Plataforma propia

Figura 5.37. Lista de Fases del Proyecto

#### 5.4.6.3. Listado de Actividades de la Fase

En esta interfaz el usuario podrá visualizar las actividades que se definieron para la fase que eligió previamente. Se pueden ver los mismos datos que se mostraron en el listado de fases, pero con el mayor detalle

que permite el listado de actividades. Al igual que en los listados previos, el color de las horas invertidas (rojo o verde) muestra al usuario si estas deben ser celebradas o revisadas.



Numero	Nº de Tareas	Horas Est.	Horas Inv.	% Completado	Fecha Inicio
1	1	10	10	100%	20/01/2010
2	2	20	15	75%	20/01/2010
3	1	10	0	0%	20/01/2010
4	1	10	0	0%	20/01/2010

Fuente: Realización propia

**Figura 5.38.** Lista de Actividades de Fase

#### 5.4.6.4. Listado de Tareas de la Actividad

En esta interfaz el usuario podrá visualizar las tareas que se han registrado para la actividad que eligió. El sistema mostrará los usuarios que se asignaron, las horas estimadas totales Vs. las invertidas (aplicando el esquema de colores explicado en la sección anterior), el porcentaje de completación calculado en función de cuántos empleados han reportado su labor dentro de la tarea como terminada, el estatus (*No iniciada, En proceso, Pausada por cliente, Pospuesta, En espera de revisión, Finalizada*), entre otros. Existe en esta interfaz la posibilidad de que el usuario priorice tareas por encima de otras arrastrándolas hacia arriba o hacia abajo con el cursor en el listado. Las tareas prioritarias se mostrarán por encima de las otras al empleado asignado. Además de esto, el usuario podrá crear nuevas tareas y asignarlas siguiendo el enlace correspondiente.

Al crear una nueva tarea, al igual que con los otros elementos (Ej. proyectos, fases), el usuario contará con la plantilla rápida de las tareas predefinidas en la configuración inicial del sistema, pudiendo crear una



nueva si ninguna de las alternativas aplicase. Si la tarea es conocida, el sistema sugerirá una dificultad basada en datos empíricos, si no es así, el usuario deberá definirla por sí mismo. Para minimizar las subjetividades en esta estimación, se diseñaron para esta interfaz los llamados *micro manejadores de dificultad*, y su formulario asociado. En parte, los micro manejadores han sido extraídos del modelo COCOMO (*Constructive Cost Model*), metodología mundialmente reconocida para la estimación de costos a nivel de desarrollo de software. El modelo en general no fue considerado debido a que la empresa no posee la madurez organizacional para hacer uso efectivo del mismo. Tales evaluaciones de las capacidades organizacionales se realizaron en una serie de reuniones con la directiva gerencial y táctica de la empresa, las cuales revelaron las capacidades del modelo resultante (página siguiente) para la satisfacción de las necesidades de estimación inmediatas, y su fidelidad en tal labor.

**Tabla 5.23. Micro Manejadores de Dificultad**

<b>Aspectos a Considerar</b>	<b>Muy Baja</b>	<b>Baja</b>	<b>Normal</b>	<b>Alta</b>	<b>Muy Alta</b>	<b>PESO</b>
Experiencia general del analista	5	4	3	2	1	0.7
Experiencia con la aplicación de desarrollo	5	4	3	2	1	0.2
Complejidad del producto	1	2	3	4	5	1
Tamaño de la base de datos	1	2	3	4	5	0.2
Experiencia con el lenguaje	5	4	3	2	1	0.2

<b>Aspectos a Considerar</b>	<b>Muy Baja</b>	<b>Baja</b>	<b>Normal</b>	<b>Alta</b>	<b>Muy Alta</b>	<b>PESO</b>
Dominio de las prácticas de programación necesarias	5	4	3	2	1	0.3
Experiencia general del programador	5	4	3	2	1	0.7
Flexibilidad de las restricciones en tiempo de realización	5	4	3	2	1	0.4
Flexibilidad de las restricciones en espacio de almacenaje disponible para aplicación	5	4	3	2	1	0.2
Flexibilidad de las restricciones de tiempo de respuesta	5	4	3	2	1	0.5

Una vez elegidos en la interfaz los factores que aplican (de no aplicar serían ignorados por el usuario y excluidos del cálculo) a la tarea específica que se está registrando, el sistema calculará de manera automática, y en un nivel ajeno al usuario, la puntuación de dificultad máxima de la tarea en función de los pesos ponderados, que son factores conocidos, así como la puntuación que acarrearía un caso de máxima dificultad para cada parámetro.

