

**UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMATICA**



PROYECTO DE GRADO

**SISTEMA WEB PARA EL CONTROL Y ADMINISTRACIÓN DE
RECURSOS HUMANOS**

CASO: EMPRESA DE LIMPIEZA INDUSTRIAL “TOTES LTDA”

**PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMATICA
MENCION: INGENIERIA DE SISTEMAS INFORMATICOS**

POSTULANTE: JHONNY RUBÉN MONRROY CASILLO

TUTOR METODOLOGICO: MSc. JORGE TERAN POMIER

ASESOR: Lic. MARCELO ARUQUIPA CHAMBI

**LA PAZ – BOLIVIA
2014**



UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA

A mis padres, Luciano y Celmira por su apoyo y comprensión en cada momento de mi formación como estudiante.

A mi Esposa Victoria, por su gran amor, comprensión y apoyo incondicional.

(Jhonny)

AGRADECIMIENTOS

Ante todo a Dios por haberme dado una oportunidad más en mi vida para poder crecer como persona y ser humano.

A mi Tutor Metodológico MSc. Jorge Teran por sus comentarios y observaciones que me permitieron culminar con el presente proyecto.

A mi Asesor Lic. Marcelo Aruquipa por su asesoría, tiempo, dedicación y guía para el desarrollo de este proyecto.

Al Gerente Administrativo de la Empresa de Limpieza Industrial “TOTE’s LTDA” Ing. Alvaro Mendez por su incondicional apoyo al permitirme desarrollar el proyecto en su institución.

Al plantel docente de la Carrera de Informática por la formación que recibí de ellos.

A mis padres por su apoyo durante mi época de formación.

A mí querida esposa Victoria por las palabras de ánimo que me brindo.

A mis verdaderos amigos que me brindaron sus consejos.

RESUMEN

Muchas empresas usan computadoras personales para administrar la documentación y datos de su personal. Una empresa de limpieza industrial no se queda al margen, emplea las hojas electrónicas de tipo EXCEL para administrar la información del personal operativo y administrativo, control de asistencia, asignación de turnos de trabajo, permisos con licencia, planillas de pago, etc.

El presente proyecto fue desarrollado en la Empresa de Limpieza Industrial “TOTE’s LTDA” específicamente en la unidad de Recursos Humanos, que realiza tareas de reclutamiento de personal, asignación de personal a los contratos a las cuales brinda servicios de limpieza, control de asistencias y elaboración de planillas de pago.

El Sistema Web para el Control y Administración de Recursos Humanos fue implementado de acuerdo a las necesidades y requerimientos de la institución, dando como resultados: reportes del estado de la documentación de cada funcionario, una mejor administración y control de las faltas y permisos del personal, informar sobre contratos en donde existen puestos vacantes, generación de planillas y almacenamiento de la documentación de cada funcionario y contrato en su formato digital para que esta pueda ser consultada de forma inmediata.

Durante la etapa de análisis y planificación se aplican las fases de la metodología de desarrollo XP y la herramienta de modelado WEBML. Por otra parte, para la etapa de codificación se utilizó como herramientas de desarrollo el lenguaje de programación Java bajo el framework Spring y el gestor de base de datos Postgres.

INDICE GENERAL

1 CAPITULO I MARCO REFERENCIAL	12
1.1 Introducción.....	12
1.2 Antecedentes	12
1.3 Planteamiento del Problema.....	15
1.4 Formulación del Problema.....	16
1.5 Justificación.....	16
1.6 Objetivos	16
1.6.1 Objetivo Central.....	16
1.6.2 Objetivos Específicos	17
1.7 Alcance	17
1.7.1 Límites.....	17
1.8 Metodología.....	17
1.8.1 Tipo de Investigación.....	17
1.8.2 Metodología de Desarrollo.....	17
1.9 Aporte.....	19
2 CAPITULO II MARCO TEORICO	20
2.1 Introducción.....	20
2.2 Marco Institucional.....	20
2.2.1 Empresa de limpieza Industrial TOTE's LTDA	20
2.3 Administración de Recursos Humanos	23
2.3.1 Administración de personal.....	23
2.3.2 Objetivos y funciones de la Administración de personal	25
2.4 Ingeniería del Software.....	26
2.5 Modelos y Metodologías de Desarrollo.....	27
2.5.1 Metodología XP	27
2.6 Ingeniería Web	32
2.6.1 Áreas	32
2.6.2 Categorías	33
2.6.3 Naturaleza multidisciplinaria	34
2.7 Metodología de modelado Web – WEBML	34
2.7.1 El Modelo estructural	35

2.7.2	Modelo de derivación.....	36
2.7.3	Modelo del hipertexto	37
2.7.4	El Proceso de Diseño	40
2.8	Herramientas para la Implementación	41
2.8.1	Java.....	42
2.8.2	PostgreSQL.....	42
2.8.3	Framework Spring	43
2.9	Calidad de software.....	46
2.9.1	ISO 9126	47
2.10	Pruebas de Software.....	48
2.10.1	Objetivos	49
2.10.2	Prueba unitaria	49
2.11	El modelo COCOMO	50
2.11.1	Modelo Básico	51
2.11.2	Modelo Intermedio.....	52
2.11.3	Modelo Detallado.....	53
	2.11.4 Validación independiente del modelo COCOMO	54
3	CAPITULO III MARCO APLICATIVO	55
3.1	Introducción.....	55
3.2	Análisis.....	55
3.2.1	Análisis del Sistema actual	55
3.2.2	Declaración de propósitos	55
3.3	Valores de la Metodología XP	55
3.3.1	Planificación incremental	55
3.3.2	Pruebas	56
3.3.3	Programación en parejas.....	56
3.3.4	Refactorización.....	56
3.3.5	Diseño simple	56
3.3.6	Propiedad colectiva del código	56
3.3.7	Integración continua	56
3.3.8	Cliente en el equipo	57
3.3.9	Pequeñas entregas.....	57
3.3.10	Semanas de 40 horas.....	57

3.3.11 Estándares de codificación	57
3.4 Fase I: Exploración.....	57
3.4.1 Historias de Usuario	57
3.4.2 Detalle de las historias de usuario	61
3.5 Fase II: Planificación	62
3.5.1 Estimaciones de esfuerzo.....	63
3.5.2 Planificación de iteraciones	63
3.6 Fase III: Iteraciones.....	64
3.6.1 Primera iteración	64
3.6.2 Segunda Iteración	75
3.6.3 Tercera Iteración	78
3.6.4 Cuarta Iteración	81
3.7 Diagrama Jerárquico del sistema	84
3.8 Modelado del sistema.....	87
4 CAPÍTULO IV MÉTRICAS DE CALIDAD	93
4.1 Introducción.....	93
4.2 Fiabilidad	93
4.3 Funcionabilidad	95
4.3.1 Completitud de la implementación funcional.....	95
4.3.2 Adecuación funcional.....	95
4.4 Usabilidad	96
4.5 Portabilidad	98
4.6 Mantenimiento.....	99
4.7 Análisis de Costos	99
5 CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES.....	102
5.1 Conclusiones.....	102
5.2 Recomendaciones.....	102
6 BIBLIOGRAFÍA	104

INDICE DE FIGURAS

Figura 1.1 Organigrama de la Empresa (Mendez, 2013)	13
Figura 2.1: Oficina central TOTE's LTDA (TOTE's, 2011)	22
Figura 2.2: Ramas de la Administración General (Marconi, 2012)	24
Figura 2.3: Definición de Administración de Personal (Marconi, 2012)	24
Figura 2.4: Funciones de la Administración de Personal (Marconi, 2012)	26
Figura 2.5: Kent Beck (Fernández, 2002)	27
Figura 2.6: El coste del cambio (Fernández, 2002)	28
Figura 2.7: Iteraciones y planes de iteración (Fernández, 2002)	29
Figura 2.8: Modelo de tarjeta CRC (Anaya)	30
Figura 2.9 Ejemplo de modelo estructural (webml.org)	36
Figura 2.10 Ejemplo de modelo de derivación (Ceballos, Arboleda, & Casallas, 2008).....	36
Figura 2.11 Ejemplo de un modelo de hipertexto (webml.org)	38
Figura 2.12 Ejemplo de Modelo de Presentación (tecnologimerlin, 2007)	40
Figura 2.13 Proceso de diseño (webml.org).....	41
Figura 2.14 Componentes de un sistema PostgreSQL (Martinez, 2010).....	43
Figura 2.15 Spring Framework Runtime (SpringFramework)	44
Figura 2.16 Gestión de Calidad (Noriega Quintana, 2007).....	47
Figura 3.1: Cronograma de desarrollo por iteraciones	64
Figura 3.2: Estructura de datos para registrar la información del usuario y funcionario.....	64
Figura 3.3: Pantalla de búsqueda de funcionarios para posterior registro como funcionario .	65
Figura 3.4: Listado de funcionarios encontrados.....	65
Figura 3.5: Mensaje de que no se puede registrar como usuario porque no se trata de un funcionario activo	66
Figura 3.6: Pantalla de registro de usuario en caso de que el funcionario este en estado activo	66
Figura 3.7: Estructura de datos para registrar la información del funcionario	67
Figura 3.8: Pantalla de inicio para verificar la existencia de un funcionario	68
Figura 3.9: Pantalla para el registro de información del funcionario	68
Figura 3.10: Pantalla de anuncio de que el funcionario tiene datos almacenados.....	69
Figura 3.11: Pantalla para lanzar el módulo de re afiliación del funcionario	69
Figura 3.12: Modulo para la re afiliación del funcionario	70
Figura 3.13: Estructura de datos para registrar la información de un contrato	71
Figura 3.14: Modulo para el registro de un nuevo contrato	71
Figura 3.15: Modulo para el llenado de datos de un nuevo contrato	72
Figura 3.16: Modulo para el registro de un nuevo contrato pantalla de contrato registrado...	72
Figura 3.17: Pantalla de búsqueda de un contrato cerrado.....	73
Figura 3.18: Pantalla de listados de contratos cerrados.....	73
Figura 3.19: Pantalla de interfaz de apertura de un contrato.....	74
Figura 3.20: Pantalla de aviso de que el contrato fue abierto correctamente	74
Figura 3.21: Diseño de la tabla para registro de documentos del funcionario	75
Figura 3.22: Pantalla de listado de documentos del funcionario.....	76

Figura 3.23: pantalla para subir y almacenar un documento en el sistema	76
Figura 3.24: Pantalla de descarga de documentos digitalizados	77
Figura 3.25: Diseño de la tabla para el control de las asistencias de un funcionario	78
Figura 3.26: Pantalla delo listado de asistencias registradas de los funcionarios de un contrato.....	78
Figura 3.27: Diseño del formulario para el registro de asistencias de los funcionarios que trabajaron en un respectivo contrato.....	79
Figura 3.28: Pantalla de inicio mostrando el estado del registro de las asistencias de los contratos supervisados	80
Figura 3.29: Modulo del llenado de datos de las asistencias para un contrato en particular..	80
Figura 3.30: Estado del listado de asistencias después del registro de datos	81
Figura 3.31: Diseño preliminar del reporte de la planilla por contratos	82
Figura 3.32: Modulo para generar planillas por contratos	82
Figura 3.33: Reporte de la planilla del mes	83
Figura 3.34: Diagrama jerárquico del sistema	84
Figura 3.35: Diagrama de clases del modelo	85
Figura 3.36 Modelo estructural	87
Figura 3.37 Modelo de derivación (Registrar Usuario)	88
Figura 3.38 Modelo de derivación (Registro de un funcionario)	88
Figura 3.39 Modelo de derivación (registrar contrato)	89
Figura 3.40 Modelo de derivación (Registro de documentación de un funcionario).....	89
Figura 3.41 Modelo de derivación (Registro de asistencias)	90
Figura 3.42 Modelo de derivación (Generación de planillas).....	90
Figura 3.43 Modelo de hipertexto (Administrador)	91
Figura 3.44 Modelo de hipertexto (Jefe de Recursos Humanos).....	91
Figura 3.45 Modelo de hipertexto (Supervisor)	92
Figura 3.46 Modelo de Presentación	92

INDICE DE TABLAS

Tabla 2.1: Modelo propuesto para una historia de usuario (Anaya)	28
Tabla 2.2 Coeficientes (Dolado).....	51
Tabla 3.1 Historia de Usuario - registro de un nuevo Usuario	58
Tabla 3.2 Diseñar la estructura de datos para registrar la información del usuario y funcionario	58
Tabla 3.3 Tarea - Registro de datos del usuario	59
Tabla 3.4 Tarea - Habilitación de solo funcionarios activos para ser asignados como usuarios	59
Tabla 3.5 Historia de usuario - Registro de funcionarios	60
Tabla 3.6 Diseño de la base de datos para la información del funcionario	60
Tabla 3.7 Tarea - Modulo de registro de información de un funcionario	61
Tabla 3.8 Tarea - Módulo de re afiliación de un ex funcionario	61
Tabla 3.9 Administración de usuarios	63
Tabla 3.10 Administración de funcionarios.....	63
Tabla 3.11 Administración y control de asistencias.....	63
Tabla 3.12 Generación de reportes de planillas	63
Tabla 3.13 Cronograma en detalle de las historias de usuario.....	63
Tabla 3.14 Acceso de los usuarios a los diferentes módulos del sistema	85
Tabla 4.1 Fiabilidad	93
Tabla 4.2 Análisis de la fiabilidad	94
Tabla 4.3 Fiabilidad a la finalización del proyecto	94
Tabla 4.4 Fiabilidad a la finalización del proyecto	94
Tabla 4.5 Análisis de la funcionalidad	96
Tabla 4.6 Mejora de la funcionalidad a la finalización del proyecto	96
Tabla 4.7 Análisis de los datos de la usabilidad	97
Tabla 4.8 Mejora en la usabilidad del sistema	98
Tabla 4.9 Análisis de los datos de portabilidad	98
Tabla 4.10 Análisis de los datos de portabilidad a la finalización del proyecto	99
Tabla 4.11 Resumen de Métricas de Calidad	99
Tabla 4.12 Conteo estimado del Total de líneas de código durante el desarrollo.....	100
Tabla 4.13 Costo del proyecto	101
Tabla 6.1 Historia de usuario - Registro de contratos.....	107
Tabla 6.2 Tarea - Diseño de la base de datos para la información de un contrato	107
Tabla 6.3 Tarea - Modulo de registro de información de un contrato	107
Tabla 6.4 Tarea - Modulo de re apertura de un contrato cerrado	108
Tabla 6.5 Historia de usuario - Verificar el estado de la documentación de un funcionario .	108
Tabla 6.6 Tarea - Diseño de la base de datos para la información de la documentación de un funcionario	109
Tabla 6.7 Tarea - Modulo de registro de documentos del funcionario	109
Tabla 6.8 Tarea - Modulo de descarga de documentos digitalizados.....	110
Tabla 6.9 Historia de usuario - Verificar el estado de las asistencias de los funcionarios....	110

Tabla 6.10 Tarea - Diseño de la base de datos para la información de las asistencias de un funcionario	111
Tabla 6.11 Tarea - Modulo de verificación de asistencias de los funcionarios por contratos y general	111
Tabla 6.12 Historia de usuario - Registrar asistencias de los funcionarios	111
Tabla 6.13 Tarea - Diseño del formulario para el registro de las asistencias.....	112
Tabla 6.14 Modulo de verificación de asistencias de los funcionarios por contrato y general	112
Tabla 6.15 Historia de usuario - Generar reportes de planilla de sueldos por contrato.....	113
Tabla 6.16 Tarea - Diseño del reporte de la planilla por contratos	113
Tabla 6.17 Tarea - Modulo de generación de reportes de planilla de sueldos.....	114

1 CAPITULO I

MARCO REFERENCIAL

1.1 Introducción

Las Tecnologías de la Información están transformando las actividades económicas y cotidianas como uno de los fenómenos sociológicos más importantes del siglo. Indiscutiblemente, las computadoras han invadido ya todos y cada uno de los campos de la actividad humana: ciencia, tecnología, arte, educación, recreación, administración, economía y de acuerdo a la tendencia actual, nuestra civilización y las venideras dependerán cada vez más de estos "cerebros" electrónicos.

Se ha venido acelerando la velocidad de cambio del medio de casi todas las organizaciones, de allí que éstas necesiten ahora más información como soporte a la toma de decisiones. Es por eso que, el desarrollo de sistemas para el manejo de información viene jugando un papel importante y cada vez más preponderante para poder competir y subsistir en el medio.

Muchas empresas usan computadoras personales para administrar la documentación y datos de su personal. Más aun ahora que existen empresas de servicios que brindan sus servicios a terceros, cuentan con bastante personal con diferentes cargas horarias y con mucho material de trabajo que administrar. Toda esa información es comúnmente almacenada en hojas electrónicas de tipo EXCEL.

Una empresa de limpieza industrial que presta servicios de limpieza con personal calificado no se queda al margen, emplea las hojas electrónicas de tipo EXCEL generadas por un programa de computador, en diferentes actividades como es el de administrar la información del personal operativo (personal de planta u obreros) y administrativo, control de asistencia, asignación de turnos de trabajo, permisos con licencia, planillas de pago, etc.

Se ve la necesidad de automatizar todos estos elementos de manera que se pueda saber con exactitud la disponibilidad de dichos elementos sin estar recurriendo solo a hojas electrónicas.

1.2 Antecedentes

La empresa de limpieza industrial “TOTES LTDA” presta sus servicios a instituciones públicas y privadas, Totes La Paz empezó sus operaciones el 10 de abril de 1986, La primera sucursal de Totes se inauguró el 30 abril de 1993 en la ciudad de Cochabamba. Viendo el éxito de esta, el 12 abril de 1997 se abren las puertas de Totes Santa Cruz. Años después se abre Totes Sucre (1ro de Octubre, 2002) y Totes Tarija (8 de agosto, 2003) atendiendo también a Potosí y Oruro.

Tiene como Misión:

“Brindar limpieza 100% garantizada a todos sus clientes.” (TOTEL's, 2011)

Tiene como Visión:

“Ser la empresa líder del mercado de limpieza nacional, contando con reconocimiento internacional, donde cada cliente encuentre las soluciones que necesita y su personal encuentre un lugar donde crecer, todo esto en un marco de respeto al medio ambiente.” (TOTE's, 2011)

La Empresa de Limpieza Industrial TOTES.LTDA Cuenta con un personal de aproximadamente 500 operarios de planta distribuidos en cada uno de los subcontratos que tienen con instituciones públicas y privadas en cada departamento del país.

Es una institución que brinda sus servicios de limpieza industrial con personal calificado a nivel nacional, alguno de estos servicios que presta son:

- Lavado de alfombras.
- Lavado y pulido de pisos fríos.
- Limpieza de vidrios interiores y exteriores.
- Desengrasado de cocinas.
- Lavado de tapices y colchones.
- Limpieza en seco de alfombras.
- Desinfectado y sanitizado de baños.
- Lavado profundo de interior de automóviles.
- Ozonizado de ambientes.
- Extracción de agua en caso de inundaciones.
- Mantenimiento general y desempolvado de oficinas.
- Servicio de jardinería, cafetería y mensajería.
- Fumigado de ambientes.

El jefe de recursos humanos depende directamente del gerente regional, cada regional cuenta con supervisores de contratos que dependen del jefe de recursos humanos, a la vez cada supervisor controla a los operarios de los distintos subcontratos, en cada subcontrato hay un jefe de grupo y un número de operarios por contrato.

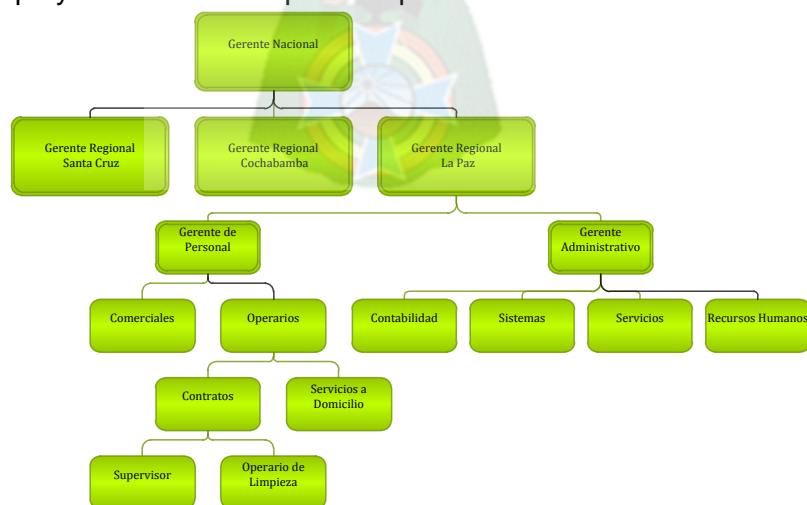


Figura 1.1 Organigrama de la Empresa (Mendez, 2013)

La empresa de limpieza industrial “TOTES LTDA” está compuesta por varias áreas, una de ellas es la de recursos humanos en la que se realizan actividades como ser:

- Reclutamiento de personal.
- Capacitación de personal en el área de limpieza industrial.
- Administración de la documentación del personal.
- Asignación del personal a determinadas áreas de trabajo.
- Supervisión y control de asistencia del personal
- Elaboración de planillas de pago.

Proyectos Similares

La temática de control y administración de Recursos humanos es tratada en algunos proyectos de grado. Mencionándose a:

- Sistema biométrico dactilar para el control de personal y planillas de pago.

(Llanque, 2009)

En la unidad educativa Villa Tunari, el control de asistencia se la realiza de manera manual. El problema central reside en que no existe un sistema automatizado para el eficiente control del personal docente y administrativo lo cual dificulta tener la información pertinente y oportuna para la adecuada toma de decisiones.

Para la elaboración del marco aplicativo se emplea la metodología RUP y el lenguaje de modelado UML. Las herramientas utilizadas para el desarrollo son el gestor de base de datos MySQL y el lenguaje de programación Visual Basic 2008 Express.

- Sistema de administración de personal Instituto Nacional de Reforma Agraria Dirección Nacional de Recursos Humanos.

(Delgado, 2008)

El presente proyecto desarrollado con el tema “Sistema de Administración de Personal Instituto Nacional de Reforma Agraria”, que da respuesta a la carencia de información oportuna y precisa acerca del control, seguimiento de documentación y evaluación del personal y los diferentes trámites que se generan y realizan al interior de la unidad a nivel nacional y que son depositados y custodiados por la unidad de archivos.

Para el proceso de desarrollo del sistema se utilizó la metodología OHDM, el lenguaje de modelado UML, en cuanto a la implementación se empleó el motor de base de datos MySQL.

- Sistema de administración de personal "Unidad de talento humano Hospital Agramont."

(Nina)

El presente proyecto fue desarrollado en el Hospital Agramont. La Unidad de Talento Humano no cuenta con un sistema automatizado y estos procesos se los realiza de forma manual ocasionando volúmenes de papelería con información muy importante a la intemperie.

Para el desarrollo del sistema se utilizó la metodología RUP y se apoya en UML, la herramienta de desarrollo fue SQL Server 2000, Cristal Reports 2003 y Visual Basic .NET 2005

- Sistema de administración de Recursos Humanos CASO: File Archivos S.A.

(Mamani, 2011)

La función de los Recursos Humanos ha evolucionado desde una concepción eminentemente administrativa, en la que lo fundamental era la confección de nómina, establecimiento y mantenimiento de un sistema de administración de RRHH.

Durante la etapa de análisis se realiza y se aplica las fase de la metodología XP aplicando el diseño con UML así también para la parte navegacional se usa OOHDM. Por otra parte la codificación se realizó con el lenguaje PHP, Servidor Apache y gestor de base de datos MySQL.

- Sistema de administración de recursos humanos para el CEMSE.

(Chávez, 2007)

En la institución es fundamental administrar los datos de los funcionarios de acuerdo a los requerimientos del usuario. Por esta razón se realiza el Sistema de Administración de Recursos Humanos para el CEMSE, basado en computador con el propósito de reducir el tiempo de procesamiento de los registros personales y la obtención de información confiable, segura e integra para un mejor servicio a la institución y al personal.

Se hace uso de Relationship Management Methology – RMM basado en enfoque estructurado y navegacional de un documento de hipertexto. En la implementación se hace uso del lenguaje PHP, para el gestor de base de datos se hace uso de MySQL.

1.3 Planteamiento del Problema

Debido al volumen de información que genera el departamento de Recursos Humanos de la Empresa de Limpieza industrial TOTES LTDA, se pueden identificar los siguientes problemas:

- Actualmente la administración de la documentación del personal se la realiza de dos formas: de manera manual registrando todos los documentos en un archivo por persona y mediante un programa que genera hojas electrónicas EXCEL.
- La manipulación manual no es muy segura ni confiable, ya que el personal encargado de administrar estos documentos puede incurrir en las siguientes irregularidades: no notificar al personal afectado que debe actualizar sus documentos (por ejemplo certificado de Antecedentes, fotocopia de CI actual, fotografía actual, etc.), si un personal quiere retornar a la empresa, verificar si

antes no trabajo o ver si no tenía antecedentes de mala conducta en contra de la empresa, descuentos injustificados, etc.

- El proceso de reclutamiento de personal es manual, recolectando la documentación en forma no organizada lo que conlleva al retraso en la asignación de áreas de trabajo a los nuevos operarios.
- Frecuentemente hay llamadas de atención contra la empresa por la seguridad de los ambientes de trabajo en los subcontratos, esto se debe a que los certificados de antecedentes del personal no son actualizados periódicamente. Y no se sabe cuándo deben actualizar sus datos y quienes aún no completaron la documentación de garantías personales.
- El control de asistencias solo se realizan de acuerdo a una pre-planilla que realiza el supervisor de contrato, no pudiendo constatarse el cumplimiento de los horarios de cada operario.
- La emisión de memorándums es manual y se entregan con un retraso de varios días lo que implica una informalidad para los operarios y los supervisores.

1.4 Formulación del Problema

¿El Sistema Web para el Control y Administración de Recursos Humanos permitirá la automatización de la administración de documentación, control de asistencias, permisos del personal, elaboración de reportes diarios y emisión de memorándums al departamento de Recursos Humanos de la empresa de limpieza industrial “TOTES LTDA”?

1.5 Justificación

La ejecución del proyecto optimizará el control de personal en forma automática, logrando con esto reducir la intervención de recurso humano para realizar la elaboración de planillas de la empresa para la cual se realiza el proyecto, cuenta con los equipos con las características mínimas necesarias para la implementación del sistema. El software a utilizar en el desarrollo del proyecto no necesita licencia, por lo que es más económico al momento de la implementación. El sistema será desarrollado en un lenguaje de programación orientado a objetos basados en Web, con un gestor de base de datos fiable y estarán sobre un servidor con un sistema operativo Linux, los mismos que tienen la característica de ser software libre.

Con el presente proyecto el área de Administración de recursos humanos podrá brindar rápidamente informaciones solicitadas por la gerencia y los funcionarios, así de esta manera la empresa de limpieza industrial TOTES LTDA brindara un mejor servicio a la sociedad.

1.6 Objetivos

1.6.1 Objetivo Central

Desarrollar un Sistema Web para el Control y Administración de Recursos Humanos para la Empresa de Limpieza Industrial “TOTES LTDA” con base en un modelo adecuado de Ingeniería de Software.

1.6.2 Objetivos Específicos

- Mantener la información actualizada de la documentación del personal como ser certificados de antecedentes, documentos de contratación, documentos del garante, etc.
- Realizar el control de los permisos y faltas del personal operativo y administrativo.
- Automatizar la emisión de reportes, informes e historial de asistencias del personal operativo y administrativo, informes sobre el estado de los contratos y el personal existente y faltante en cada una de las instituciones a las que se brinda el servicio de limpieza industrial.
- Diseñar e Implementar una base de datos para el manejo de la información referente a los recursos humanos, implementar procedimientos almacenados para el mejor manejo de los datos en el gestor de base de datos.

1.7 Alcance

Los alcances del Sistema Web para el Control y Administración de Recursos Humanos serán:

Para la unidad de Administración de personal, se consideraran las siguientes funciones:

- Registro de Información del personal.
- Generar reportes de movimientos de los usuarios como: cambios de turnos, inasistencias, retardos, documentación faltante, etc.
- Elaboración de planillas de pago.
- Obtener reportes de asistencia.
- Generará reportes de la hora de entrada y salida del personal de acuerdo al horario asignado.

1.7.1 Límites

El sistema Web tendrá las siguientes limitaciones:

- El sistema Web no se enfocara en el área de almacenes.
- El sistema Web no cumplirá tareas del área de contabilidad.

1.8 Metodología.

1.8.1 Tipo de Investigación

El tipo de investigación es mixta (cuantitativa y cualitativa), ya que comprende la descripción, registro, análisis e interpretación de la naturaleza actual y la finalidad es presentar una interpretación correcta, del objeto de estudio.

1.8.2 Metodología de Desarrollo

El presente proyecto usara una metodología basada en la metodología de Desarrollo Ágil de Programación Extrema XP (eXTREME PROGRAMMING) pero esta será adecuada para poder ser aplicada para propósitos de este proyecto de grado y así cumplir con los plazos establecidos por la empresa TOTES LTDA.

1.8.2.1 Para la Planificación

En el ciclo de vida de eXtreme Programming la primera fase es la exploración, fase en la que se planean las **historias de usuario**, al mismo tiempo se **familiariza con las herramientas, tecnología y prácticas** que se utilizaran en el presente proyecto. Se probara la tecnología y se explorara las posibilidades de la arquitectura del sistema construyendo un **prototipo**.

1.8.2.2 Para el Diseño

Se basara en la elaboración de las cartas CRC basándose en las historias de usuario, se identificaran las relaciones entre clases y se elaborara el diagrama de clases para luego diseñar el modelo de base de datos.

1.8.2.3 Para la parte de Codificación

En metodologías pesadas, la codificación es un proceso al cual solo se llega después de largas fases de análisis y diseño de las que queda una gran cantidad de documentación a partir de la cual el proceso de codificación es relativamente sencillo. En XP el proceso es muy diferente. Prácticamente desde un principio se inicia con la codificación, favoreciendo el logro del objetivo de estar haciendo entregas frecuentes al cliente.

Se implementara una estrategia de comunicación distinta a la que propone el XP en la cual el programador podrá llamar por vía telefónica al cliente en el momento que se requirieran solucionar cualquier duda en el proceso de implementación.

Para el caso de programación en parejas solo se contara con un programador que estará totalmente involucrado en el proyecto.

1.8.2.4 Pruebas:

Se plantea realizar pruebas para comprobar el funcionamiento de los códigos que se vayan a implementar en el sistema.

- Se crearan las aplicaciones que realizarán los test con un entorno de desarrollo específico para cada prueba.
- Se someterá a pruebas las distintas clases del sistema omitiendo los métodos más triviales.
- Se crearan las pruebas que pasarán los códigos antes de implementarlos

Para el modelado, se usara la metodología de modelado **WebML: Web Modeling Language** que es un modelado orientado a aplicaciones con un uso intensivo de datos, donde hay gran cantidad de datos, con estructura compleja y las aplicaciones tienen que acceder a ellos. Esta metodología se divide en 4 fases:

- **Modelado de datos:** El modelo de datos representa las diferentes tablas de datos y sus relaciones que son necesarias para una aplicación Web concreta. Se pueden utilizar:
 - Diagramas de Entidad-Relación (E-R) que muestran todas las tablas, los diferentes campos de cada tabla, y las relaciones entre ellas.

- Diagramas UML de clases que pueden representar la misma información que un diagrama de Entidad-Relación (por lo que puede usarse de manera equivalente), e incluso información adicional sobre el modelo de datos.
- **Modelo de Hipertexto:** Un modelo por cada hipertexto, cada hipertexto describe una vista del sitio.
- **Modelo de composición:** Representa las páginas de un hipertexto y cada página que elementos de contenido tiene.
- **Modelo de navegación:** Representa los enlaces entre las diferentes páginas y sus elementos de contenido

En el presente proyecto se tomará muy en cuenta el **paradigma de programación orientada a objetos**.

La orientación a objetos ha tomado por asalto y en forma legítima al mundo del software. Como medio para la generación de programas, tiene varias ventajas. Fomenta una metodología basada en componentes para el desarrollo de software, de manera que primero se genere un sistema mediante un conjunto de objetos, luego se podrá ampliar el sistema agregándole funcionalidad a los componentes que ya habría generado o agregándole nuevos componentes, y finalmente podrá volver a utilizar los objetos que generó para el sistema cuando cree uno nuevo, con lo cual reducirá sustancialmente el tiempo de desarrollo de un sistema.

La orientación a objetos se refiere a algo más que tan sólo atributos y acciones; también considera otros aspectos. Dichos aspectos se conocen con *abstracción, herencia, polimorfismo y encapsulamiento o encapsulación*. Otros aspectos importantes de la orientación a objetos son: el *envío de mensajes, las asociaciones y la agregación*.

1.9 Aporte

El Sistema web para el Control y Administración de Recursos Humanos muestra cómo se puede realizar el control de las asistencias del personal dispersado en diferentes contratos mediante relojes biométricos en contratos grandes y bajo reportes de asistencia de los supervisores en contratos pequeños que al combinarlos permiten la reducción del tiempo de acceso de la información y con esto se posibilite la ejecución de otras tareas en el departamento de Recursos Humanos.

2 CAPITULO II

MARCO TEORICO

2.1 Introducción

Realizar un Sistema Web para el Control y Administración de Recursos Humanos requiere del conocimiento de metodologías de desarrollo que utilicen modelos y estructuras formales de diseño e implementación de software, que ayuden a la programación de tareas antes de la construcción del sistema, obtener los requerimientos del usuario y organizar cada una de las etapas de entrega de prototipos y pruebas del sistema, esto con el fin de mantener un cuidadoso manejo de la información de los funcionarios y cumplir con las expectativas del usuario final que hará uso de los módulos del sistema más concretamente los módulos de administración y control de recursos humanos.

Un proceso de desarrollo de software es una definición del conjunto complejo de actividades necesarias para convertir los requisitos de usuario en un conjunto de programas que resuelvan estas necesidades y conformen al final un producto que en este caso es El Sistema web para el control y administración de recursos humanos.

En este capítulo se describirá los conceptos referentes a la metodología que se usara como línea referencial en el proceso de desarrollo de software, por otro lado también se detallará los conceptos básicos del patrón modelo vista controlador, descripción del mapeo objeto/relacional, es decir como a partir de un modelo de datos orientado a objetos se consigue su almacenamiento en una base de datos relacional; para ello se hará el uso del framework spring y sus propios métodos para las consultas a la base de datos JDBC, que precisamente nos permitirá esta conversión y principalmente el desarrollo de la aplicación netamente orientado a objetos sin la necesidad de preocuparse por la conversión manual para el almacenamiento en la base de datos relacional.

Finalmente se describirá brevemente los parámetros y conceptos relacionados con la calidad de software, además se definirán criterios de calidad con los que contará el sistema.

2.2 Marco Institucional

2.2.1 Empresa de limpieza Industrial TOTE's LTDA Historia de la Empresa

Totes La Paz empezó sus operaciones el 10 de abril de 1986 con la idea visionaria del señor Carlos España y su esposa Isabel Ayala de España. Empezaron las operaciones de Totes desde el pequeño departamento donde vivían junto a algunos amigos y familiares que los apoyaron desde el principio.

(TOTE's, 2011)

En esta época el Sr. España realizaba la limpieza, siguiendo el entrenamiento que recibió cuando vivía en Estados Unidos, mientras que la Sra. España atendía

llamadas y manejaba la primera combi Volkswagen que compraron de segunda mano. Poco a poco fueron creciendo y lograron alquilar la primera oficina de Totes en la zona de Sopocachi frente a la Plaza del Cóndor. Cuentan que durante estos años era muy común ver alfombras colgadas de las alas de la estatua del cóndor por la falta de espacio en la oficina. Mientras los trabajos iban aumentando también crecía la familia Totes.

En los próximos años, la empresa, que ya contaba con más de 30 empleados, se mudó a una pequeña casa por la calle 10 de Obrajes de la Zona Sur que contaba con tres pequeños cuartos y un amplio garaje donde se realizaba la limpieza de alfombras. Ya no se necesitarían estatuas para secar las alfombras. Los años que siguieron, mostraron el liderazgo de Totes en el mercado de limpieza, y fueron marcados por el crecimiento de la empresa. Se empezó a construir la oficina central en la ciudad de La Paz en 1995, estableciendo así el edificio actual que cuenta con un sótano y cuatro pisos con talleres, almacenes y oficinas en la zona de Obrajes. Por fin se completó la construcción dos años después en febrero de 1997 (Ver Figura 2.1). Entretanto, se logró cumplir uno de los sueños de los señores España: llegar a tener presencia nacional.

La primera sucursal de Totes se inauguró el 30 abril de 1993 en la ciudad de Cochabamba. Viendo el éxito de esta, el 12 abril de 1997 se abren las puertas de Totes Santa Cruz. Años después se abre Totes Sucre (1ro de Octubre, 2002) y Totes Tarija (8 de agosto, 2003) atendiendo también a Potosí y Oruro.

Misión

“Brindar limpieza 100% garantizada a todos sus clientes”. (TOTE's, 2011)

Visión

“Ser la empresa líder del mercado de limpieza nacional, contando con reconocimiento internacional, donde cada cliente encuentre las soluciones que necesita y nuestro personal encuentre un lugar donde crecer, todo esto en un marco de respeto al medio ambiente”. (TOTE's, 2011)

Servicios que presta

La Empresa de limpieza industrial TOTE's ofrece servicios de limpieza y mantenimiento general a domicilios y todo tipo de instituciones de manera eventual o fija. Ofrece todos sus servicios de manera integral a casas, oficinas, clínicas, hospitales, fábricas, petroleras, entidades financieras, embajadas, colegios, universidades, entre otras.



Figura 2.1: Oficina central TOTE's LTDA (TOTE's, 2011)

(TOTE's, 2011)

- **LAVADO DE ALFOMBRAS:** Limpieza a presión con agua y extracción inmediata que lava las fibras desde su base, recomendado por la mayoría de fabricantes de alfombras.
- **LAVADO Y PULIDO DE PISOS FRÍOS:** Encerado con lustrado de pisos de madera y lavado con pulido de otros pisos fríos (cerámica, azulejo, mármol, etc.) garantizando brillo duradero.
- **LIMPIEZA DE VIDRIOS:** Limpieza de todo tipo de vidrios con insumos y productos importados que no dejan rastros ni residuos.
- **DESENGRASADO DE COCINAS:** Limpieza profunda y desengrasado de cocinas dejando todo el ambiente desinfectado sin bacteria ni olores.
- **LAVADO DE TAPICES Y COLCHONES:** Dependiendo del tipo de tapiz o colchón, se realiza el lavado a presión con agua y extracción de manera más gentil y con productos especiales para diferentes telas, superficies y texturas.
- **LIMPIEZA EN SECO:** Máquinas importadas, especializadas en cepillado y aspirado, junto con micro-esponjas cargadas de químicos biodegradables, limpian las fibras de cada alfombra en un nuevo proceso patentado de limpieza de alfombras en seco.
- **DESINFECTADO Y SANITIZADO DE BAÑOS:** Limpieza profunda del inodoro, tina, lavamanos, espejo, paredes y piso, dejando el cuarto de baño desinfectado y sanitizado con productos especiales.
- **LAVADO PROFUNDO DE INTERIOR DE AUTOMÓVILES:** Utiliza el proceso de lavado a presión con agua y su extracción inmediata para lavar todo el interior del vehículo y silicona para proteger el tablero y otras áreas.
- **OZONIZADO DE AMBIENTES:** Ozonizado de ambientes para el eliminado permanente de todo tipo de olores y bacteria.
- **EXTRACCIÓN DE AGUA:** Extracción de agua en caso de emergencias por cualquier tipo de inundación.
- **MANTENIMIENTO GENERAL Y DESEMPOLVADO:** Personal capacitado y garantizado se encarga de remover toda tierra, polvo y otros residuos que se

acumulan a diario en todas las superficies del domicilio y oficina, dejando todo brillando de limpio.

- **FUMIGADO DE AMBIENTES:** Fumigado para eliminar pestes indeseadas en todo tipo de ambientes.
- **SERVICIO DE JARDINERIA, CAFETERÍA Y MENSAJERÍA:** Servicios complementarios principalmente para contratos.

Actualmente la Empresa de Limpieza Industrial TOTE's cuenta con un aproximado de 4000 funcionarios distribuidos en sus regionales de La Paz, Santa Cruz, Cochabamba y Tarija. Los cargos que cumplen los funcionarios son:

(Mendez, 2013)

- Gerente General
- Gerente Regional
- Jefe de Contabilidad
- Jefe de Almacenes
- Jefe de Recursos Humanos
- Secretarías y Administrativos
- Choferes
- Supervisores
- Encargados de Contrato
- Operarios de Limpieza
- Personal de Planta

2.3 Administración de Recursos Humanos

2.3.1 Administración de personal

El recurso humano o personal laboral representa la base fundamental de toda organización, a tal grado que una organización es el retrato de sus miembros; siendo estos últimos guiados por una persona profesional, una unidad o un departamento de Recursos Humanos (RRHH).

(Marconi, 2012) Afirma

A comienzos del siglo XX surge la unidad de **Relaciones industriales** dentro de las empresas, debido al crecimiento y a la complejidad de las tareas organizacionales en los albores de la revolución industrial. Aun así, el empleado se reducía a ser un elemento más dentro de la organización, como lo era la materia prima o los procesos de trabajo, sin tomar en consideración sus necesidades como individuo independiente y pensante.

En la década de 1950 se le llamo **Administración de personal**, pues ya no se trataba tan solo de mediar en las desavenencias y de aminorar los conflictos, sino, principalmente, de administrar el personal de acuerdo con la legislación laboral vigente y administrar los conflictos que surgían cotidianamente. Ya para 1970 surgió el concepto **Administración de Recursos Humanos**, aun cuando los empleados seguían considerándose como recursos productivos o meros agentes pasivos, cuyas actividades deben ser planificadas y controladas a partir de las necesidades de la organización. (p. 1)



Figura 2.2: Ramas de la Administración General (Marconi, 2012)

2.3.1.1 Definición de Administración de Recursos Humanos

(Marconi, 2012) Afirma

La ciencia de la Administración general cuenta con una rama denominada Administración de Recursos Humanos, que corresponde al recurso humano de una organización. Estos recursos (RRHH), se interpretan como el trabajo que aporta el conjunto de los empleados o colaboradores de esa organización, pero generalmente se entienden como la función o gestión que se ocupa de seleccionar, contratar, formar, emplear y retener a los participantes de la organización, las que pueden ser desempeñadas por una persona profesional, una unidad o un departamento de recursos humanos, conjuntamente con los directivos de la organización.

Esta disciplina es conceptualizada también como el proceso administrativo aplicado al acrecentamiento y conservación del esfuerzo, las experiencias, la salud, los conocimientos, las habilidades, etc., de los miembros de la organización, en beneficio del individuo, de la propia organización y del país en general. (p. 2)

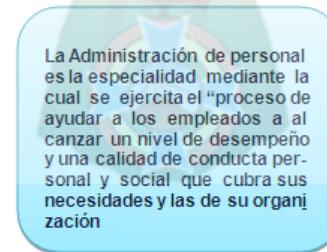


Figura 2.3: Definición de Administración de Personal (Marconi, 2012)

(Marconi, 2012) Afirma

Un concepto neto sostiene que la Administración de personal constituye la especialidad que tiene por objeto administrar la cultura laboral; existiendo otras definiciones al respecto, tales como:

- Es la utilización de los recursos humanos para alcanzar objetivos organizacionales.