El resultado real de la interacción del usuario (respuestas al formulario) sería posteriormente relacionado con esta dificultad máxima a manera de porcentaje de la misma. El porcentaje de dificultad real en

comparación con la máxima asociada a los parámetros aplicables será el determinante del Grado de Dificultad, según la siguiente tabla:

**Tabla 5.24. Determinante del Grado de Dificultad**

<b>Porcentaje</b>	<b>Grado de Dificultad</b>
0% – 20%	E
20% – 40%	D
40% – 60%	C
60% – 80%	B
80% – 100%	A

Aunado a toda esta temática, el usuario finalmente especificará las horas que estima tardará en realizarse la tarea, el equipo de trabajo asignado y el tiempo de participación de cada miembro.



Fuente: Realización propia

Figura 6.39. Utile Tareas de Actividad



Fuente: Realización propia

Figura 6.40. Registro de Nueva Tarea

#### 5.4.6.5. Evaluación del Desempeño

En esta interfaz el usuario podrá visualizar un resumen del trabajo que ha venido realizando un empleado en un mes específico, o en un rango de fechas establecido por el mismo, así como una cuenta global. El parámetro de medición de la efectividad mostrado en pantalla son los llamados *puntos de productividad*, así como las tareas realizadas en el rango, y pendientes en la fecha de la consulta.

Los puntos de productividad funcionan de manera acumulativa y están dirigidos a otorgar un mérito con criterio al trabajo realizado por los empleados, considerándose para su cálculo la dificultad establecida de la

tarea realizada y la desviación existente entre el tiempo estimado y el real. El criterio de cálculo es el siguiente:

$$[1] \quad [Dv - 1] \times Gd = Pp$$

$$[2] \quad Dv = Tr / Te$$

**Tabla 5.25. Criterio de Cálculo de los Puntos de Productividad**

<b>Grado De Dificultad (Gd)&gt;&gt;</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
<b>Tipo Desviación</b>					
<b>Desviación &lt; 1</b>	-20	-16	-12	-8	-4
<b>Desviación &gt; 1</b>	-2	-4	-6	-8	-10

Donde:

- Tr: Tiempo real de trabajo
- Te: Tiempo estimado de trabajo
- Dv: Desviación temporal
- Gd: Grado de dificultad
- Pp: Puntos de productividad

El usuario obtendrá un mayor nivel de detalle si presiona sobre algún empleado específico, apreciando las tareas que ha realizado, la dificultad de las mismas, las horas que invirtió Vs. las estimadas, la manera en la afectó su cuenta de puntos y las fechas asociadas.

Si desea verse aún más detalle, el usuario puede presionar sobre una tarea para observar los registros precisos que hizo el usuario sobre ella, incluyendo las observaciones respectivas.

Nombre	Actividades Pendientes	Actividades Comenzadas	Puntos de Productividad M1	Puntos de Productividad Otros
Andrés C.	10	0	0	0
Andrés D.	10	0	0	0
Andrés E.	10	0	0	0

Fuente: Plataforma propia

Figura 5.41. Evaluación del Desempeño

Nombre	Actividades Pendientes	Tareas	Horas Totales	Horas Eje	Horas No	Horas Productivas	Horas No Productivas	Horas Totales
Andrés C.	10	0	0	0	0	0	0	0
Andrés D.	10	0	0	0	0	0	0	0
Andrés E.	10	0	0	0	0	0	0	0

Fuente: Plataforma propia

Figura 5.42. Evaluación del Desempeño por Empleado

Fecha Requiere	Observación	Hora Inicio	Hora Fin	Duración

Fuente: Plataforma propia

Figura 5.43. Interfaz de Trabajo de Tarea

#### 5.4.7. Módulo de Control de Tareas Personales

Este módulo presentará al usuario todas las aplicaciones relacionadas con el manejo de las tareas que le han sido asignadas a través de la aplicación de creación de nuevas tareas encontrada en el módulo de control de actividades.