- Es la disciplina que se encarga de organizar a los trabajadores y a las personas que laboran en una empresa para alcanzar los objetivos de ambas partes.
- Es la promoción y mantenimiento del más alto grado de bienestar físico, mental y social de los trabajadores en todas las ocupaciones.
- Es el proceso de ayudar a los empleados a alcanzar un nivel de desempeño y una calidad de conducta personal y social que cubra sus necesidades y las de su organización. (p. 3)

2.3.2 Objetivos y funciones de la Administración de personal

2.3.2.1 Objetivos

(Marconi, 2012) Afirma

La Administración de Personal se reconoce como un área interdisciplinaria, y tiene como objeto de estudio y de acción la administración de las personas vinculadas directa o indirectamente a una empresa o conjunto de empresas, a partir de la búsqueda de una coherencia cultural mínima en su dimensión laboral. El propósito fundamental de la Administración de recursos humanos es mejorar las contribuciones productivas del personal a la organización. (p. 3)

2.3.2.2 Importancia

Toda disciplina reviste importancia igual que la Administración de Personal, la que adquiere su importancia al evitar o solucionar situaciones como:

(Marconi, 2012) Afirma

- Empleados que se desempeñan por debajo de su capacidad.
- Contratar a la persona equivocada para el puesto.
- Alta rotación de personal.
- Demandas laborales.
- Sanciones por prácticas inseguras de acuerdo a las leyes (seguridad e higiene).
- Prácticas laborales injustas.
- Bajos niveles de eficacia debido a la falta de capacitación.

Por lo anterior, esta disciplina debe de llevar a cabo acciones para proporcionar y mantener una fuerza de trabajo adecuada, es decir, con las características y en la cantidad que la organización necesita para lograr su propósito. (p. 3)



Figura 2.4: Funciones de la Administración de Personal (Marconi, 2012)

2.3.2.3 Funciones

(Marconi, 2012) Afirma

- Análisis de puestos (determinar la naturaleza del trabajo de cada empleado)
- Planeación de las necesidades de mano de obra y el reclutamiento de los candidatos de los puestos.
- Selección de los candidatos a ocupar los puestos.
- Inducción y capacitación a los nuevos empleados.
- La Administración de sueldos y salarios (la forma de compensar a los empleados).
- Ofrecimiento de incentivos y beneficios.
- Evaluación del desempeño.
- Comunicación interpersonal (entrevistas, asesoría, disciplinar).
- Desarrollo de gerentes. (p. 4)

2.4 Ingeniería del Software

(Sommerville, 2005) Afirma

La ingeniería de software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después que se utiliza. En esta definición, existen dos frases clave:

1. **Disciplina de la ingeniería.** Los ingenieros hacen que las cosas funcionen. Aplican teorías, métodos y herramientas donde sean convenientes, pero las utilizan de forma selectiva y siempre tratando de descubrir soluciones a los problemas, aun cuando no existan teorías métodos aplicables para resolverlos. Los ingenieros también saben que deben trabajar con restricciones financieras y organizacionales, por lo que buscan soluciones tomando en cuenta estas restricciones.

2. **Todos los aspectos de producción de software.** La ingeniería del software no sólo comprende los procesos técnicos del desarrollo de software, sino también con actividades tales como la gestión de proyectos de software y el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software.

En general, los ingenieros de software adoptan un enfoque sistemático y organizado en su trabajo, ya que es la forma más efectiva de producir software de alta calidad. Sin embargo, aunque la ingeniería consiste en seleccionar el método más apropiado para un conjunto de circunstancias, un enfoque más informal y creativo de desarrollo podría ser efectivo en algunas circunstancias. El desarrollo informal es apropiado para el desarrollo de sistemas basados en Web, los cuales requieren una mezcla de técnicas de software y de diseño gráfico. (p. 23-24)

2.5 Modelos y Metodologías de Desarrollo

2.5.1 Metodología XP

La Programación Extrema es una metodología ágil de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado.

2.5.1.1 Origen de la metodología XP

Desarrollada por Kent Beck.

«*Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.*» Kent Beck



Figura 2.5: Kent Beck (Fernández, 2002)

XP surgió como respuesta y posible solución a los problemas derivados del cambio en los requerimientos se plantea como una metodología a emplear en proyectos de riesgo aumenta la productividad.

(Fernández, 2002) Afirma

Las cuatro variables:

- **Coste:** Máquinas, especialistas y oficinas.
- **Tiempo:** Total y de Entregas.
- **Calidad:** Externa e Interna.
- **Alcance:** Intervención del cliente. (p. 3)

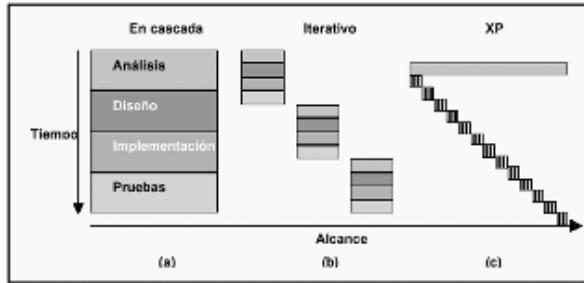


Figura 2.6: El coste del cambio (Fernández, 2002)

2.5.1.2 Etapas la metodología XP

La XP utiliza un enfoque OO, como su paradigma de desarrollo preferido. La XP abarca un conjunto de reglas y prácticas que ocurren en el contexto de 4 actividades del marco de trabajo:

1º fase: Planificación del proyecto.

Historias de usuario

(Fernández, 2002) Afirma

Las historias de usuario tienen el mismo propósito que los casos de uso. Las escriben los propios clientes, tal y como ven ellos las necesidades del sistema.

Las historias de usuario son similares al empleo de escenarios, con la excepción de que no se limitan a la descripción de la interfaz de usuario. También conducirán el proceso de creación de los test de aceptación (empleados para verificar que las historias de usuario han sido implementadas correctamente). Existen diferencias entre estas y la tradicional especificación de requisitos. La principal diferencia es el nivel de detalle. Las historias de usuario solamente proporcionaran los detalles sobre la estimación del riesgo y cuánto tiempo conllevará la implementación de dicha historia de usuario. (p. 6)

Tabla 2.1: Modelo propuesto para una historia de usuario (Anaya)

Historia de usuario		
Numero:	Nombre Historia de Usuario:	
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):		
Usuario:	Iteración asignada:	
Prioridad en Negocio: (Alta / Media / Baja)	Puntos estimados:	
Riesgo en desarrollo: (Alto / Medio / Bajo)	Puntos Reales:	
Descripción:		
Observaciones:		

Iteraciones y planes de iteración

Velocidad del proyecto

La velocidad del proyecto es una medida que representa la rapidez con la que se desarrolla el proyecto. Usando la velocidad del proyecto se controla todas las tareas que se puedan desarrollar en el tiempo del que dispone la iteración. Es conveniente reevaluar esta medida cada 3 ó 4 iteraciones y si se nota que no es adecuada hay que negociar con el cliente un nuevo (Plan de Entregas).

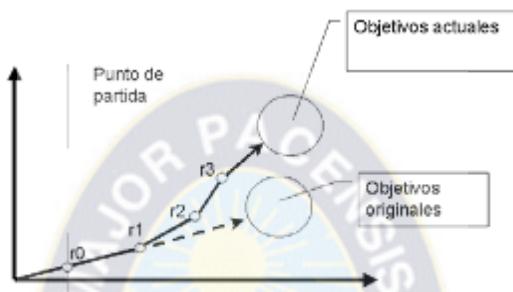


Figura 2.7: Iteraciones y planes de iteración (Fernández, 2002)

Programación en pareja

Se aconseja la programación en parejas pues incrementa la productividad y la calidad del software desarrollado. El trabajo en pareja involucra a dos programadores trabajando en el mismo equipo; mientras uno codifica, el otro analiza si ese método o función es adecuado y está bien diseñado.

2^afase: Diseño

(Anaya)

- **Diseños simples.** La metodología X.P sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo desarrollar.
- **Glosarios de términos.** Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código.
- **Riesgos.** Si surgen problemas potenciales durante el diseño, X.P sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema.
- **Funcionalidad extra.** Nunca se debe añadir funcionalidad extra al programa aunque se piense que en un futuro será utilizada. Sólo el 10% de la misma es utilizada, lo que implica que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos.
- **Refactorizar.** Refactorizar es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Refactorizar supone revisar de

nuevo estos códigos para procurar optimizar su funcionamiento. Es muy común rehusar códigos ya creados que contienen funcionalidades que no serán usadas y diseños obsoletos. Esto es un error porque puede generar código completamente inestable y muy mal diseñado; por este motivo, es necesario refactorizar cuando se va a utilizar código ya creado.

- **Tarjetas C.R.C.** El uso de las tarjetas C.R.C (Class, Responsibilities and Collaboration) permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica. Las tarjetas C.R.C representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad.

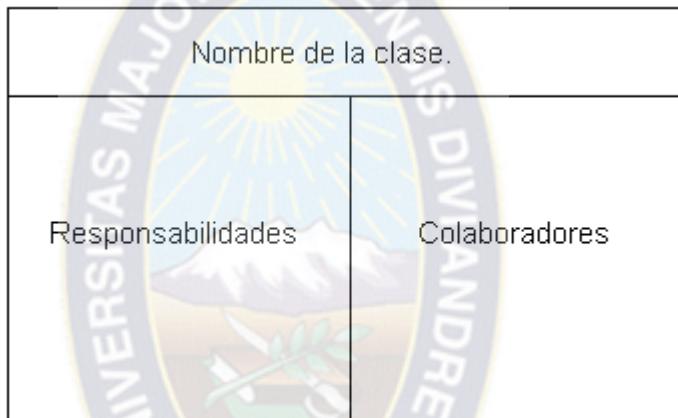


Figura 2.8: Modelo de tarjeta CRC (Anaya)

3^a Fase: Codificación.

Como ya se dijo, el cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de X.P. A la hora de codificar una historia de usuario su presencia es aún más necesaria. Los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada.

La codificación debe hacerse ateniendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.

Crear test que prueben el funcionamiento de los distintos códigos implementados permite ayudar a desarrollar dicho código. Crear estos test antes ayuda a saber qué es exactamente lo que tiene que hacer el código a implementar y saber que una vez implementado pasará dichos test sin problemas ya que dicho código ha sido diseñado para ese fin. Se puede dividir la funcionalidad que debe cumplir una tarea a programar en pequeñas unidades, de

esta forma se crearán primero los test para cada unidad y a continuación se desarrollará dicha unidad, así poco a poco se consigue un desarrollo que cumpla todos los requisitos especificados. X.P opta por la programación en pareja ya que permite un código más eficiente y con una gran calidad.

(Anaya)

X.P sugiere un modelo de trabajo usando repositorios de código dónde las parejas de programadores publican cada pocas horas sus códigos implementados y corregidos junto a los test que deben pasar. De esta forma el resto de programadores que necesiten códigos ajenos trabajarán siempre con las últimas versiones. Para mantener un código consistente, publicar un código en un repositorio es una acción exclusiva para cada pareja de programadores.

X.P también propone un modelo de desarrollo colectivo en el que todos los programadores están implicados en todas las tareas; cualquiera puede modificar o ampliar una clase o método de otro programador si es necesario y subirla al repositorio de código. El permitir al resto de los programadores modificar códigos que no son suyos no supone ningún riesgo ya que para que un código pueda ser publicado en el repositorio tiene que pasar los test de funcionamiento definidos para el mismo.

La optimización del código siempre se debe dejar para el final. Hay que hacer que funcione y que sea correcto, más tarde se puede optimizar.

X.P afirma que la mayoría de los proyectos que necesiten más tiempo extra que el planificado para ser finalizados no podrán ser terminados a tiempo se haga lo que se haga, aunque se añadan más desarrolladores y se incrementen los recursos. La solución que plantea X.P es realizar un nuevo "Release plan" para concretar los nuevos tiempos de publicación y de velocidad del proyecto.

.A la hora de codificar no seguimos la regla de X.P que aconseja crear test de funcionamiento con entornos de desarrollo antes de programar. Nuestros test los obtendremos de la especificación de requisitos ya que en ella se especifican las pruebas que deben pasar las distintas funcionalidades del programa, procurando codificar pensando en las pruebas que debe pasar cada funcionalidad.

4ª Fase: Pruebas.

Uno de los pilares de la metodología X.P es el uso de pruebas para comprobar el funcionamiento de los códigos que se vayan a implementar.

(Anaya)

El uso de los test en X.P es el siguiente:

- Se deben crear las aplicaciones que realizarán los test con un entorno de desarrollo específico para test.
- Hay que someter a tests las distintas clases del sistema omitiendo los métodos más triviales.
- Se deben crear los test que pasaran los códigos antes de implementarlos; en el apartado anterior se explicó la importancia de crear antes los test que el código.

Un punto importante es crear pruebas que no tengan ninguna dependencia del código que en un futuro probará. Hay que crear las pruebas absteniéndose de saber cómo será el futuro código, de esta forma se asegura la independencia de la prueba respecto al código que evalúa.

(Anaya)

"El uso de los test es adecuado para observar la refactorización. Los test permiten verificar que un cambio en la estructura de un código no tiene por qué cambiar su funcionamiento. Test de aceptación. Los test mencionados anteriormente sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario. Para asegurar el funcionamiento final de una determinada historia de usuario se deben crear (Test de aceptación); estos test son creados y usados por los clientes para comprobar que las distintas historias de usuario cumplen su cometido. Al ser las distintas funcionalidades de nuestra aplicación no demasiado extensas, no se harán test que analicen partes de las mismas, sino que las pruebas se realizarán para las funcionalidades generales que debe cumplir el programa especificado en la descripción de requisitos."

2.6 Ingeniería Web

La **ingeniería web** es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la Red de Internet.

La ingeniería web se debe al crecimiento desenfrenado que está teniendo la Web y como está ocasionando un impacto en la sociedad y el nuevo manejo que se le está dando a la información en las diferentes áreas en que se presenta, ha hecho que las personas tiendan a realizar todas sus actividades por esta vía.

(Wikipedia-IngenieríaWeb, 2014) Afirma

La ingeniería Web es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la Web. En este sentido, la ingeniería Web hace referencia a las metodologías, técnicas y herramientas que se utilizan en el desarrollo de aplicaciones Web complejas y de gran dimensión en las que se apoya la evaluación, diseño, desarrollo, implementación y evolución de dichas aplicaciones.

2.6.1 Áreas

El desarrollo de aplicaciones Web posee determinadas características que lo hacen diferente del desarrollo de aplicaciones o software tradicional y sistemas de información. La ingeniería Web es multidisciplinaria y aglutina contribuciones de diferentes áreas como ser: la arquitectura de la información, ingeniería de hipermedia/hipertexto, ingeniería de requisitos, diseño de interfaz del usuario, usabilidad, diseño gráfico y de presentación, diseño y análisis de sistemas, ingeniería de software, ingeniería de datos, indexado y recuperación de información, testeо, modelado y simulación, despliegue de aplicaciones, operación de sistemas y gestión de proyectos.

La ingeniería Web no es un clon o subconjunto de la ingeniería de software aunque ambas incluyen desarrollo de software y programación, pues a pesar de que la ingeniería Web

utiliza principios de ingeniería de software, incluye nuevos enfoques, metodologías, herramientas, técnicas, guías y patrones para cubrir los requisitos únicos de las aplicaciones web. Sin embargo el término de ingeniería web ha sido un término muy controvertido especialmente para profesionales en disciplinas tales como la ingeniería del software ya que no la consideran como un campo dentro de la ingeniería.

(Wikipedia-IngenieríaWeb, 2014) Afirma

Los principales aspectos de la ingeniería de la Web incluyen, entre otros, los siguientes temas:

- Diseño de procesos de negocio para aplicaciones web.
- Herramientas CASE para aplicaciones web.
- Generación de código para aplicaciones web.
- Desarrollo web colaborativo.
- Modelado conceptual de aplicaciones web.
- Diseño de Modelo de datos para sistemas de información web.
- Ingeniería web empírica.
- Entornos de desarrollo de aplicaciones web integrados.
- Herramientas de autor para contenido multimedia.
- Pruebas de rendimiento de aplicaciones basadas en la web.
- Personalización y adaptación de aplicaciones web.
- Herramientas y métodos de prototipado.
- Control de calidad y pruebas de sistemas.
- Ingeniería de requisitos para aplicaciones web.
- Aplicaciones para la Web Semántica.
- Factorías de software para la web.
- Métodos, herramientas y automatización de pruebas para aplicaciones web.
- Aplicaciones web móviles y ubicuas.
- Usabilidad de aplicaciones web.
- Accesibilidad para la web.
- Metodologías de diseño web.
- Formación en ingeniería web.
- Diseño de interfaz de usuario.
- Métricas para la web, estimación de costes y medición.
- Gestión de proyectos web y gestión de riesgos.
- Desarrollo y despliegue de servicios web.

2.6.2 Categorías

(Wikipedia-IngenieríaWeb, 2014) Afirma

Los sitios web pueden ser categorizados de la siguiente forma:

- Sólo estático que se enfoca en la organización de la estructura y el contenido, en la forma como se va a presentar la información y que sea fácil de manejar para cualquier usuario, pero debe tener en cuenta la eficiencia y la confiabilidad.
- Sitio estático con formularios, de entrada este sitio tiene las mismas características que el anterior, adicionándole que él le permite a los

usuarios la interacción por medio de cuestionarios, comentario y sugerencias.

- Sitio con acceso de datos dinámicos aquí, además de las características antes mencionadas, cuenta con bases de datos en las cuales el usuario puede realizar consultas y búsquedas.
- Sitio creado dinámicamente en este sitio los requerimientos son parecidos pero deben suprir con las necesidades de cada usuario; creando sitios dinámicos que sean compatibles con el entorno de navegación de cada usuario.
- Aplicación de software basada en la Web este sitio puede tener todas las características antes mencionadas, pero logrando un parecido con una implementación cliente/servidor comúnmente conocido que a un sitio web estático.

Con el pasar del tiempo y la constante evolución de la tecnología se ha podido observar la necesidad y la utilidad de la Internet para mejorar de cierta manera nuestras condiciones de vida y así fortalecer más nuestro proceso de formación educativa y contribuir con un mejoramiento del global de las necesidades de cada individuo.

2.6.3 Naturaleza multidisciplinaria

(Wikipedia-IngenieríaWeb, 2014)

La ingeniería del software, incluye nuevas metodologías de desarrollo esenciales para la administración de proyectos. Actualmente la ingeniería web ha adoptado también metodologías de la ingeniería del software y ha creado muchas nuevas. Debido a que la información es publicada para conocimiento de todo el mundo, hay que tener muy en cuenta aspectos sociales, jurídicos y éticos que pueden influir a la hora de la publicación. De acuerdo con esto, la ingeniería Web puede utilizar una parte de cada una de estas disciplinas y no ser dominada por puntos de vista muy particulares, es una respuesta de carácter multidisciplinario para las aplicaciones Web.

2.7 Metodología de modelado Web – WEBML

WebML (Web Modeling Language) es una metodología visual para diseñar aplicaciones basadas en la Web. Proporciona unos gráficos, todavía formales, especificaciones, incluidas en un proceso del plan completo que puede ayudarse por las herramientas del diseño visual.

(webml.org) Afirma

WebML proporciona un entorno gráfico formal con especificaciones incluidas en un proceso de planificación completo que se ayuda con elementos visuales. Los objetivos principales del proceso del diseño WebML son:

- a) Expresar la estructura de una aplicación de Web con una descripción de alto nivel orientado a la evolución, y mantenimiento;
- b) Proporcionar múltiples vistas de un mismo modulo;
- c) Separar la información de la que está en páginas de composición, navegación y presentación que pueden definirse y pueden reestructurarse independientemente;
- d) Guardar la meta-information durante el proceso de planificación dentro de una base de datos para que esta pueda usarse durante la vida de la aplicación de páginas Web para que estas generen dinámicamente;
- e) Permitir a los usuarios la especificación de políticas de personalización de una a una de las aplicaciones;

- f) Habilitar la especificación de la funcionalidad del manejo de los datos para que este actualizada y actuar recíprocamente con los servicios externos.

Traducción mía

WebML provides graphical, yet formal, specifications, embodied in a complete design process, which can be assisted by visual design tools. The main objectives of the WebML design process are:

- a) expressing the structure of a Web application with a high-level description, which can be used for querying, evolution, and maintenance;
- b) providing multiple views of the same content;
- c) separating the information content from its composition into pages, navigation, and presentation, which can be defined and evolved independently;
- d) storing the meta-information collected during the design process within a repository, which can be used during the lifetime of the application for dynamically generating Web pages;
- e) modelling users and communities explicitly in the repository, to permit the specification of personalization policies and one-to-one applications;
- f) Enabling the specification of data manipulation operations for updating the site content or interacting with arbitrary external services.

Este método tiene cinco modelos: la estructura, derivación, composición, navegación y presentación.

2.7.1 El Modelo estructural

(webml.org) Afirma

El modelo de estructural (modelo de datos) es una adaptación conveniente de modelos conceptuales para el diseño de datos, como ya se utiliza en otras disciplinas, como el modelo de la base de datos, diseño de software, y representación del conocimiento. Es compatible con los modelos de datos del tipo Entidad-relación, modelo de la base de datos conceptual, y con los diagramas UML para la programación orientada a objetos, usados el planeando objeto-orientado, clasifican. Los elementos fundamentales de modelos de los datos son entidades, definidas como los recipientes de elementos de los datos, y relaciones, definidas como las conexiones semánticas entre las entidades. Las entidades tienen atributos llamados propiedades con un tipo de datos asociado. Pueden organizarse las entidades en las jerarquías de la generalización y relaciones puede restringirse por medio de los constreñimientos del cardinalidad. En el caso de las entidades son direccionados individualmente por medio de un único identificador (OID). WebML OIDs son conceptos abstractos que pueden llevarse a cabo de las maneras alternativas en manejador de la base de datos por ejemplo, las llaves primarias en un datos correlativo se guardan en un archivo XML ID.

Traducción mía

The WebML data model is a suitable adaptation of conceptual models for data design, as already in use in other disciplines, such as database design, software engineering, and knowledge representation. It is compatible with the Entity-Relationship data model, used in conceptual database design, and with UML class diagrams, used in object-oriented modeling. The fundamental elements of data models are **entities**, defined as containers of data elements, and **relationships**, defined as semantic connections between entities. Entities have named properties, called attributes, with

an associated type. Entities can be organized in **generalization hierarchies** and relationships can be restricted by means of cardinality constraints. Instances of entities are considered individually addressable by means of a **unique identifier (OID)**. WebML OIDs are abstract concepts, which can be implemented in alternative ways in the underlying storage manager, e.g., primary keys in a relational data store or XML ID attributes in a XML data source.

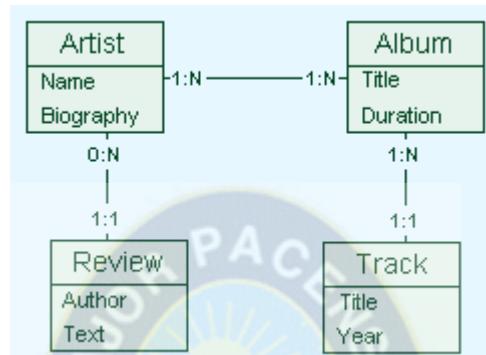


Figura 2.9 Ejemplo de modelo estructural (webml.org)

2.7.2 Modelo de derivación

“En otros términos es similar a las VISTAS en el modelado de las bases de datos. Como la VISTA en Oracle o MySQL.”

Para cada página hay Una tabla abstracta de datos. Pero se une con otras tablas.”
(Wikipedia-WebML, 2013)

In other words it is similar to **VIEWS** in databases modelling. Like VIEW in Oracle or MySQL.
For each page there is **One abstract Table** of datas. But it is merged from other tables.
Traducción mía.

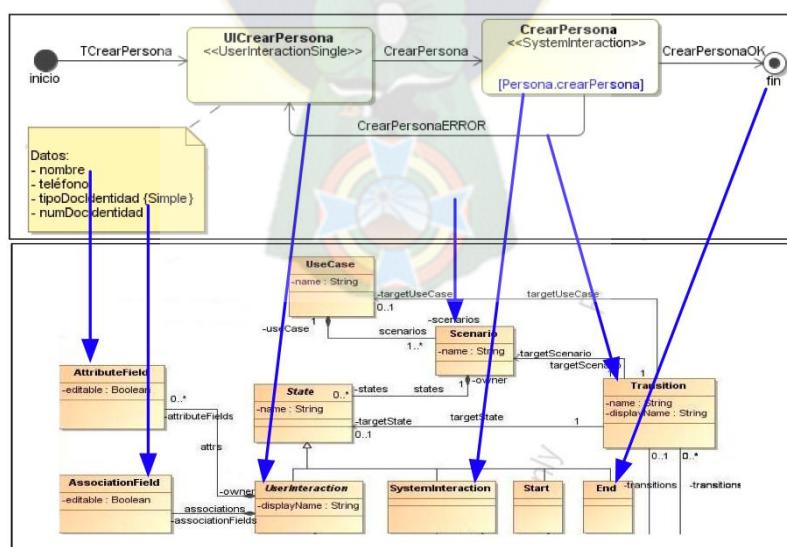


Figura 2.10 Ejemplo de modelo de derivación (Ceballos, Arboleda, & Casallas, 2008)

2.7.3 Modelo del hipertexto

El modelo más importante de la metodología de WebML, modela la navegación de usuario en la red.

El Modelo del hipertexto se compone de 2 modelos: La composición y el modelo De navegación.

2.7.3.1 Modelo de composición

El propósito de composición es planear y definir qué todos los nodos sea parte del conjunto de hipertextos contenido en el sitio Web. Más precisamente, la composición diseña específicamente las unidades satisfechas (las unidades para el usuario), es decir, los elementos de información individuales que pueden aparecer en el sitio de Web, y páginas, es decir, recipientes por medio de que la información sea realmente útil para el usuario. En una escena concreta, por ejemplo, un HTML o aplicación de WML de un sitio de WebML, se trazan páginas y unidades a las estructuras convenientes en el idioma de la entrega, por ejemplo, las unidades pueden trazar a archivos HTML y páginas organizados por marcos HTML.

(webml.org) Afirma

La composición describe qué páginas componen el hipertexto, y qué unidades satisfechas constituyen una página. Las páginas del sitio de Web realmente son los recipientes de información entregados al lector.

Las unidades son los elementos en la información descrita en el modelo de datos. Siete tipos de unidades son los componentes de páginas predefinidos en WebML: los datos, los multi-datos, el índice (y su multichoice de las variantes y jerárquico), entrada, el scroller. Cada unidad se asocia a una entidad subyacente de que el volumen de la unidad se computa. La especificación de la entidad subyacente dicta el tipo del objeto de que el volumen de una unidad se deriva (el ej., álbumes, artistas,...). Cuando apropiado, pueden asociarse las unidades opcionalmente a un seleccionador, es decir, la especificación de un juego de restricciones que determina los casos reales de la entidad subyacente ser usado como el volumen de la unidad al ejecutarse.

Traducción mía

Composition describes which pages compose the hypertext, and which content units make up a page. The **pages** of the Web site are the containers of information actually delivered to the reader. **Units** are atomic content elements used to publish the information described in the data model. Seven types of units are predefined in WebML to compose pages: **data**, **multi-data**, **index** (and its variants multichoice and hierarchical), **entry**, **scroller**. Each unit is associated to one underlying entity, from which the content of the unit is computed. The specification of the underlying entity dictates the object type from which the content of a unit is derived (e.g., albums, artists,...). When appropriate, units may be optionally associated to a selector, i.e., the specification of a set of restrictions that determine the actual instances of the underlying entity to be used as the content of the unit at runtime.

2.7.3.2 El Modelo de navegación

(webml.org) Afirma

La navegación del sitio se especifica por los eslabones. Pueden definirse los eslabones entre las unidades dentro de una sola página, entre unidades puestas en las páginas diferentes, y entre las páginas. Se llama contexto de navegación a la información llevada a lo largo de un eslabón. Se llaman eslabones contextuales a aquellos que llevan la información, se llaman eslabones no-contextuales a los que no tienen la información del contexto asociada. La información del contexto es típicamente necesaria para asegurar la compatibilidad de unidades.

Traducción mía

Navigation of the site is specified thru links. Links can be defined between the units inside a single page, between units placed in different pages, and between pages. The information carried along a link is called navigation context, or simply context. Links that carry context information are called contextual links, whereas links that have no associated context information are called non-contextual links. Context information is typically necessary to ensure the computability of units.

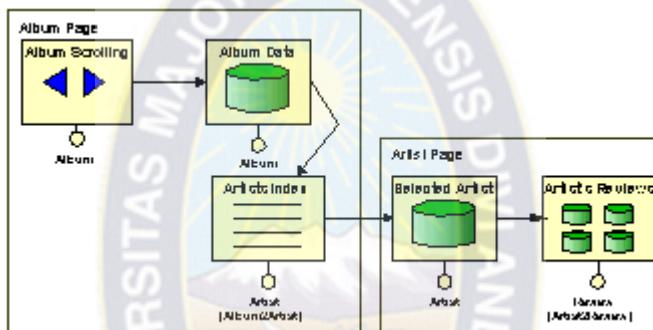


Figura 2.11 Ejemplo de un modelo de hipertexto (webml.org)

2.7.3.3 Modelo de personalización

La personalización es la definición del estilo de la presentación basada en el usuario. En WebML, las unidades, las páginas, y la presentación que los llama, pueden definirse las vistas del sitio para presentar al usuario o a un grupo específico de ellos. Esto puede hacerse de dos maneras complementarias:

(Wikipedia-WebML, 2013)

- **La personalización declaratoria:** el diseñador define los conceptos derivados (ej., entidades, los atributos, los componentes multi-estimados) de quien la definición depende de los datos específicos para el usuario. De esta manera, la personalización se especifica declaratoriamente; el sistema rellena al modelo de información para cada usuario al computar el volumen de unidades.
- **La personalización procesal:** WebML incluye una sintaxis de XML para escribir reglas comerciales que computan y guardan la información específica para el usuario. Una regla comercial es un evento-condición-acción triple que especifica el evento a ser supervisado la condición previa a ser verificada cuando el evento ocurre, y la acción a ser tomada cuando la condición se encuentra verdadera. Tareas típicas realizadas por las reglas comerciales son la asignación de usuarios a grupos del usuario basados en información dinámicamente reunida, la notificación de mensajes a los usuarios en la actualización de la base de información (la tecnología del empujón), el registro de acciones del usuario en las estructuras de los datos usuario-específicas, y así sucesivamente.

Traducción mía

- **Declarative personalization:** the designer defines derived concepts (e.g., entities, attributes, multi-valued components) whose definition depends on user-specific data. In this way, customization is specified declaratively; the system fills in the information relative to each user when computing the content of units.
- **Procedural personalization:** WebML includes an XML syntax for writing business rules that compute and store user-specific information. A business rule is a triple event-condition-action, which specifies the event to be monitored, the precondition to be checked when the event occurs, and the action to be taken when the condition is found true. Typical tasks performed by business rules are the assignment of users to user groups based on dynamically collected information, the notification of messages to users upon the update of the information base (push technology), the logging of user actions into user-specific data structures, and so on.

2.7.3.4 Modelo de Presentación

(webml.org) Afirma

La presentación consiste en definir la mirada y cómo se comportan las páginas en una vista del sitio. WebML no incluye a un modelo específico para expresar la presentación al nivel conceptual, pero utiliza gráficos que se acerquen un poco al concepto que se quiere expresar.

Desde que pueden representar esta característica técnica de WebML usando XML, la presentación es considerada como una transformación del documento que traza la especificación de WebML de una página en una página escrita en un idioma de aplicación concreto como JSP o ASP.NET. Por consiguiente, la presentación se dirige en WebML usando XSL que llame a las hojas vistas del sitio, páginas, unidades y subelementos de la unidad. XSL llaman las hojas alojan la entrada las especificaciones de WebML, codificado como documentos XML que conforman al Documento Tipo de Definición de WebML, y plantillas de páginas de rendimiento que incluyen el código de enlace requerido y los datos solicitados por las consultas. Una aplicación WebML puede incluir algunos hojas de estilo pre-definidas de presentación y los componentes del lado del servidor que apoyan a los datos que acceden a las consultas necesarias para mostrar en las plantillas de las páginas y hojas de estilo producidos por el XSL.

Traducción mía

Presentation is the orthogonal task of defining the look and feel of pages in a site view. WebML does not include a specific model for expressing presentation at the conceptual level, but leverages standard approaches, more familiar to graphic and communication experts.

Since WebML specifications can be represented using XML, presentation is considered like a document transformation mapping the WebML specification of a page into a page written in a concrete implementation language like JSP or ASP.NET. Consequently, presentation is addressed in WebML by attaching XSL style sheets to site views, pages, units and unit subelements. XSL style sheets take in input WebML specifications, coded as XML documents conforming to the WebML Document Type Definition, and output page templates embodying the required mark-up code and data access queries. An implementation of WebML may include several pre-defined presentation style sheets and the server-side components supporting the data access queries needed to populate the content of the page templates produced by the XSL style sheets.

. Sitio = Estructura + Composición + Navegación + Presentación

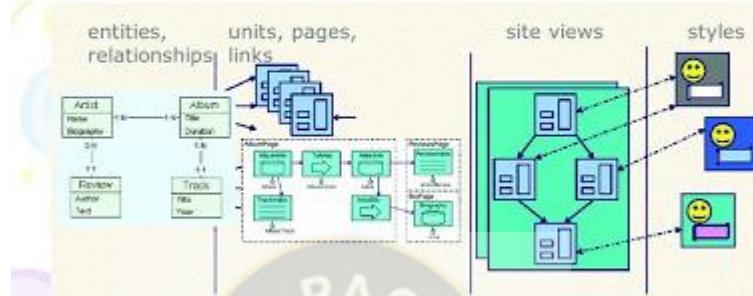


Figura 2.12 Ejemplo de Modelo de Presentación (tecnologimerlin, 2007)

2.7.4 El Proceso de Diseño

(Wikipedia-WebML, 2013) Afirma

Un proceso de diseño típico que usa los beneficios de WebML iterativamente. Los pasos durante cada ciclo de diseño son:

- **La Captura de requisitos.** Se recogen los requisitos de la aplicación, qué incluye los objetivos principales del sitio, su público designado, los ejemplos de volumen, pautas de estilo, personalización requerida y constreñimientos debido a los datos del legado.
- **El diseño de datos.** Los datos los planes especiales, el modelo estructural, los esquemas lógicos existentes de fuentes de datos antiguos.
- **El diseño del hipertexto "en grande".** El desarrollador Web define la estructura "en grande" del modelo de hipertexto, identificando páginas y unidades, uniéndolas, y trazando las unidades a las entidades principales y relaciones del esquema de la estructura. De esta manera, él desarrolla una vista del sitio "esquema", y entonces el iterativamente lo mejora.
- **El diseño del hipertexto "en pequeño".** El desarrollador Web luego en el diseño "en pequeño" del hipertexto, considerando cada página y unidad individualmente. En esta fase, él puede agregar los enlaces no-contextuales entre páginas, puede consolidar los atributos que deben ser incluidos dentro de una unidad, y puede introducir nuevas páginas o unidades para los requerimientos especiales (por ejemplo, páginas del índice alternativas para localizar los objetos, filtros para investigar la información deseada, y así sucesivamente).
- **El diseño de la presentación.** Una vez que todas las páginas son suficientemente estables, el diseñador gráfico agrega a cada página Web un estilo para su presentación.
- **El Diseño de y del Grupo.** El administrador Web define los rasgos de perfiles del usuario, basado en los requisitos de personalización. Se trazan usuarios potenciales y grupos del usuario a los usuarios de WebML y grupos, y posiblemente una vista del sitio diferente se crea para cada grupo. El ciclo del diseño en la próxima iteración tendrá que tener todas sus vistas identificadas.

Traducción mía

A typical design process using WebML proceeds by iterating the following steps for each design cycle:

- **Requirements Collection.** Application requirements are gathered, which include the main objectives of the site, its target audience, examples of content, style guidelines, required personalization and constraints due to legacy data.
- **Data Design.** The data expert designs the structural model, possibly by reverse-engineering the existing logical schemas of legacy data sources.
- **Hypertext Design "in the large".** The Web application architect defines the structure "in the large" of the hypertext, by identifying pages and units, linking them, and mapping units to the main entities and relationships of the structure schema. In this way, he develops a "skeleton" site view, and then iteratively improves it.
- **Hypertext Design "in the small".** The Web application architect concentrates next in the design "in the small" of the hypertext, by considering each page and unit individually. At this stage, he may add non-contextual links between pages, consolidate the attributes that should be included within a unit, and introduce novel pages or units for special requirements (e.g., alternative index pages to locate objects, filters to search the desired information, and so on).
- **Presentation Design.** Once all pages are sufficiently stable, the Web style architect adds to each page a presentation style.
- **User and Group Design.** The Web administrator defines the features of user profiles, based on personalization requirements. Potential users and user groups are mapped to WebML users and groups, and possibly a different site view is created for each group. The design cycle is next iterated for each of the identified site views.

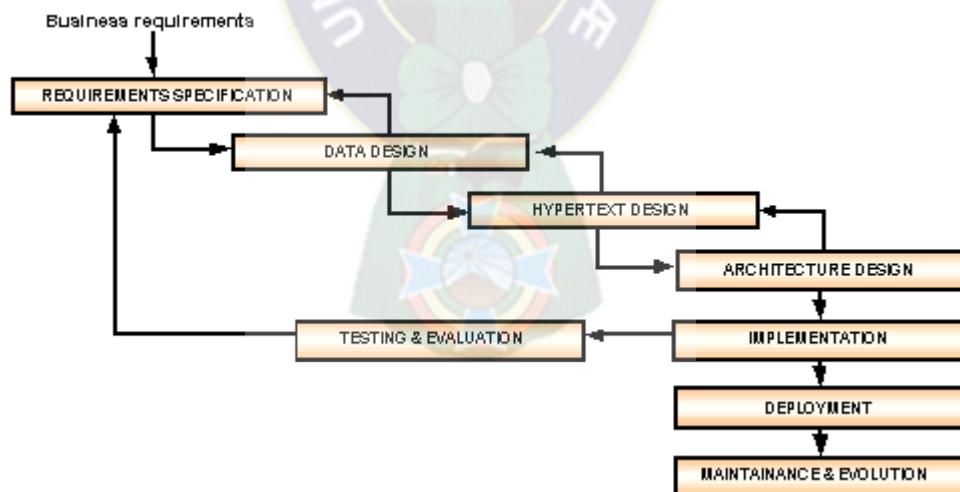


Figura 2.13 Proceso de diseño (webml.org)

2.8 Herramientas para la Implementación

Aquí se citan las herramientas que son usadas para el desarrollo de aplicaciones web.

2.8.1 Java

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores distintos. Pero no se queda ahí, Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria.

“Java es la base de casi todos los tipos de aplicaciones en red y el estándar global para el desarrollo y suministro de aplicaciones móviles, juegos, contenido basado en web y software de empresa. Con más de 9 millones de desarrolladores en todo el mundo, Java permite desarrollar y desplegar de un modo eficiente interesantes aplicaciones y servicios. Con un conjunto integral de herramientas, un ecosistema maduro y un sólido rendimiento, Java ofrece portabilidad de aplicaciones incluso entre los entornos informáticos más dispares.”
(Oracle y Java _ Tecnologías _ Oracle ES)

2.8.2 PostgreSQL

(Martinez, 2010) Afirma

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

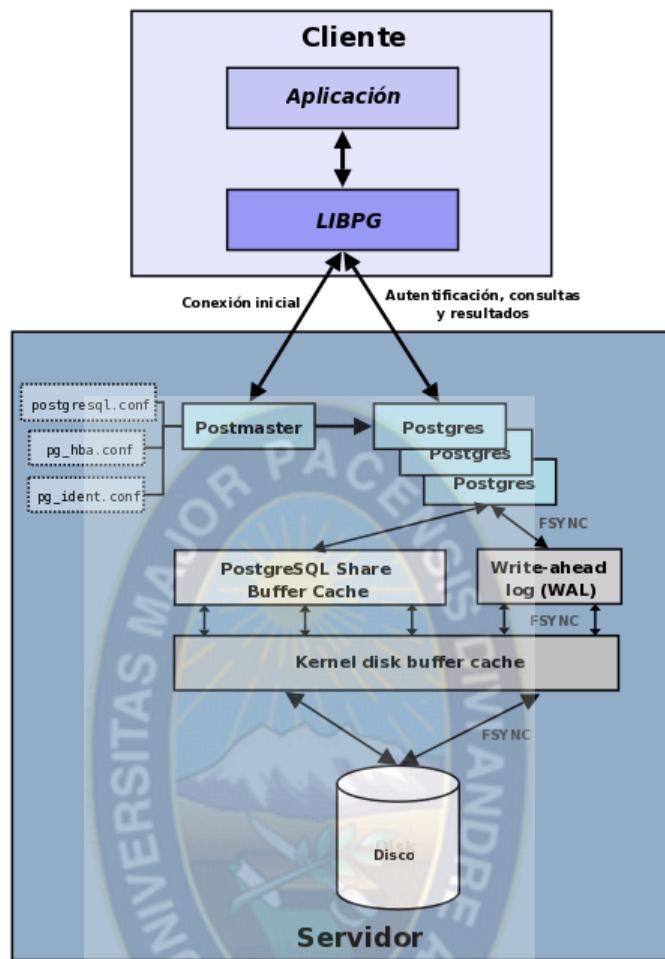


Figura 2.14 Componentes de un sistema PostgreSQL (Martinez, 2010)

2.8.3 Framework Spring

(Sevilla Sánchez)

Spring es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java. La primera versión fue escrita por Rod Jonhson. Es una plataforma Java que otorga una infraestructura de apoyo global al desarrollo de aplicaciones Java. De este modo, Spring se encarga de la infraestructura para que nosotros nos centremos en la aplicación.

Unos ejemplos pueden ser:

- Inyección de dependencia e inversión de control.
- Integración del acceso a datos.
- Facilitar el desarrollo de aplicaciones web separando claramente las partes del modelo, vista y controlador.
- Poder ejecutar métodos transaccionales en una base de datos sin necesidad de lidiar con API de transacción, métodos remotos sin tener que lidiar con API de procedimientos remotos, métodos de gestión sin JMX, control de mensajes sin JMS. (p. 8)

Java proporciona una gran cantidad de herramientas para desarrollar aplicaciones, pero carece de medios para organizar los elementos. Normalmente es el arquitecto Java el que se encarga de esta tarea pudiendo utilizar patrones.

(Sevilla Sánchez) Afirma

La inversión de control de Spring lo que hace es preocuparse de proporcionar una manera formal de creación de componentes dispares de una manera homogénea y de una manera funcional. Codifica componentes que se integran en las aplicaciones. Diversas instituciones y empresas eligen Spring como una manera de diseñar aplicaciones robustas fáciles de mantener. (p. 8)

2.8.3.1 Módulos del Spring

Spring consiste en elementos organizados en veinte módulos. Estos módulos se agrupan en un Contenedor (core container), Acceso a datos e integración, modelo vista controlador (módulo web MVC), aspectos (AOP), instrumentación y test.



Figura 2.15 Spring Framework Runtime (SpringFramework)

(Sevilla Sánchez) Afirma

El contenedor

El contenedor consiste en un núcleo, objetos bean, un contexto y un lenguaje de expresiones.

El núcleo y los beans son la parte fundamental de Spring, incluyendo la inversión de control y la inyección de dependencia. Este contenedor es una versión más compleja del patrón Factory.

Elimina la necesidad de programar singletons y permite desacoplar la configuración y especificación de dependencias de la lógica de programación.

El contexto se construye sobre la sólida base del núcleo. Así permite determinadas configuraciones. Así la internacionalización, propagación de eventos, lectura de recursos o la creación de contextos (como el web) formarán parte de este módulo.

Los Lenguajes de expresión permiten una potente herramienta de consulta y manipulación de un objeto en tiempo de ejecución. Es una extensión del “Unified EL”, especificado en la especificación JSP 2.1. El lenguaje permite asignar y obtener valores de las propiedades, asignar propiedades, invocar métodos, acceder al contexto de matrices, listas, índices, operadores aritméticos y lógicos, variables, y obtención de objetos por nombre del contendor de Spring.