#### **5.4.7.1. Listado de Tareas**

En esta interfaz el usuario podrá visualizar todas las tareas en las cuales ha sido incluido como miembro del equipo de trabajo. Podrá ver las horas que ha invertido y el estatus actual de la tarea. Así mismo, al presionar sobre ella podrá ver el historial de registros que ha realizado, y crear uno nuevo si lo considera necesario. Al crear un nuevo registro, indicando que ha trabajado en la tarea, el usuario tendrá la oportunidad de reportar si ha finalizado su participación a través de un cambio de estatus. Si todos los miembros del equipo de trabajo han finalizado la tarea, ésta se visualizará como tal en los paneles administrativos, actualizando el estatus del proyecto como resultado.



Fuente: Plataforma propia

Figura 6.44. Listado de Tareas



Fuente: Plataforma propia

Figura 6.45. Nuevo Registro de Trabajo Realizado



Fuente: Plataforma propia

Figura 5.46. Nuevo Registro de Trabajo Realizado

## 5.4.8. Módulo de Reportes



Este módulo presentará al usuario todos los reportes del sistema.

#### 5.4.8.1. Reporte de Proyectos

En este reporte el usuario podrá visualizar todos los proyectos registrados en un rango de tiempo específico, el costo que han acumulado hasta la fecha, su estatus, la duración que se estimó y lo que efectivamente ha durado en realizarse.

Nombre	Cliente	Fecha Inicio	Estado	Duración Estimada (Semanas)	Duración Real (Semanas)	Costo (B/L)	Condición
Sistema de Pago en Línea	Demis	11/2017	Finalizado	6	4	850,14	Activo
Sistema de Control de Gastos de Comod	Boswell	06/2017	Finalizado	2	3	1200,12	Activo
Sistema administrativo de pago	Sant	02/2018	En Proceso	10	10	2400,20	Activo

Fuente: Realización propia

Figura 5.47. Reporte de Proyectos

#### 5.4.8.2. Reporte de Carga Laboral por Empleado

En este reporte el usuario podrá visualizar un rápido resumen del trabajo que ha sido asignado a un usuario cualquiera, apreciándose los proyectos en los que está involucrado, las tareas en las que está trabajando y aquellas que tiene pendiente por realizar. La dificultad de las tareas, las horas estimadas y las que ya se han invertido también se muestran resumidas en esta interfaz.

Nombre	Proyectos Activos	Tareas Activas (Hs. Est., Hs. Inv., DifCulad)	Tareas Pendientes (DifCulad, Hs. Est.)

Figura 5.48. Reporte de Carga Laboral por Empleado

### 5.4.8.3. Reporte Detallado de Proyectos

El detallado de proyectos resume el estatus de cada tarea definida dentro de un proyecto, organizadas por fases y actividades internas. Puede apreciarse fácilmente la fecha de inicio de las tareas, los empleados involucrados y su participación, así como el estatus.

The screenshot shows a report titled 'Reporte Detallado de Proyectos' with the SGP logo. It includes a date 'Fecha: 12/1/08' and a project name 'Proyecto: Sistema Gestión de Proyectos'. Below this is a table with the following columns: Fase, Actividad, Tarea, Fecha In, Empleado (Cm, Est, Hrs, Dnc, Costo Asociado), and Estatus. The table contains several empty rows.

Fase	Actividad	Tarea	Fecha In	Empleado (Cm, Est, Hrs, Dnc, Costo Asociado)	Estatus

**Figura 5.49.** Reporte Detallado de Proyectos

### 5.4.8.4. Reporte de Actividades de Producción

En este reporte el usuario podrá visualizar todos los registros hechos por los empleados para cada tarea en la que han tomado parte, de cada proyecto que ha sido registrado (aplican filtrados). Es el reporte más detallado de las actividades del sistema.