Integración y acceso de Datos

La capa de Integración y acceso a datos consiste en la integración de los módulos JDBC, ORM, OXM, JMS y de transacción.

El módulo JDBC otorga una capa de abstracción que elimina la necesidad de crear código tedioso y trasforma las excepciones generadas por el proveedor de base de datos.

El módulo ORM otorga una integración con los APIS más populares de ORM como puedan ser JPA, JDO, Hibernate o iBatis.

El módulo OXM otorga una capa de abstracción para el mapeo Objeto/XML en distintas implementaciones como JAXB, Castor, XMLBeans, JiBX o XStream.

El módulo JMS contiene características para la producción y consumo de mensajes.

El módulo de Transacción permite transacciones programáticas y declarativas para las clases que implementan interfaces especiales y para todos los POJO.

Web

La capa web consiste en los módulos Web, Web-Servlet, Web-Struts y Web-Portlet.

El módulo Web permite integración básica de características como la subida multiparte de un fichero, la inicialización de la inversión de control del contenedor usando listeners Servlet y un contexto de lógica web.

El módulo Servlet contiene la implementación modelo vista controlador. El framework Spring MVC permite una separación entre el modelo, el código y los formularios web y se integra con todas las otras características de Spring.

El módulo Web-Struts permite la integración de clases integrando Struts, actualmente este soporte está obsoleto en Spring 3.0.

El módulo Web-Portlet permite las características web en sistemas empotrados.

AOP

El módulo AOP de Spring permite una implementación de programación orientada a aspectos permitiendo definir métodos e interceptores, puntos de corte, etc. Para desacoplar el código.

Permite la integración con AspectJ

El módulo de instrumentación otorga instrumentación de clases así como un cargador de clases a ser usadas en determinadas aplicaciones de servidor.

Test

El módulo de test permite probar las aplicaciones de Spring y los componentes con JUnit o TestNG. Permite la carga consistente de contextos de Spring. Así se permiten objetos mock que prueban tu código de manera aislada. (p. 9-11)

2.9 Calidad de software

(Noriega Quintana, 2007)

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, Mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

Los requisitos del software son la base de las medidas de calidad. La falta de concordancia con los requisitos es una falta de calidad.

Los estándares o metodologías definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se sigue ninguna metodología siempre habrá falta de calidad.

Existen algunos requisitos implícitos o expectativas que a menudo no se mencionan, o se mencionan de forma incompleta (por ejemplo el deseo de un buen mantenimiento) que también pueden implicar una falta de calidad.

La política establecida debe estar sustentada sobre tres principios básicos: tecnológico, administrativo y ergonómico.

El principio tecnológico define las técnicas a utilizar en el proceso de desarrollo del software.

El principio administrativo contempla las funciones de planificación y control del desarrollo del software, así como la organización del ambiente o centro de ingeniería de software.

El principio ergonómico define la interfaz entre el usuario y el ambiente automatizado.

La adopción de una buena política contribuye en gran medida a lograr la calidad del software, pero no la asegura. Para el aseguramiento de la calidad es necesario su control o evaluación.

A partir del siguiente gráfico se observa la interrelación existente entre la Gestión de la Calidad, el Aseguramiento de la Calidad y el Control de la Calidad. (p. 2)

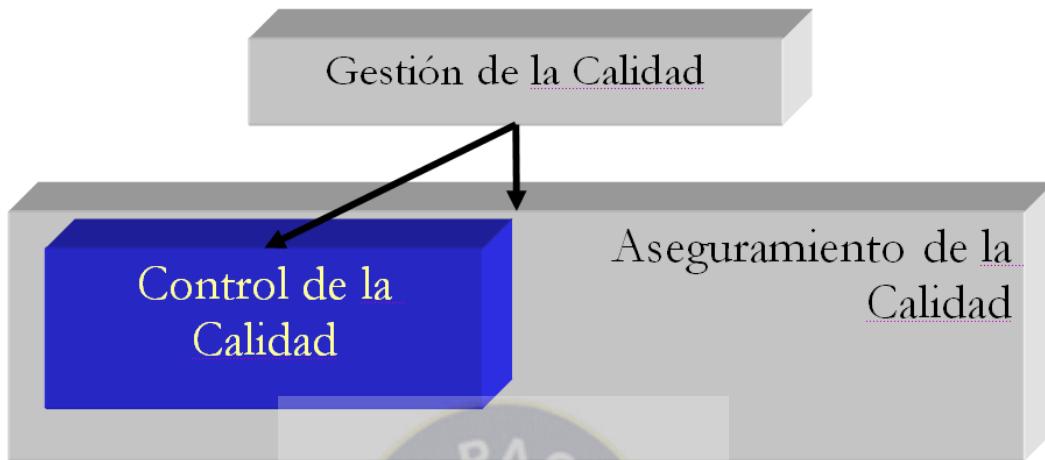


Figura 2.16 Gestión de Calidad (Noriega Quintana, 2007)

2.9.1 ISO 9126

“ISO 9126 es un estándar internacional para la evaluación del Software, fue originalmente desarrollado en 1991 para proporcionar un esquema para la evaluación de calidad del software. La normativa define seis características de la aplicación, estas seis características son divididas en un número de sub-características, las cuales representan un modelo detallado para la evaluación de cualquier sistema informático.” (Smartsys Cia Ltda, 2011)

2.9.1.1 Características

(Smartsys Cia Ltda, 2011) Afirma

El modelo establece diez características, seis que son comunes a las vistas interna y externa y cuatro que son propias de la vista en uso.

A continuación se describen las características y subcaracterísticas propias de este estándar que se encuentran dentro de las vistas interna y externa, las cuales usaremos para evaluar el software de CMI.

Funcionalidad: capacidad del software de proveer los servicios necesarios para cumplir con los requisitos funcionales.

Sub características:

- **Idoneidad.**- Hace referencia a que si el software desempeña las tareas para las cuales fue desarrollado.
- **Exactitud.**- Evalúa el resultado final que obtiene el software y si tiene consistencia a lo que se espera de él.
- **Interoperabilidad.**- Consiste en revisar si el sistema puede interactuar con otro sistema independiente.
- **Seguridad.**- Verifica si el sistema puede impedir el acceso a personal no autorizado.

Fiabilidad: capacidad del software de mantener las prestaciones requeridas del sistema, durante un tiempo establecido y bajo un conjunto de condiciones definidas.

Sub características:

- **Madurez.**- Se debe verificar las fallas del sistema y si muchas de estas han sido eliminadas durante el tiempo de pruebas o uso del sistema.

- **Recuperabilidad.**- Verificar si el software puede reasumir el funcionamiento y restaurar datos perdidos después de un fallo ocasional.
- **Tolerancia a fallos.**- Evalúa si la aplicación desarrollada es capaz de manejar errores.

Usabilidad: esfuerzo requerido por el usuario para utilizar el producto satisfactoriamente.

Sub características:

- **Aprendizaje.**- Determina que tan fácil es para el usuario aprender a utilizar el sistema.
- **Comprendión.**- Evalúa que tan fácil es para el usuario comprender el funcionamiento del sistema
- **Operatividad.**- Determina si el usuario puede utilizar el sistema sin mucho esfuerzo.
- **Atractividad.**- Verifica que tan atractiva se ve la interfaz de la aplicación.

Eficiencia: relación entre las prestaciones del software y los requisitos necesarios para su utilización.

Sub características:

- **Comportamiento en el tiempo.**- Verifica la rapidez en que responde el sistema
- **Comportamiento de recursos.**- Determina si el sistema utiliza los recursos de manera eficiente

Mantenibilidad: esfuerzo necesario para adaptarse a las nuevas especificaciones y requisitos del software.

Sub características:

- **Estabilidad.**- Verifica si el sistema puede mantener su funcionamiento a pesar de realizar cambios.
- **Facilidad de análisis.**- Determina si la estructura de desarrollo es funcional con el objetivo de diagnosticar fácilmente las fallas.
- **Facilidad de cambio.**- Verifica si el sistema puede ser fácilmente modificado
- **Facilidad de pruebas.**- - Evalúa si el sistema puede ser probado fácilmente

Portabilidad: capacidad del software ser transferido de un entorno a otro.

Sub características:

- **Capacidad de instalación.**- Verifica si el software se puede instalar fácilmente
- **Capacidad de reemplazamiento.**- Determina la facilidad con la que el software puede remplazar otro software similar.
- **Adaptabilidad.**- El software se puede trasladar a otros ambientes
- **Co-Existencia.**- El software puede funcionar con otros sistemas

Cada una de las características debe ser evaluada dentro del software basándose en pruebas de funcionamiento, medición de rendimiento y pruebas con usuarios que harán uso del sistema.

2.10 Pruebas de Software

Las **Pruebas de Software**, o Testing es una investigación empírica y técnica cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto bajo pruebas a la parte interesada o Stakeholder. Las **Pruebas de Software** son una actividad más en el proceso de "Aseguramiento de la Calidad"

"Las Pruebas son básicamente un conjunto de actividades dentro del desarrollo de software. Dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento de dicho proceso de desarrollo". (Wikipedia-Pruebas, 2014)

2.10.1 Objetivos

El objetivo de las pruebas es presentar información sobre la calidad del producto a las personas responsables de este.

Teniendo esta afirmación en mente, la información que puede ser requerida es de lo más variada. Esto hace que el proceso de "Testing" sea completamente dependiente del **Contexto** en el que se desarrolla.

A pesar de lo que muchos promueven, no existen las "Mejores Prácticas" como tal. Toda práctica puede ser ideal para una situación pero completamente inútil o incluso perjudicial en otra.

Por esto, las actividades, técnicas, documentación, enfoques y demás elementos que condicionaran las pruebas a realizar, deben ser seleccionados y utilizados de la manera más eficiente según contexto del proyecto.

- **Pruebas Estáticas:** Son el tipo de pruebas que se realizan sin ejecutar el código de la aplicación.
- **Pruebas Dinámicas:** Todas aquellas pruebas que para su ejecución requieren la ejecución de la aplicación.

2.10.2 Prueba unitaria

En programación, una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Luego, con las Pruebas de integración, se podrá asegurar el correcto funcionamiento del sistema o subsistema en cuestión.

La idea es escribir casos de prueba para cada función no trivial o método en el módulo de forma que cada caso sea independiente del resto.

2.10.2.1 Ventajas

El objetivo de las pruebas unitarias es aislar cada parte del programa y mostrar que las partes individuales son correctas. Proporcionan un contrato escrito que el trozo de código debe satisfacer.

(Rodriguez, 2006) Afirma

Asegura calidad del código entregado. Es la mejor forma de detectar errores tempranamente en el desarrollo. No obstante, esto no asegura detectar todos los errores, por tanto prueba de integración y aceptación siguen siendo necesarias.

- Ayuda a definir los requerimientos y responsabilidades de cada método en cada clase probada.

- Constituye una buena forma de ejecutar pruebas de concepto. Cuando es necesario hacer pruebas de conceptos sin integrar usar pruebas unitarias se convierte en un método efectivo.
- Permite hacer refactoring tempranamente en el código. No es necesario todo un ciclo de integración para hacer refactoring en la aplicación, basta con ver cómo se comporta un caso de prueba para hacer refactoring unitario sobre la clase que estamos probando en cuestión.
- Permite incluso hacer pruebas de estrés tempranamente en el código. Por ejemplo un método que realice una consulta SQL que exceda los tiempos de aceptación es posible optimizarla antes de integrar con la aplicación.
- Permite encontrar errores o bugs tempranamente en el desarrollo. Y está demostrado Limitaciones

Es importante darse cuenta de que las pruebas unitarias no descubrirán todos los errores del código. Por definición, sólo prueban las unidades por sí solas. Por lo tanto, no descubrirán errores de integración, problemas de rendimiento y otros problemas que afectan a todo el sistema en su conjunto. Además, puede no ser trivial anticipar todos los casos especiales de entradas que puede recibir en realidad la unidad de programa bajo estudio. Las pruebas unitarias sólo son efectivas si se usan en conjunto con otras pruebas de software.

Herramientas

- **JUnit:** Entorno de pruebas para Java creado por Erich Gamma y Kent Beck. Se encuentra basado en SUnit creado originalmente para realizar pruebas unitarias para el lenguaje Smalltalk.
- **TestNG:** Creado para suplir algunas *deficiencias* en JUnit.
- **JTiger:** Basado en anotaciones, como TestNG.
- **SimpleTests:** Entorno de pruebas para aplicaciones realizadas en PHP.
- **PHPUnit:** Sistema para la realización pruebas unitarias en PHP.
- **CPPUnit:** Versión del framework para lenguajes C/C++.
- **NUnit:** Versión del framework para la plataforma.NET.
- **FoxUnit:** Entorno OpenSource de pruebas unitarias para Microsoft Visual FoxPro
- **MOQ:** Entorno para la creación dinámica de objetos simuladores (mocks).

2.11 El modelo COCOMO

(Dolado) Afirma

El Modelo Constructivo de Costes (Constructive Cost Model) fue desarrollado por B. W. Boehm a finales de los 70 y comienzos de los 80, exponiéndolo detalladamente en su libro "Software Engineering Economics" (Prentice-Hall, 1981). COCOMO es una jerarquía de modelos de estimación de costes software que incluye submodelos básico, intermedio y detallado.

Las ecuaciones de estimación del esfuerzo de desarrollo tienen la forma.

$$E = a_i S^{b_i} m(X)$$

Dónde S es el número de miles de líneas de código fuente, $m(X)$ es un multiplicador que depende de 15 atributos, en la siguiente tabla se muestran los coeficientes para los diferentes modos.

Tabla 2.2 Coeficientes (Dolado)

Modo	Básico		Intermedio	
	a_i	b_i	a_i	b_i
Orgánico	2.4	1.05	3.2	1.05
Semiencajado	3.0	1.12	3.0	1.12
Empotrado	3.6	1.2	2.8	1.2

2.11.1 Modelo Básico

(Dolado) Afirma

Este modelo trata de estimar, de una manera rápida y más o menos burda, la mayoría de proyectos pequeños y medianos. Se consideran tres modos de desarrollo en este modelo: orgánico, semiencajado y empotrado.

2.11.1.1 Modo orgánico

(Dolado) Afirma

En este modo, un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía de unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles de líneas (medio), mientras que en los otros dos modos el tamaño varía de pequeño a muy grandes (varios cientos de miles de líneas). En este modo, al igual que en los otros, el coste se incrementa a medida que el tamaño lo hace, y el tiempo de desarrollo se alarga.

Se utilizan dos ecuaciones para determinar el esfuerzo del personal y el tiempo de desarrollo. El coste es:

$$K_m = 2.4S_k^{1.05}$$

Donde K_m se expresa en personas-mes y S_k es el tamaño expresado en miles de líneas de código fuente. El tiempo de desarrollo se da por:

$$t_d = 2.5K_m^{0.38}$$

Donde K_m se obtiene de la ecuación anterior y t_d es el tiempo de desarrollo en meses.¹

2.11.1.2 Modo Empotrado

“En este modo, el proyecto tiene unas fuertes restricciones, que pueden estar relacionadas con el procesador y el interface hardware. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.” (Dolado)

¹ “Estas ecuaciones se han obtenido por medio de ajustes de curvas realizado por Boehm en TRW sobre 63 proyectos.” (Dolado)

Las estimaciones de tiempo y coste se basan en las mismas ecuaciones que en el modo orgánico, pero con diferentes constantes. Así, el coste se da por

$$K_m = 3.6 S_k^{1.20}$$

Y el tiempo de desarrollo por

$$t_d = 2.5 K_m^{0.32}$$

2.11.1.3 Modo Semiencajado.

“Es un modo intermedio entre los dos anteriores. Dependiendo del problema, el grupo puede incluir una mezcla de personas experimentadas y no experimentadas.” (Dolado)

Las ecuaciones son

$$K_m = 3.0 S_k^{1.12}$$

Y el tiempo de desarrollo por

$$t_d = 2.5 K_m^{0.35}$$

2.11.1.4 Notas al Modelo Básico

(Dolado) Afirma

Se puede observar que a medida que aumenta la complejidad del proyecto, las constantes aumentan de 2.4 a 3.6, que corresponde a un incremento del esfuerzo del personal. Hay que utilizar con mucho cuidado el modelo básico puesto que se obvian muchas características del entorno.

2.11.2 Modelo Intermedio

En este modelo se introducen 15 atributos de coste para tener en cuenta el entorno de trabajo. Estos atributos se utilizan para ajustar el coste nominal del proyecto al entorno real, incrementando la precisión de la estimación.

2.11.2.1 Ecuaciones nominales de coste.

Para cada modo de desarrollo, los 15 atributos del coste intervienen como multiplicadores en el coste nominal, K_n , para producir el coste ajustado.

Las ecuaciones nominales de coste para el modelo intermedio son:

1. Modo orgánico

$$K_n = 3.2 S_k^{1.05}$$

2. Modo semiencajado

$$K_n = 3.0 S_k^{1.12}$$

3. Modo empotrado

$$K_n = 2.8 S_n^{1.20}$$

2.11.2.2 Atributos de coste

Estos atributos tratan de capturar el impacto del entorno del proyecto en el coste de desarrollo. De un análisis estadístico de más de 100 factores que influyen el coste.

Estos atributos se agrupan en cuatro categorías: atributos del producto, atributos del ordenador, atributos del personal y atributos del proyecto.

(Dolado) Afirma

1. Atributos del producto
 - RELY: garantía de funcionamiento requerida al software
 - DATA: tamaño de la base de datos
 - CPLX: complejidad del producto
2. Atributos del ordenador
 - TIME: restricción de tiempo de ejecución
 - STOR: restricción del almacenamiento principal
 - VIRT: volatilidad de la máquina virtual
 - TURN: tiempo de respuesta del ordenador
2. Atributos del personal
 - ACAP: capacidad del analista
 - AEXP: experiencia en la aplicación
 - PCAP: capacidad del programador
 - VEXP: experiencia en máquina virtual
 - LEXP: experiencia en lenguaje de programación
3. Atributos del proyecto
 - MODP: prácticas de programación modernas
 - TOOL: utilización de herramientas software
 - SCED: plan de desarrollo requerido.

Cada atributo se cuantifica para un entorno de proyecto. La escala es muy bajo -- bajo -- nominal -- alto -- muy alto -- extremadamente alto.

2.11.3 Modelo Detallado

Este modelo puede procesar todas las características del proyecto para construir una estimación. Introduce dos características principales

(Dolado) Afirma

1. Multiplicadores de esfuerzo sensitivos a la fase. Algunas fases se ven más afectadas que otras por los atributos. El modelo detallado proporciona un conjunto de multiplicadores de esfuerzo para cada atributo. Esto ayuda a determinar la asignación del personal para cada fase del proyecto.
2. Jerarquía del producto a tres niveles. Se definen tres niveles de producto. Estos son módulo, subsistema y sistema. La cuantificación se realiza al nivel apropiado, esto es, al nivel al que es más susceptible la variación.

2.11.3.1 Estimación del esfuerzo.

(Dolado) Afirma

A. Fases de desarrollo

El desarrollo del software se lleva a cabo a través de cuatro fases consecutivas: requerimientos/planes, diseño del producto, programación y prueba/integración.

Requerimientos/planes. Esta es la primera fase del ciclo de desarrollo. Se analiza el requerimiento, se muestra un Plan de Producto y se genera una especificación completa del producto. Esta fase consume del 6% al 8% del esfuerzo nominal K_n ,

y dura del 10% al 40% del tiempo nominal de desarrollo td. Estos porcentajes dependen del modo y del tamaño (de 2000 LOC a 512000 LOC).

Diseño del producto. La segunda fase del ciclo de desarrollo COCOMO se preocupa de la determinación de la arquitectura del producto y de las especificaciones de los subsistemas. Esta fase requiere del 16% al 18% del esfuerzo nominal Kn, y puede durar del 19% al 38% del tiempo nominal de desarrollo td.

Programación. La tercera fase del ciclo de desarrollo COCOMO se subdivide en dos subfases: diseño detallado y prueba del código. Esta fase requiere del 48% al 68% del esfuerzo nominal Kn, y dura del 24% Al 64% del tiempo nominal de desarrollo.

Prueba/Integración. Esta última fase consiste principalmente en unir las diferentes unidades ya probadas. Se utiliza del 16% al 34% del coste nominal Kn y dura del 18% al 34% del td.

B. Principio de estimación del esfuerzo.

Tamaño equivalente. Como parte del software puede haber sido ya desarrollado, no se requiere entonces un desarrollo completo. En tales casos se estiman las partes de diseño (D%), código (C%) e integración (I%) a ser modificadas. Se calcula un factor de ajuste A

$$A = 0.4 D + 0.3 C + 0.31$$

El tamaño equivalente, S_{equ} es

$$S_{equ} = \frac{(S \cdot A)}{100}$$

Cálculo del esfuerzo. El tamaño equivalente se calcula para cada módulo. El esfuerzo asignado al desarrollo de cada módulo se obtiene entonces a través de:

- 1) seleccionar los valores apropiados de los atributos de coste para cada fase
- 2) multiplicar los atributos de coste para cada módulo y fase, obteniendo un conjunto de 4 multiplicadores globales
- 3) multiplicar los atributos globales por el esfuerzo nominal en cada fase y sumarlos para obtener el esfuerzo total estimado.

2.11.4 Validación independiente del modelo COCOMO

(Dolado) Afirma

El rendimiento de COCOMO en su propia base de datos se indica en la tabla de coeficientes². El COCOMO Básico no funciona bien en su propia base de datos. El Intermedio lo hace bastante mejor.

Es muy difícil evaluar COCOMO Intermedio en otras bases de datos debido a que hay que proporcionar muchos parámetros. El modelo Básico ha sido evaluado según indica la tabla 6.15, dando unos resultados pobres (quizá debido a la falta de datos). A continuación se indican unas fórmulas mejoradas sobre las originales

$$\begin{aligned} E &= 2.6 S^{1.08} m(X) \text{ (modo orgánico)} \\ E &= 2.9 S^{1.12} m(X) \text{ (modo semiencajado)} \\ E &= 2.9 S^{1.20} m(X) \text{ (modo empotrado)} \end{aligned}$$

² Ver Tabla 2.2

3 CAPITULO III

MARCO APPLICATIVO

3.1 Introducción

En este capítulo se explicara en detalle los aspectos relacionados con la funcionalidad, organización, descripción de funciones y diferentes procesos que existe en el departamento de Recursos Humanos de la Empresa de Limpieza industrial TOTE's LTDA.

Posteriormente se detallara los procesos del nuevo sistema mediante la metodología de desarrollo XP y diagramas del lenguaje de modelado web WebML.

3.2 Análisis

3.2.1 Análisis del Sistema actual

El área de recursos Humanos de la empresa de limpieza industrial TOTE's LTDA cuenta con un programa elaborado en fox pro con el cual lleva el registro de sus funcionarios, así como la emisión de pre-planillas de pago para que sean aprobadas por el departamento de contabilidad. El registro de asistencia en la oficina central es mediante tarjetas, mientras que en los subcontratos se lleva el control por medio de un reporte de asistencia elaborado por el supervisor.

El jefe de recursos humanos es el encargado de realizar las planillas, implementados en Excel y para ello tiene un pequeño programa de escritorio que realiza el mismo.

Actualmente como ya se especificó en los antecedentes de la institución los procesos son realizados manualmente: la asistencia se realiza mediante el llenado de una hoja de asistencia.

3.2.2 Declaración de propósitos

El propósito del Sistema Web de control y administración de Recursos Humanos es la centralización de la información para realizar un seguimiento y control, luego procesarlas, consultarlas y desplegar los informes en reportes que muestren una información clara y concisa.

Se ha optado por la metodología extrema XP ya que esta se ajusta a las características del proyecto y se basa en la metodología de modelado WebML.

A continuación se especifica las prácticas aplicable al Sistema Web de control y Administración de recursos Humanos CASO: Empresa de Limpieza Industrial TOTE's LTDA.

3.3 Valores de la Metodología XP

3.3.1 Planificación incremental

Se decidió realizar los siguientes módulos con iteraciones que se especifican más adelante y que a groso modo serán:

- a) **Registro de personal:** Registro del personal administrativo y de planta de la empresa.
- b) **Control de asistencia:** Gestión de usuarios, control de asistencia del personal.
- c) **Registro de documentación:** Registro de documentación como ser: Certificados de antecedentes, certificados médicos, documentos de garantía, etc.
- d) **Evaluación al personal, Informes, Memorándums:** En caso de nuevas designaciones se emitirán reportes de asistencia, así como también la emisión de memorándums en forma automática.

3.3.2 Pruebas

Una de las principales aportaciones de esta metodología es el concepto de desarrollo por tests (pruebas). Los tests son realizados a priori con el fin de prevenir errores, no de solucionarlos. Esto confiere una gran cantidad de software resultante.

Los tests son ejecutados automáticamente cada día para asegurar que el sucesivo desarrollo del sistema no afecta a su estabilidad. Dado que los tests son creados a priori fallaran hasta que la funcionalidad esté implementada, en el proceso que comúnmente se llama rojo a verde. En caso de que se encuentren fallos no detectados previamente.

3.3.3 Programación en parejas

El presente trabajo será realizado solo por una persona la cual es responsable del desarrollo del presente proyecto, pero mediante el constante intercambio de opiniones y trabajo conjunto con el personal encargado de recursos humanos, por lo tanto conoce de lo que el código hará, aunque no será un involucrado a profundidad, se convierte y/o será como si fuera el colaborador logístico.

3.3.4 Refactorización

En las sucesivas iteraciones ha sido necesario refactorizar partes del núcleo del sistema y este proceso ha sido realmente sencillo, a lo que ha contribuido en gran parte la cantidad de tests realizados.

3.3.5 Diseño simple

El diseño se ha mantenido sencillo, desde luego pasando los tests, sin código duplicado y un mínimo de código gracias al framework spring que proporciona un gran dinamismo y evita la necesidad de operaciones repetitivas.

3.3.6 Propiedad colectiva del código

Como no existe un equipo de desarrollo como tal y por ser un proyecto de grado individual, la propiedad del código es netamente del autor.

3.3.7 Integración continua

La integración es permanente gracias a los tests de integración, porque al realizar los tests se elimina por completo los errores que puede tener el sistema, hasta que funcione correctamente el sistema.

3.3.8 Cliente en el equipo

Aunque esto estrictamente no se realiza dado la naturaleza del proyecto, la experiencia del autor como personal antiguo de la empresa ayudo a entender los procesos a automatizar, además que el cliente siempre estuvo en contacto con el autor mediante correo electrónico y chat.

3.3.9 Pequeñas entregas

Se ha seguido esta práctica, liberando nuevas versiones según la funcionalidad se han ido implementando los módulos restantes.

3.3.10 Semanas de 40 horas

Promediando las horas de trabajo entre el tiempo total de desarrollo se cumple las 40 horas de trabajo semanales.

3.3.11 Estándares de codificación

Se ha seguido el estándar de codificación sugerido por el framework spring 3.1 con el lenguaje de programación JSP y el gestor de base de datos PostgreSQL. Además, debido a que el proyecto es individual no se tuvo problemas con el estándar en la codificación.

3.4 Fase I: Exploración

La primera fase en el ciclo de vida de eXtreme Programming es la exploración, fase en la que se plantean las historias de usuario (user stories) al mismo tiempo que el equipo de desarrollo se familiarizara con las herramientas, tecnologías y prácticas que se utilizan en el proyecto. Se prueba la tecnología y se explora las posibilidades de la arquitectura del sistema construyendo un prototipo.

3.4.1 Historias de Usuario

Las historias de usuario son la técnica utilizada en XP para especificar los requerimientos del usuario que debe de cumplir el software a desarrollar, lo que equivaldría a los casos de uso en el proceso unificado.

El modelo de casos de uso es la interacción típica en el usuario y el sistema computacional, un caso de uso muestra el comportamiento deseado del sistema.

Se describe brevemente las características que el sistema debe tener desde la perspectiva del cliente.

Las Historias de usuario definidas para este proyecto son:

Administración de usuarios

El módulo se puede dividir en las siguientes historias de usuario y sus respectivas tareas:

Historia 1

Tabla 3.1 Historia de Usuario - registro de un nuevo Usuario

Historia de Usuario	
Número: 1	Usuario: Administrador
Nombre de Historia: Registro de un nuevo usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta (Alta / Media / Baja)
Puntos estimados: 2	Iteración asignada: 1
Descripción: Solo debe de asignarse como usuario a un funcionario activo que esté trabajando como Jefe de Recursos Humanos, Administrador de Sistema, o Supervisor	
Observaciones: CONFIRMADO por el cliente	

Tareas

Tabla 3.2 Diseñar la estructura de datos para registrar la información del usuario y funcionario

Tarea	
Número de tarea: 1.1	Número de historia: 1
Nombre de tarea: Diseñar la estructura de datos para registrar la información del usuario y funcionario	
Tipo de tarea: Diseño	Puntos estimados: ½
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monroy Castillo	
Descripción: Se realiza el diseño de la base de datos para la información del funcionario y del usuario.	

Tabla 3.3 Tarea - Registro de datos del usuario

Tarea	
Número de tarea: 1.2	Número de historia: 1
Nombre de tarea: Registro de datos del usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 1 ½
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monroy Castillo	
Descripción: Se desarrolla una interfaz para que se busque la información de un funcionario para luego poderlo habilitar como usuario.	

Tabla 3.4 Tarea - Habilitación de solo funcionarios activos para ser asignados como usuarios

Tarea	
Número de tarea: 1.3	Número de historia: 1
Nombre de tarea: Habilitación de solo funcionarios activos para ser asignados como usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: ½
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monroy Castillo	
Descripción: Se desarrolla una interfaz para que se muestren solo a los funcionarios activos para que solo estos puedan ser habilitados como usuarios del sistema.	

Administración de funcionarios

El módulo se puede dividir en las siguientes historias de usuario y sus respectivas tareas:

Historia 2

Tabla 3.5 Historia de usuario - Registro de funcionarios

Historia de Usuario	
Número: 2	Usuario: Jefe de Recursos Humanos
Nombre de Historia: Registro de funcionarios	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta (Alta / Media / Baja)
Puntos estimados: 2	Iteración asignada: 1
Descripción: Diseñar el módulo de registro de funcionarios, y verificar si el funcionario a registrar ya estaba trabajando con anterioridad en la empresa	
Observaciones: CONFIRMADO por el cliente	

Tareas

Tabla 3.6 Diseño de la base de datos para la información del funcionario

Tarea	
Número de tarea: 2.1	Número de historia: 2
Nombre de tarea: Diseño de la base de datos para la información del funcionario	
Tipo de tarea: Diseño	Puntos estimados: ½
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monroy Casillo	
Descripción: Se diseña la tabla que corresponde a la información del funcionario.	

Tabla 3.7 Tarea - Modulo de registro de información de un funcionario

Tarea	
Número de tarea: 2.2	Número de historia: 2
Nombre de tarea: Modulo de registro de información de un funcionario	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monrroy Castillo	
Descripción: Se desarrolla el modulo para el registro y habilitación de un nuevo funcionario.	

Tabla 3.8 Tarea - Módulo de re afiliación de un ex funcionario

Tarea	
Número de tarea: 2.3	Número de historia: 2
Nombre de tarea: Modulo de re afiliación de un ex funcionario	
Tipo de tarea: Desarrollo	Puntos estimados: ½
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monrroy Castillo	
Descripción: Se desarrolla el modulo para re afiliar a un funcionario que ya tiene almacenado datos como un ex funcionario.	

Historia 3 (Ver Anexos)

3.4.2 Detalle de las historias de usuario

A continuación se detalla un listados de las historias de usuario por módulos que describieron en el punto anterior del presente proyecto, además de hacer una correspondencia con las tareas que se tiene para cada historia de usuario, así se pretende presentar un marco entendible de las historias de usuario y poder realizar la planificación posteriormente como lo señala la metodología XP.

3.4.2.1 Módulo de administración de usuarios

- **Historia de usuario 1:** Registro de usuarios

- **Tarea 1:** Diseñar la estructura de datos para registrar la información del usuario y funcionario
- **Tarea 2:** Registro de datos del usuario
- **Tarea 3:** Habilitación de solo funcionarios activos para ser asignados como usuarios

3.4.2.2 Módulo de administración de funcionarios.

- **Historia de usuario 2:** Registro de funcionarios
 - **Tarea 1:** Diseño de la base de datos para la información del funcionario
 - **Tarea 2:** Modulo de registro de información de un funcionario
 - **Tarea 3:** Modulo de re afiliación de un ex funcionario
- **Historia de usuario 3³:** Registro de contratos
 - **Tarea 1:** Diseño de la base de datos para la información del contrato
 - **Tarea 2:** Modulo de registro de información de un contrato
 - **Tarea 3:** Modulo de re apertura de un contrato cerrado
- **Historia de usuario 4:** Verificar el estado de la documentación de un funcionario
 - **Tarea 1:** Diseño de la base de datos para la información de la documentación de un funcionario
 - **Tarea 2:** Modulo de registro de documentos del funcionario
 - **Tarea 3:** Modulo de descarga de documentos digitalizados

3.4.2.3 Módulo de administración y control de asistencias

- **Historia de usuario 5:** Verificar el estado de las asistencias de los funcionarios
 - **Tarea 1:** Diseño de la base de datos para la información de la asistencia de un funcionario
 - **Tarea 2:** Modulo de verificación de asistencia del funcionario por contratos y general
 - **Tarea 3:** Modulo de re afiliación de un ex funcionario
- **Historia de usuario 6:** Registrar asistencias de los funcionarios
 - **Tarea 1:** Diseño del formulario para el registro de las asistencias
 - **Tarea 2:** Modulo de registro de asistencias

3.4.2.4 Módulo de generación de reportes

- **Historia de usuario 7:** Generar reportes de planilla de sueldos por contrato
 - **Tarea 1:** Diseño del reporte de la planilla por contratos
 - **Tarea 2:** Modulo de generación de reportes de planilla de sueldos

3.5 Fase II: Planificación

Para el desarrollo del proyecto se ha planificado cuatro iteraciones en los cuales se entregaran diferentes versiones del sistema web. La cantidad y el orden en las iteraciones de las historias de usuario así como las estimaciones de esfuerzo se detallan a continuación, de las cuales algunas han sido modificándose en el transcurso del desarrollo del proyecto y otras fueron eliminadas o reagrupadas debido a nuevos requerimientos.

³ Ver anexos

3.5.1 Estimaciones de esfuerzo

- Administración de Usuarios

Tabla 3.9 Administración de usuarios

Historias de usuario	Pts.
Registro de usuarios	2

- Administración de funcionarios

Tabla 3.10 Administración de funcionarios

Historias de usuario	Pts.
Registro de funcionarios	2
Registro de contratos	2
Verificar el estado de la documentación de un funcionario	3 ½

- Administración y Control de Asistencias

Tabla 3.11 Administración y control de asistencias

Historias de usuario	Pts.
Verificar el estado de las asistencias de los funcionarios	3
Registrar asistencias de los funcionarios	3

- Generación de Reportes

Tabla 3.12 Generación de reportes de planillas

Historias de usuario	Pts.
Generar reportes de planilla de sueldos por contrato	3

3.5.2 Planificación de iteraciones

Tabla 3.13 Cronograma en detalle de las historias de usuario

ITERACIONES	Nº	HISTORIAS	INICIO	FIN	OBSERVACIÓN
Primera	1	Registro de usuarios	05/09/2013	09/10/2013	
	2	Registro de funcionarios	12/10/2013	16/10/2013	
	3	Registro de contratos	19/10/2013	23/10/2013	
Segunda	4	Verificar el estado de la documentación de un funcionario	28/10/2013	13/12/2013	
Tercera	5	Verificar el estado de las asistencias	16/01/2014	27/01/2014	

		de los funcionarios			
	6	Registrar asistencias de los funcionarios	30/01/2014	19/02/2014	
Cuarta	7	Generar reportes de planilla de sueldos por contrato	20/02/2014	26/03/2014	



Figura 3.1: Cronograma de desarrollo por iteraciones

3.6 Fase III: Iteraciones

3.6.1 Primera iteración

Historia de usuario 1: Registro de un nuevo usuario

Tarea 1.1: Diseñar la estructura de datos para registrar la información del usuario y funcionario

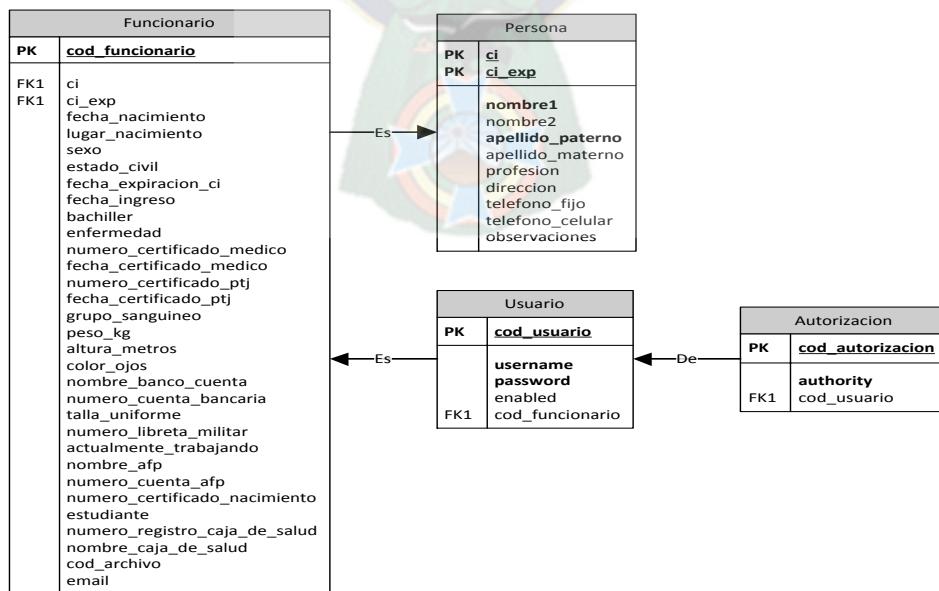


Figura 3.2: Estructura de datos para registrar la información del usuario y funcionario

Tarea 1.2: Diseñar la interfaz para que se busque la información de un funcionario para luego poderlo habilitar como usuario.

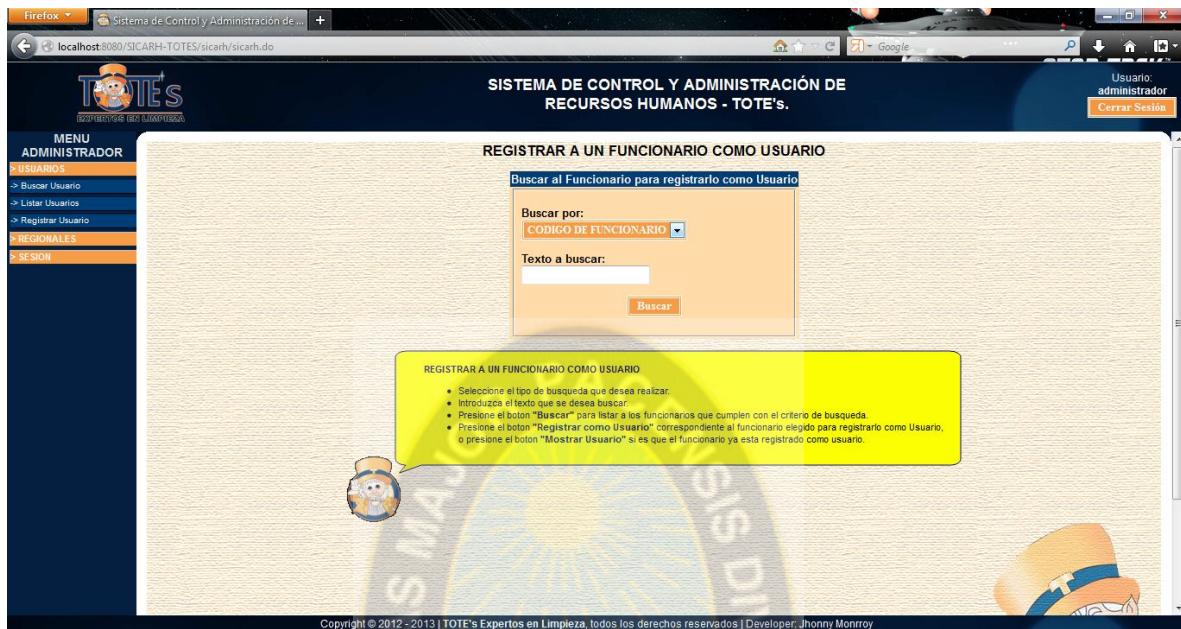


Figura 3.3: Pantalla de búsqueda de funcionarios para posterior registro como funcionario

Tarea 1.3: Habilitación de solo funcionarios activos para ser asignados como usuarios

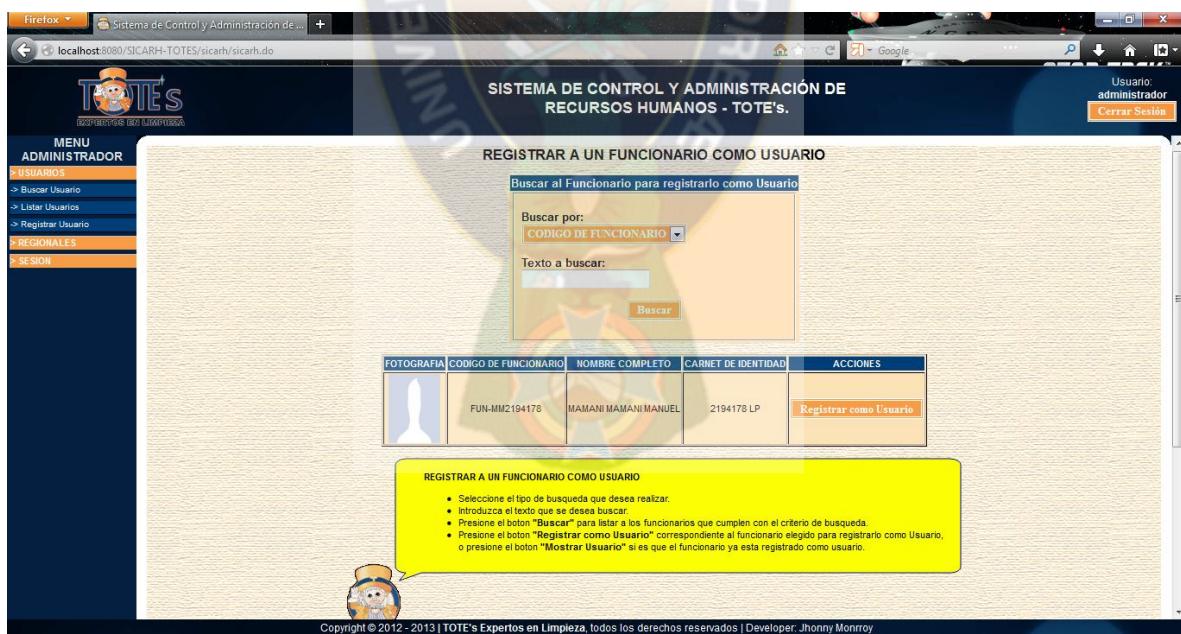


Figura 3.4: Listado de funcionarios encontrados

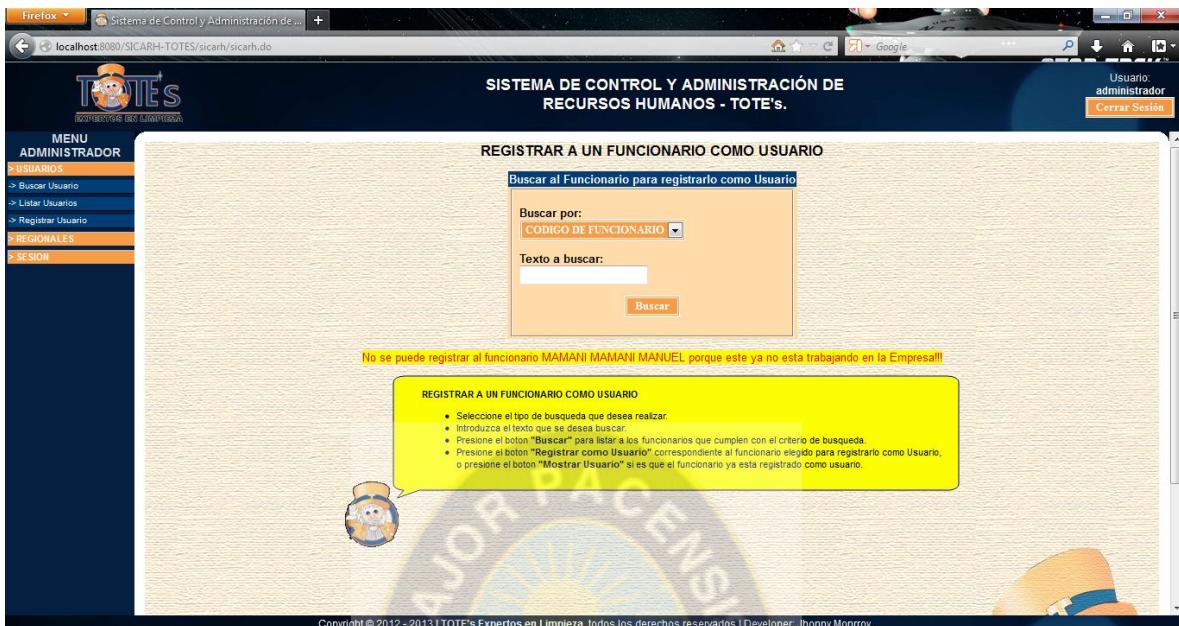


Figura 3.5: Mensaje de que no se puede registrar como usuario porque no se trata de un funcionario activo