## CONCLUSIONES

- Se describió exitosamente el sistema de estudio a través de un análisis guiado por la metodología inicial del proceso unificado de desarrollo de software.
- Se identificaron los requerimientos por medio de una serie de entrevistas no estructuradas con la directiva de la empresa y los diferentes elementos del sistema que se convertirán eventualmente en usuarios. Para esta labor se utilizaron diagramas de análisis básicos de UML.
- Se diseñó la estructura del software a partir de diagramas de paquetes y clases de diseño basados en programación orientada a objetos.
- Se diseñó la base de datos mediante la creación de una nueva metodología de conceptualización denominada *estructura de dominio del sistema*, finalizándose el proceso en un modelo relacional normalizado.
- Se diseñaron todas las interfaces de usuario que deberá tener el sistema, así como las metodologías de cálculo que le permitirán mostrar la información requerida al usuario de manera automática y actualizada.

## RECOMENDACIONES

- **Saetha** debe considerar darle continuidad al desarrollo de este proyecto a la brevedad posible, esto debido al alto impacto que tendrá sobre sus operaciones y rentabilidad una vez operativo el sistema diseñado.
- Es importante, para mantener los requerimientos de disponibilidad, que el sistema sea desarrollado en ambiente web, tal y como fue diseñado.
- Deben utilizarse tecnologías de programación actualizadas que permitan implementar las funcionalidades más complejas aquí descritas, como el esquema de priorización móvil de tareas. Se recomiendan JAVA (portabilidad), AJAX ó ASP.NET.
- El primer trimestre de uso del sistema debe ser de recopilación de datos y prueba, y el administrador designado debe asegurarse de que se dé el uso correcto a las aplicaciones para garantizar que esta etapa crítica se lleve a cabo sin anomalías o retrasos.
- Se debe poner en marcha una campaña de concientización previa referente a los costos de producción de la empresa y a la importancia que tienen los mismos en su capacidad de crecimiento y expansión. El sistema requerirá una modificación considerable en los patrones de conducta y trabajo de los empleados y podría llegar a fracasar si este aspecto no es manejado con especial atención.
- Es recomendable esperar a mediados del segundo trimestre de vida del sistema para fiarse de los reportes que se van obteniendo a partir de él como guía en la toma de decisiones; esto se debe a la característica incremental de la eficiencia en sus operaciones de resumen de data.

- A pesar de haber sido diseñado primordialmente para coordinar desarrollos de software, el sistema puede aplicarse a cualquier tipo de proyecto de ingeniería o ciencias afines que dependan de horas hombre con relativamente pocas modificaciones a nivel de diseño. Esta es una faceta conceptual que la empresa debería considerar a mediano plazo para continuar expandiendo su línea de negocios con las necesidades del mercado a través de sus otras unidades operativas.

## BIBLIOGRAFIA

- [1]. Rodríguez, M. (2004). **“Diseño de un sistema de información para el control de los datos generales de los yacimientos oficiales de los campos petroleros de PDVSA Oriente”**. Trabajo de Grado. Ingeniería de Sistemas. Universidad de Oriente. Anzoátegui, Venezuela.
- [2]. Domínguez, N. (2005). **“Diseño de un sistema de información para la automatización de las actividades llevadas a cabo en el área de operación y almacén de una empresa de energía eléctrica”**. Trabajo de Grado. Ingeniería de Sistemas. Universidad de Oriente. Anzoátegui, Venezuela.
- [3]. Sánchez, M. (2005). **“Diseño de un sistema de información para automatizar algunas de las actividades relacionadas con el proceso de producción de crudo y gas, desde el yacimiento hasta las estaciones de flujo, que se realizan en una empresa petrolera, en Punta de Mata”**. Trabajo de Grado. Ingeniería de Sistemas. Universidad de Oriente. Anzoátegui, Venezuela.
- [4]. Cedeño, R. (2006). **“Diseño de un sistema de información para el proceso de formulación del presupuesto de inversiones de PDVSA gas distrito Anaco”**. Trabajo de Grado. Ingeniería de Sistemas. Universidad de Oriente. Anzoátegui, Venezuela.
- [5]. Rodríguez, M. (2006). **“Diseño de un sistema de información para el control interno y el manejo de proyectos en la gerencia de proyectos de una empresa consultora”**. Trabajo de Grado. Ingeniería de Sistemas. Universidad de Oriente. Anzoátegui, Venezuela.
- [6]. Puleo, F. (1980). **“Una definición de sistemas”**. Revista Sistemas, Universidad de los Andes, Venezuela.

- [7]. Mendoza, H. (2006). "Introducción a los sistemas de información" <http://www.monografias.com/trabajos36/sistemas-informacion/sistemas-informacion.shtml>.
- [8]. McKeever, J. (1973). "Sistemas de información para la gerencia". Tercera edición. Editorial Mc Graw Hill. México.
- [9]. Trejo, J. (2001). "Bases de datos" <http://www.monografias.com/trabajos11/basda/basda.shtml>.
- [10]. Harwryszkiewicz, T. (1994). "Análisis y diseño de base de datos". Tercera Edición. Editorial Megabyte. México.
- [11]. Elmasri R. y Navathe S. (2002). "Fundamentos de Sistemas de Bases de Datos", Editorial Pearson Educación, Tercera Edición, Madrid
- [12]. Kroenke, D. (1996). "Procesamiento de base de datos". Quinta Edición. Editorial Prentice Hall. México.
- [13]. Cota, A. (1994). "Ingeniería de software". Revista Soluciones avanzadas. USA.
- [14]. Jacobson, I. (1992). "*Object oriented software engineering; a use case driven approach*". Addison-Wesley Publishing Co. USA.
- [15]. Harmon, P. (1998). "Entendiendo UML: La guía del desarrollador, con una aplicación java basada en Web". Morgan-Kauffman Publishers, Inc. USA.
- [16]. Pitone D. y Pitman N. (2005) "UML 2.0 in a Nutshell". Editorial O'Reilly, Estados Unidos
- [17]. Pressman, R. (2002). "Ingeniería del software: Un enfoque práctico". Quinta edición. Mc Graw Hill. Madrid, España



- [18]. La Torre, L. (2000). “**Propuesta de métrica de perfeccionamiento de gestión de la calidad en el proceso de desarrollo de software**” <http://www.monografias.com/trabajos55/proceso-de-desarrollo-software/proceso-de-desarrollo-software.shtml>
- [19]. Sommerville, I. (1992). “**Ingeniería de software**”. Sexta edición. Editorial Addison Wesley. Madrid, España.
- [20]. Ejiogo, L. (1991). “**Software Engineering with formal metrics**”. QED Technical Publishing Group.
- [21]. Archer T. (2001). “**A fondo C#**”, Editorial MacGraw'-Hill/Interamericana de España, Madrid.

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y**

**ASCENSO:**

<b>TÍTULO</b>	"DISEÑO DE UN SISTEMA DE INFORMACIÓN PARA EL CONTROL, EVALUACIÓN Y ESTIMACIÓN DE LAS HORAS HOMBRE INVERTIDAS EN EL PROCESO DE DESARROLLO DE SOFTWARE"
<b>SUBTÍTULO</b>	

**AUTOR (ES):**

<b>APELLIDOS Y NOMBRES</b>	<b>CÓDIGO CULAC / E MAIL</b>
Pino V., Carlos J.	CVLAC: V-17.762.974 E MAIL: carlospino20@hotmail.com
	CVLAC: E MAIL:
	CVLAC: E MAIL:
	CVLAC: E MAIL:

**PALÁBRAS O FRASES CLAVES:**

Costos, Horas Hombre, Diseño de Software, Estructura de Dominio, Estimación, Control, Evaluación de Productividad, COCOMO

---

---

---

---

---

---

---

---

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:**

<b>ÁREA</b>	<b>SUBÁREA</b>
Ingeniería y Ciencias Aplicadas	Ingeniería en Sistemas