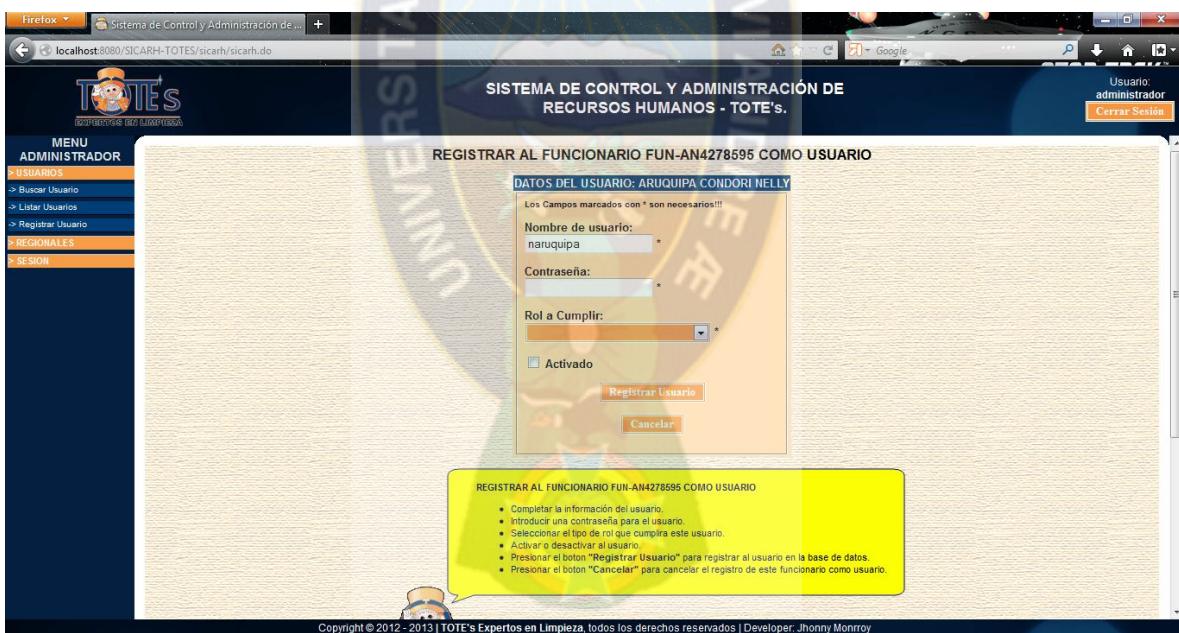


Figura 3.6: Pantalla de registro de usuario en caso de que el funcionario este en estado activo

Pruebas de aceptación para la Historia de usuario 1

a. Identificar todos los posibles resultados de la historia:

- Interfaz para la búsqueda del funcionario
- Resultado y listado de funcionarios encontrados
- Interfaz cuando un funcionario no puede ser activado como usuario
- Interfaz cuando el funcionario está habilitado para registrarlo como usuario.

b. Identificar los resultados que terminan la historia y los que permiten continuar dentro la historia:

- La historia termina cuando se termina de registrar a un funcionario como usuario.
- La historia termina cuando se pulsa el botón cancelar en cualquiera de sus interfaces.
- La historia no permite continuar en ella, para registrar a otro usuario se debe iniciar el proceso desde el principio de la historia.

c. Identificar los caminos de ejecución posibles:

- Comienza la historia cuando el administrador decide registrar a un nuevo usuario en el sistema, si el funcionario es válido se le asigna un nombre de usuario y una contraseña además de elegir el tipo de privilegios que tendrá este usuario,
- Se valida los datos del funcionario y usuario.
- Se almacena y actualiza esta información en la base de datos.
- La historia termina cuando el usuario ha sido registrado exitosamente.

d. Asignar un conjunto de valores válidos y valores de entorno a cada camino de ejecución para obtener el resultado esperado:

- El conjunto de valores válidos está dado por la información ya almacenada de los funcionarios.

e. Eliminación de caminos redundantes:

- No se encontraron caminos redundantes.

Historia de usuario 2: Registro de funcionarios

Tarea 2.1: Diseño de la base de datos para la información del funcionario

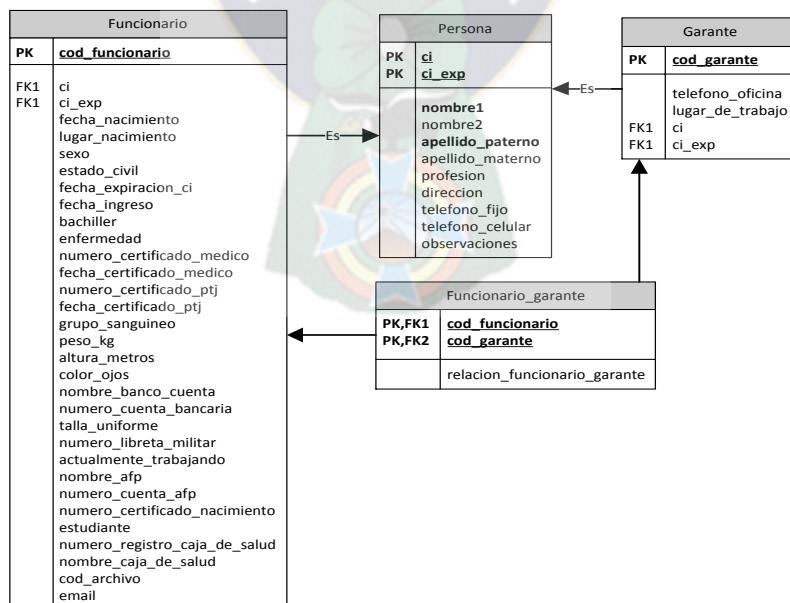


Figura 3.7: Estructura de datos para registrar la información del funcionario

Tarea 2.2: Modulo de registro de información de un funcionario

Figura 3.8: Pantalla de inicio para verificar la existencia de un funcionario

Figura 3.9: Pantalla para el registro de información del funcionario

Tarea 2.3: Modulo de re afiliación de un ex funcionario

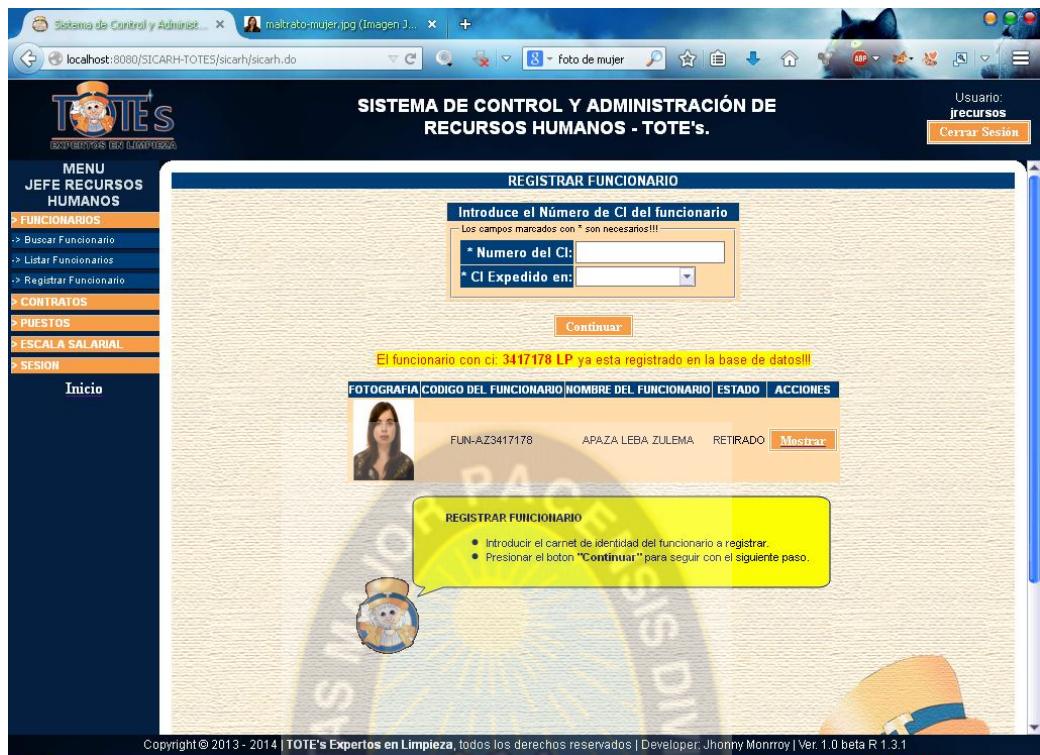


Figura 3.10: Pantalla de anuncio de que el funcionario tiene datos almacenados



Figura 3.11: Pantalla para lanzar el módulo de re afiliación del funcionario

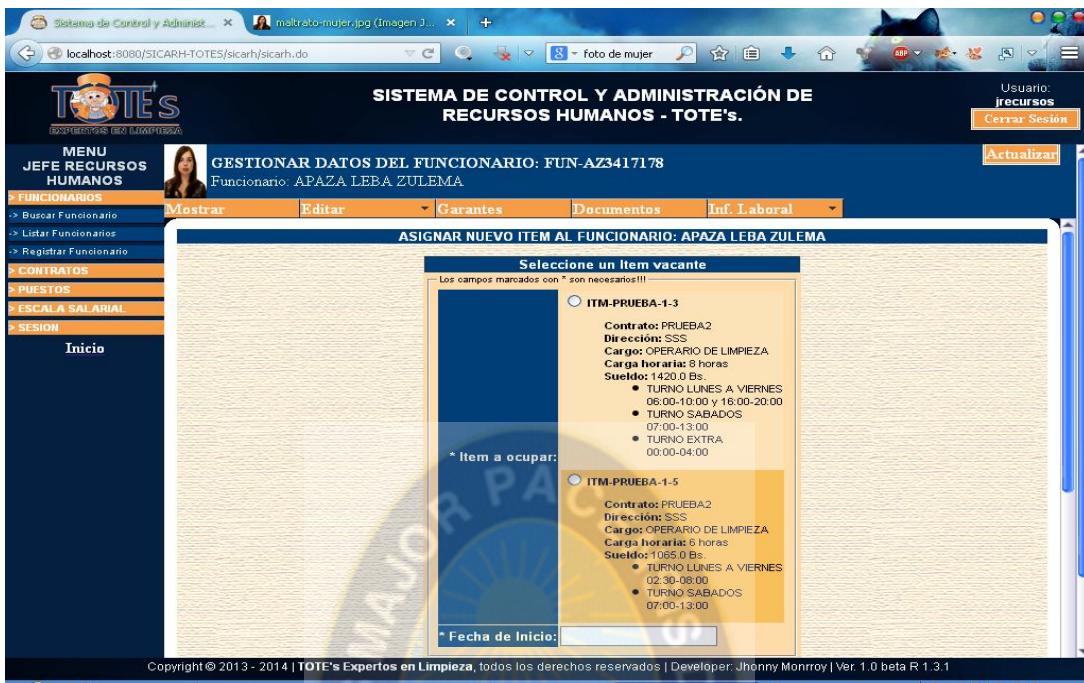


Figura 3.12: Modulo para la re afiliación del funcionario

Pruebas de aceptación para la Historia de usuario 2

a. Identificar todos los posibles resultados de la historia:

- Interfaz para la búsqueda de registros mediante su Documento de identidad.
- Resultado de la búsqueda y listado de funcionarios ya registrados si existieran
- Interfaz cuando un funcionario ya está registrado en el sistema
- Interfaz cuando el funcionario ya está registrado como garante de otro funcionario.

b. Identificar los resultados que terminan la historia y los que permiten continuar dentro la historia:

- La historia termina cuando se termina de registrar a un funcionario.
- La historia termina cuando se pulsa el botón cancelar en cualquiera de sus interfaces.
- La historia no permite continuar en ella, para registrar a otro funcionario se debe iniciar el proceso desde el principio de la historia.

c. Identificar los caminos de ejecución posibles:

- Comienza la historia cuando el jefe de recursos humanos decide registrar a un nuevo funcionario en el sistema, si el funcionario es no está registrado se procede a mostrar la interfaz de llenado de datos.
- Si el funcionario ya está registrado se procede a mostrar su interfaz de edición.
- Si el funcionario ya está registrado como garante se procede a mostrar la interfaz para completar la información faltante para que sea registrado como funcionario.
- Se almacena y actualiza esta información del funcionario en la base de datos.
- La historia termina cuando el funcionario ha sido registrado exitosamente.

d. Asignar un conjunto de valores válidos y valores de entorno a cada camino de ejecución para obtener el resultado esperado:

- El conjunto de valores válidos está dado por la información ya almacenada de los funcionarios y los introducidos por el jefe de recursos humanos.

e. Eliminación de caminos redundantes:

- No se encontraron caminos redundantes.

Historia de usuario 3: Registro de contratos.

Tarea 3.1: Diseño de la base de datos para la información del contrato.

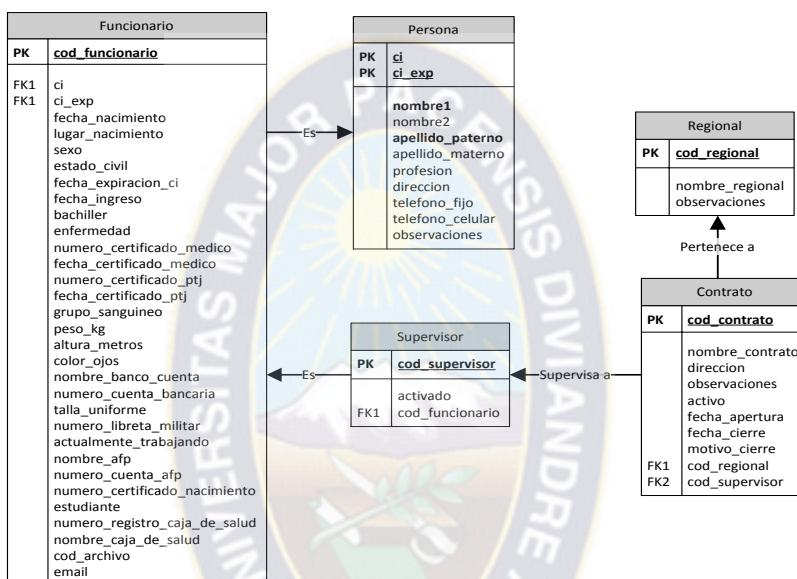


Figura 3.13: Estructura de datos para registrar la información de un contrato

Tarea 3.2: Modulo de registro de información de un contrato.

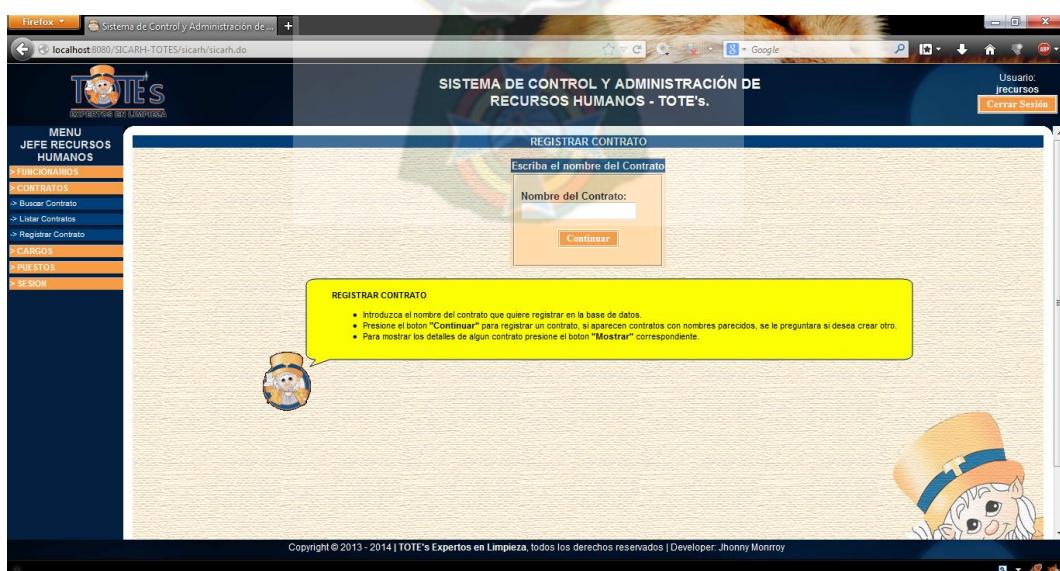


Figura 3.14: Modulo para el registro de un nuevo contrato

SISTEMA DE CONTROL Y ADMINISTRACIÓN DE RECURSOS HUMANOS - TOTE's.

REGISTRAR CONTRATO

DATOS DEL CONTRATO

* Nombre del Contrato: SENAPE
 * Regional a la que pertenece el contrato: LA PAZ
 * Dirección:
 * Número de Operarios:
 * Número de Encargados:
 Observaciones:

Registrar Contrato
 Cancelar

REGISTRAR CONTRATO

- Llenar los datos requeridos del contrato
- Presionar el botón "Registrar Contrato" para Registrar el contrato en la base de datos
- Presionar el botón "Cancelar" para cancelar el registro del contrato.

Figura 3.15: Modulo para el llenado de datos de un nuevo contrato

SISTEMA DE CONTROL Y ADMINISTRACIÓN DE RECURSOS HUMANOS - TOTE's.

GESTIONAR DATOS DEL CONTRATO: CON-SENAPE

Contrato: SENAPE

Mostrar Editar

DATOS DEL CONTRATO

Código de contrato: CON-SENAPE
 Nombre del Contrato: SENAPE
 Regional: LA PAZ
 Dirección del Contrato: PLAZA DEL STADIUM
 Estado: CONTRATO CERRADO

No se encontraron puestos registrados para este contrato!!!

Aquí se muestra la información del contrato: SENAPE

Figura 3.16: Modulo para el registro de un nuevo contrato pantalla de contrato registrado

Tarea 3.3: Modulo de re apertura de un contrato cerrado.

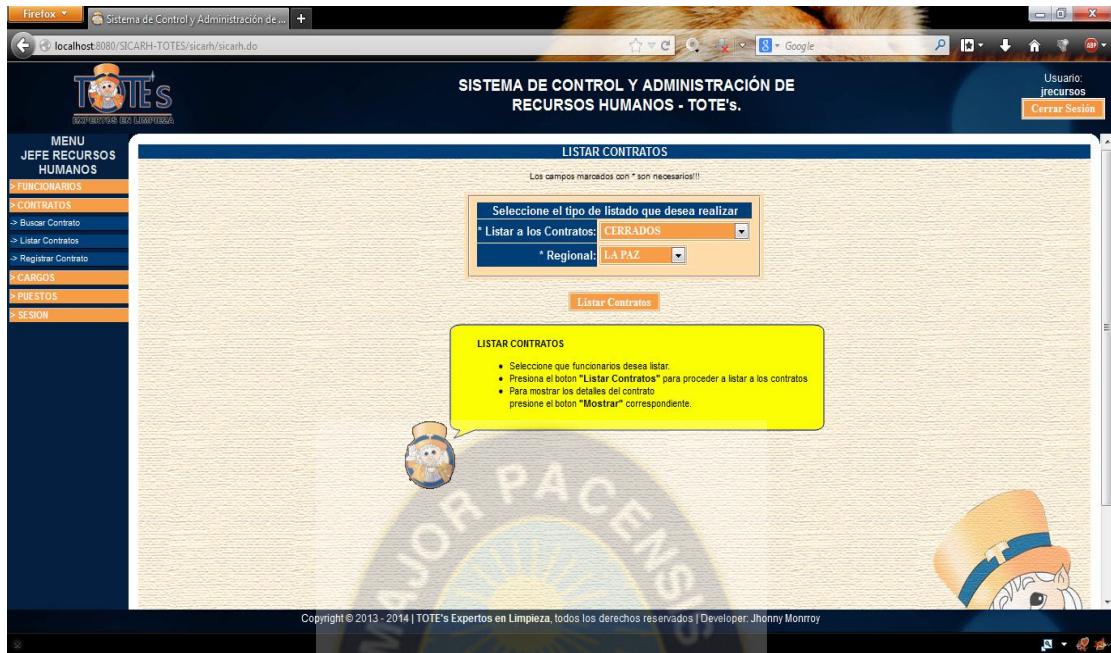


Figura 3.17: Pantalla de búsqueda de un contrato cerrado

The screenshot shows the same application interface as Figure 3.17, but now displaying a list of contracts. The title is "LISTAR CONTRATOS". The table below lists five contracts:

#	LOGO	NOMBRE DEL CONTRATO	DIRECCIÓN	ESTADO	ACCIONES
1		BANCO BISA - SUCURSAL AVENIDA CAMACHO	AV. CAMACHO	CONTRATO CERRADO	<button>Mostrar</button>
2		BANCO DE CRÉDITO - LA PAZ	CALLE MERCADO A UNA CUADRA DE LA ALCALDIA DE LA PAZ	CONTRATO CERRADO	<button>Mostrar</button>
3		COMIBOL ARCHIVOS	Zona Ferropetrol Ciudad del Alto	CONTRATO CERRADO	<button>Mostrar</button>
4		MUTUAL LA PAZ	AV 16 DE JULIO ESQUINA CALLE BUENO	CONTRATO CERRADO	<button>Mostrar</button>
5		MUTUAL LA PRIMERA - CENTRAL LA PAZ	PRADO DE LA CIUDAD DE LA PAZ	CONTRATO CERRADO	<button>Mostrar</button>

At the bottom, there is a copyright notice: "Copyright © 2013 - 2014 | TOTE's Expertos en Limpieza, todos los derechos reservados | Developer: Jhony Monroy".