**RESUMEN (ABSTRACT):**

Durante esta investigación se diseñó un sistema Web para una empresa desarrolladora de Software; este permite llevar un control completo de las actividades de producción. A través de él se registran todos los proyectos activos de la empresa, se subdividen en fases, éstas a su vez en actividades y éstas últimas en tareas asignables a equipos de trabajo donde cada miembro es poseedor de una interfaz propia para el registro de sus actividades diarias, las cuales actualizan el estatus total del proyecto en tiempo real. Adicionalmente, la información de control recopilada es usada para estimar costos en el futuro de forma confiable, tomando en cuenta dificultad en los procesos y experiencia de los participantes. Para realizar la labor de diseño de se conceptualizaron nuevas metodologías de diagramación de entidades y cálculo de costos, así como la creación de un indicador de gestión apropiado y veraz orientado a la evaluación justa del trabajo.

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y**

**ASCENSO:**

**CONTRIBUIDORES:**

<b>APELLIDOS Y NOMBRES</b>	<b>ROL / CÓDIGO CVLAC / E_MAIL</b>				
<b>Cortínez N., Claudio A.</b>	<b>ROL</b>	<b>CA</b>	<b>AS</b>	<b>TU</b>	<b>JU X</b>
	<b>CVLAC:</b>	<b>V-12.155.334</b>			
	<b>E_MAIL</b>	<b>cl_cortinez@cantv.net</b>			
	<b>E_MAIL</b>				
<b>Carrasquero, Manuel</b>	<b>ROL</b>	<b>CA</b>	<b>AS X</b>	<b>TU</b>	<b>JU</b>
	<b>CVLAC:</b>	<b>V-7.374.987</b>			
	<b>E_MAIL</b>	<b>manuelscm@hotmail.com</b>			
	<b>E_MAIL</b>				
<b>Torrealba, Aquiles</b>	<b>ROL</b>	<b>CA</b>	<b>AS</b>	<b>TU</b>	<b>JU X</b>
	<b>CVLAC:</b>	<b>V-7385840</b>			
	<b>E_MAIL</b>	<b>taquiles@hotmail.com</b>			
	<b>E_MAIL</b>				
	<b>ROL</b>	<b>CA</b>	<b>AS</b>	<b>TU</b>	<b>JU</b>
	<b>CVLAC:</b>				
	<b>E_MAIL</b>				
	<b>E_MAIL</b>				

**FECHA DE DISCUSIÓN Y APROBACIÓN:**

<b>09</b>	<b>01</b>	<b>27</b>
<b>AÑO</b>	<b>MES</b>	<b>DÍA</b>

**LENGUAJE. SPA**

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:**

**ARCHIVO (S):**

<b>NOMBRE DE ARCHIVO</b>	<b>TIPO MIME</b>
TESIS.SaethaSGP.doc	Aplication/msword

**CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS:** A B C D E F G  
H I J K L M N O P Q R S T U V W X Y Z . a b c d e f g h i j k l m n o p q r  
s t u v w x y z . 0 1 2 3 4 5 6 7 8 9 .

**ALCANCE**

**ESPACIAL:** \_\_\_\_\_ (OPCIONAL)

**TEMPORAL:** \_\_\_\_\_ (OPCIONAL)

**TÍTULO O GRADO ASOCIADO CON EL TRABAJO:**

\_\_\_\_\_ Ingeniero en Sistemas \_\_\_\_\_

**NIVEL ASOCIADO CON EL TRABAJO:**

\_\_\_\_\_ Pre-Grado \_\_\_\_\_

**ÁREA DE ESTUDIO:**

\_\_\_\_\_ Departamento de Computación y Sistemas \_\_\_\_\_

**INSTITUCIÓN:**

Universidad de Oriente – Núcleo de Anzoátegui

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y  
ASCENSO:**

**DERECHOS**

De acuerdo con el artículo 44 del reglamento de trabajo de grado:

“Los trabajos de grado son de exclusiva propiedad de la Universidad de Oriente y sólo podrán ser utilizados a otros fines con el consentimiento del consejo de núcleo respectivo, quien lo participará al Consejo Universitario”.

**AUTOR**

**AUTOR**

**AUTOR**

**TUTOR**

**JURADO**

**JURADO**

**POR LA SUBCOMISIÓN DE TESIS**