Figura 3.18: Pantalla de listados de contratos cerrados

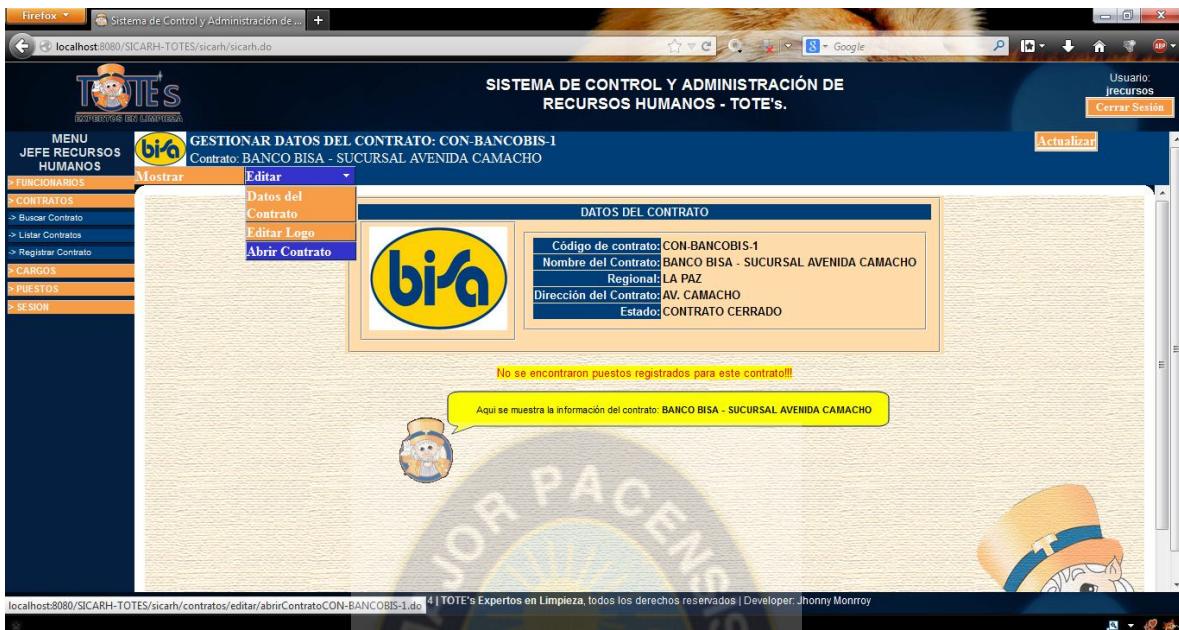


Figura 3.19: Pantalla de interfaz de apertura de un contrato

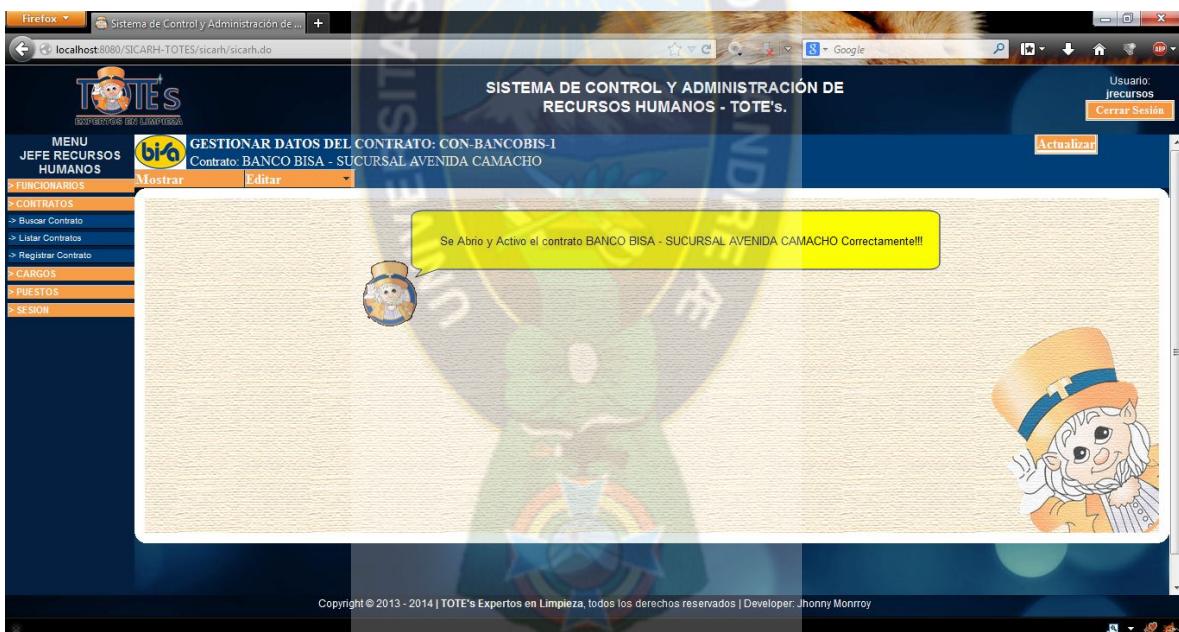


Figura 3.20: Pantalla de aviso de que el contrato fue abierto correctamente

Pruebas de aceptación para la Historia de usuario 3

- Identificar todos los posibles resultados de la historia:**
 - Interfaz para el registro de un contrato mediante su nombre.
 - Interfaz cuando un contrato de nombre similar ya está registrado en el sistema.
- Identificar los resultados que terminan la historia y los que permiten continuar dentro la historia:**
 - La historia termina cuando se termina de registrar a un contrato.

- La historia termina cuando se pulsa el botón cancelar en cualquiera de sus interfaces.
- La historia no permite continuar en ella, para registrar a otro contrato se debe iniciar el proceso desde el principio de la historia.

c. Identificar los caminos de ejecución posibles:

- Comienza la historia cuando el jefe de recursos humanos decide registrar a un nuevo contrato en el sistema, si el contrato es no está registrado se procede a mostrar la interfaz de llenado de datos.
- Si el contrato ya está registrado con un nombre similar se procede a mostrar su interfaz de edición.
- Se almacena y actualiza esta información del contrato en la base de datos.
- La historia termina cuando el contrato ha sido registrado exitosamente.

d. Asignar un conjunto de valores válidos y valores de entorno a cada camino de ejecución para obtener el resultado esperado:

- El conjunto de valores válidos está dado por la información ya almacenada de los contratos y los introducidos por el jefe de recursos humanos.

e. Eliminación de caminos redundantes:

- No se encontraron caminos redundantes.

3.6.2 Segunda Iteración

Historia de usuario 4: Verificar el estado de la documentación de un funcionario.

Tarea 4.1: Diseño de la base de datos para la información de la documentación de un funcionario.

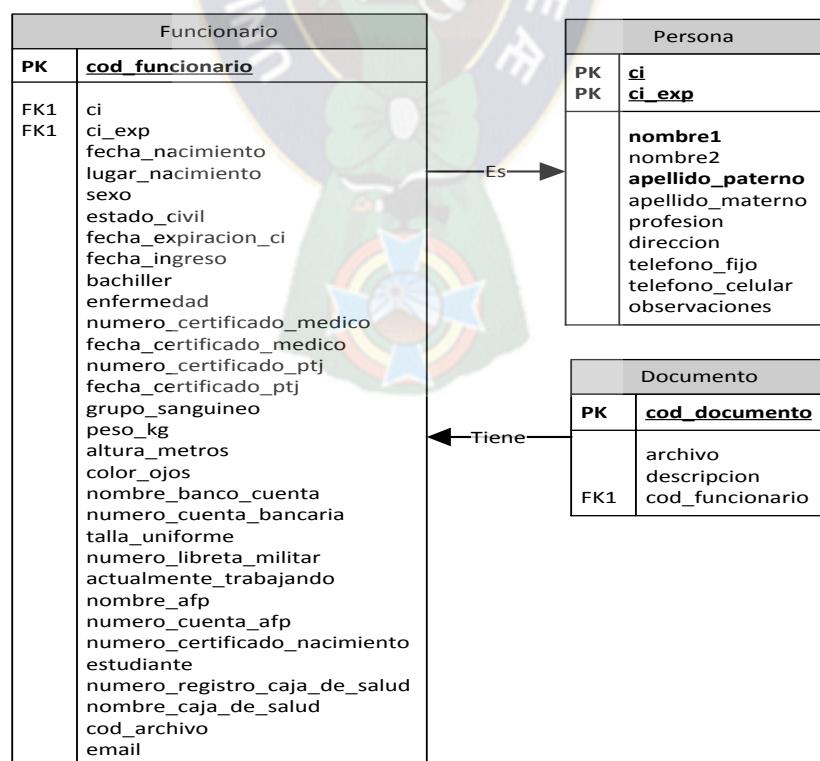


Figura 3.21: Diseño de la tabla para registro de documentos del funcionario

Tarea 4.2: Modulo de registro de documentos del funcionario.

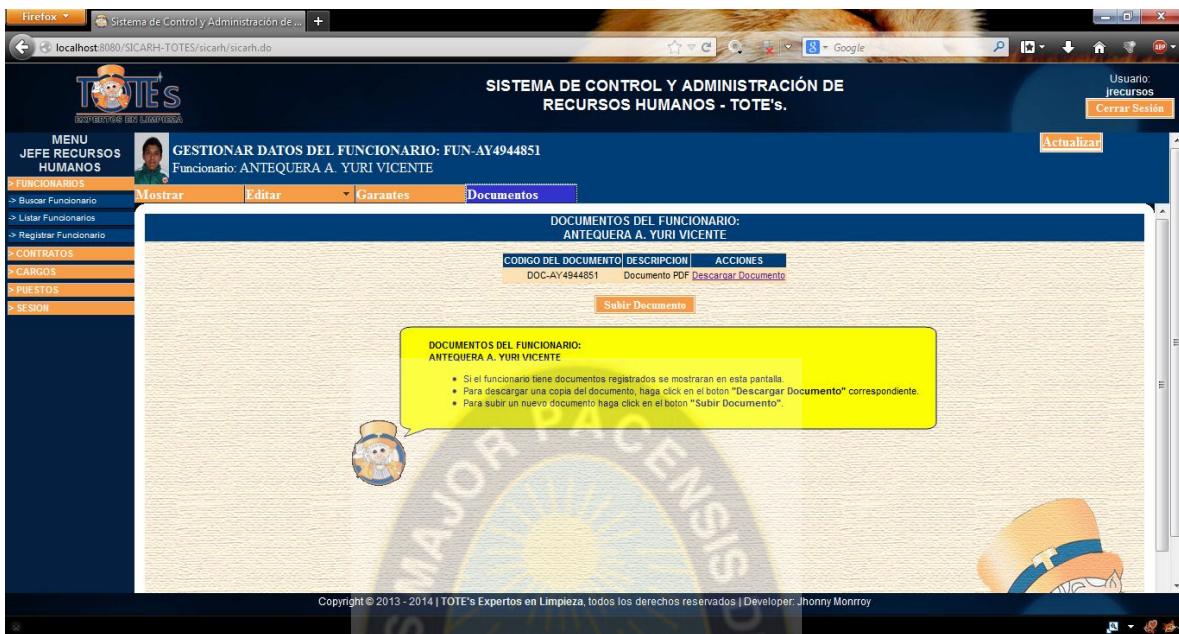


Figura 3.22: Pantalla de listado de documentos del funcionario

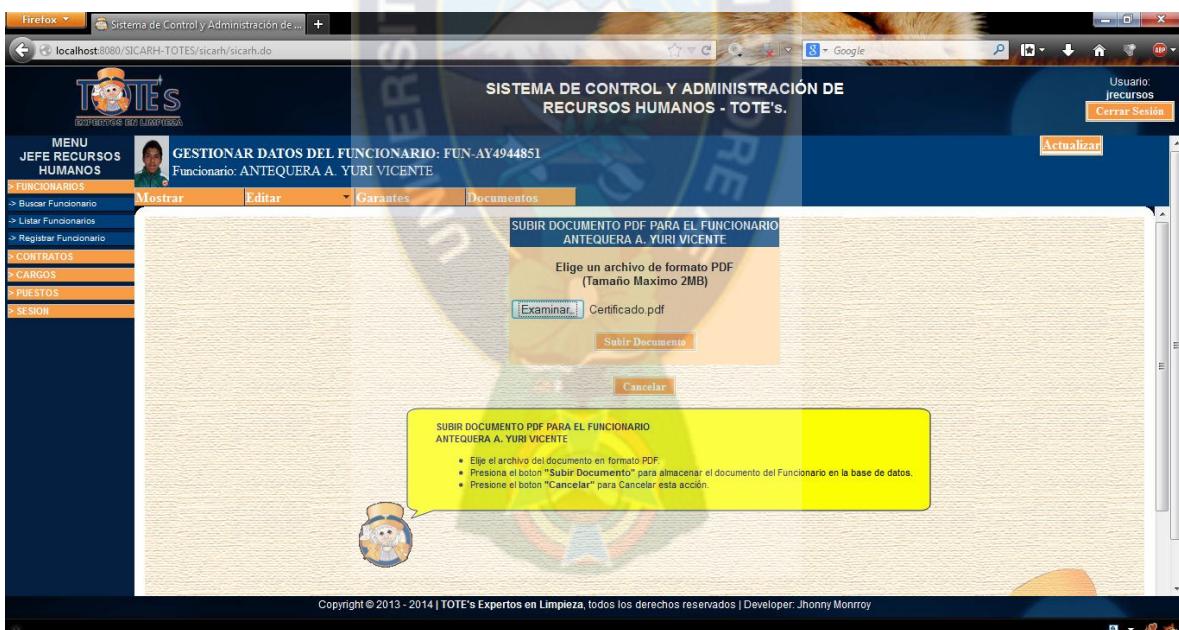


Figura 3.23: pantalla para subir y almacenar un documento en el sistema

Tarea 4.3: Modulo de descarga de documentos digitalizados.

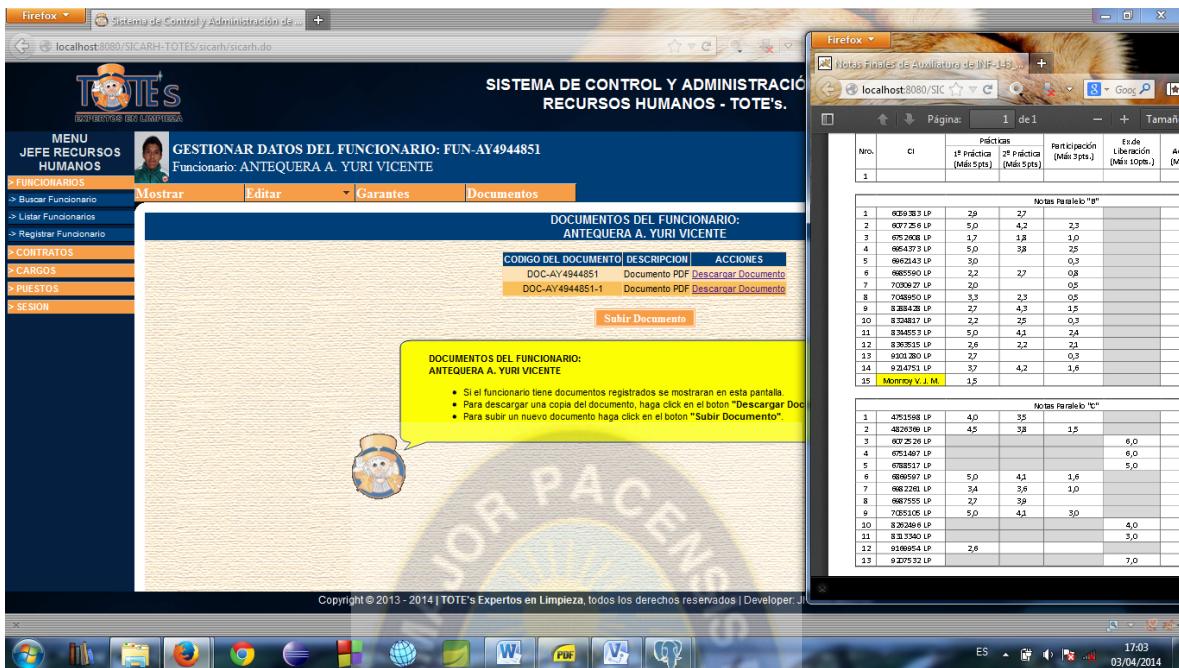


Figura 3.24: Pantalla de descarga de documentos digitalizados

Pruebas de aceptación para la Historia de usuario 4

- a. Identificar todos los posibles resultados de la historia:**
 - Interfaz para el listado de documentos de un funcionario.
 - Interfaz de carga y subida de documentos al sistema.
 - Interfaz de descarga de documentos almacenados en la base de datos.
 - b. Identificar los resultados que terminan la historia y los que permiten continuar dentro la historia:**
 - La historia termina cuando se sube un documento al sistema.
 - La historia termina cuando se descarga un documento digitalizado.
 - La historia termina cuando se pulsa el botón cancelar en cualquiera de sus interfaces.
 - La historia permite continuar en ella mientras se liste los documentos almacenados.
 - c. Identificar los caminos de ejecución posibles:**
 - Comienza la historia cuando el jefe de recursos humanos decide listar los documentos registrados de un funcionario.
 - Si el Jefe de Recursos Humanos decide subir un documento, se muestra la interfaz correspondiente.
 - Si el Jefe de Recursos Humanos decide descargar un documento, este se descargara automáticamente.
 - La historia termina cuando decide continuar con otro modulo.
 - d. Asignar un conjunto de valores válidos y valores de entorno a cada camino de ejecución para obtener el resultado esperado:**
 - El conjunto de valores válidos está dado por la información ya almacenada de los funcionarios y los introducidos por el jefe de recursos humanos.

e. Eliminación de caminos redundantes:

- No se encontraron caminos redundantes.

3.6.3 Tercera Iteración

Historia de usuario 5: Verificar el estado de las asistencias de los funcionarios.

Tarea 5.1: Diseño de la base de datos para la información de la asistencia de un Funcionario.

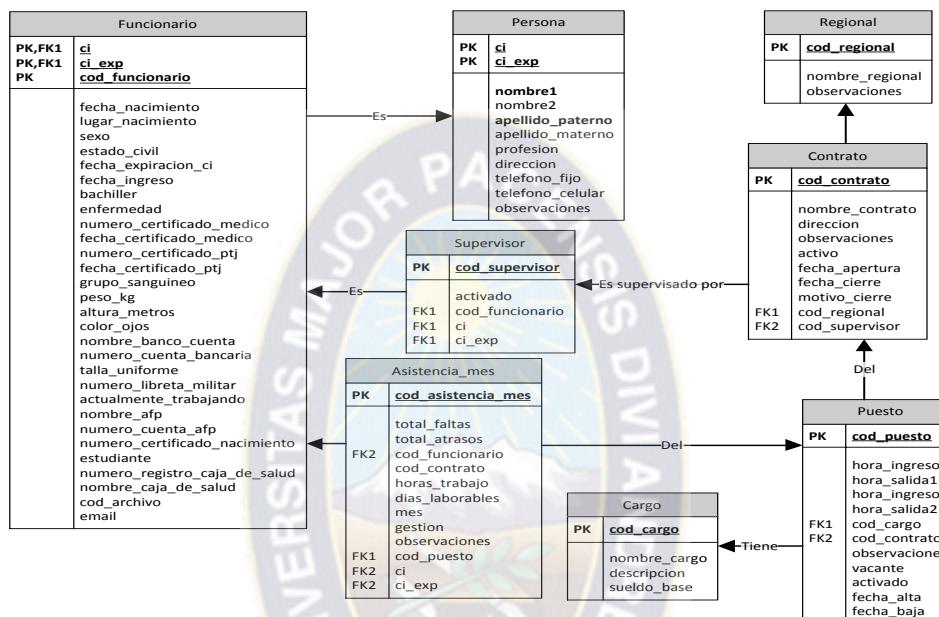


Figura 3.25: Diseño de la tabla para el control de las asistencias de un funcionario

Tarea 5.2: Modulo de verificación de asistencia del funcionario por contratos y general.

Firefox → Sistema de Control y Administración de...

localhost:8080/SCARH-TOTES/cirth/scarh.ds

SISTEMA DE CONTROL Y ADMINISTRACIÓN DE RECURSOS HUMANOS - TOTE's.

MENÚ SUPERVISOR

- CONTRATOS
 - Buscar Contrato
 - Listar Contratos
- REGISTRAR ASISTENCIAS
- SESION

GESTIONAR ASISTENCIAS DEL CONTRATO: CON-ENTELDAT

Contrato: ENTEL - DATACOM

Mostrar Asistencias

Actualizar

LISTADO DE LAS ASISTENCIAS DEL CONTRATO: ENTEL - DATACOM
DE ENERO-2014 A MARZO-2014

ENERO-2014							
#	NOMBRE DEL FUNCIONARIO	HORAS DE TRABAJO	NUMERO DE DIAS LABORABLES	TOTAL DIAS TRABAJADOS	TOTAL ATRASOS	TOTAL FALTAS	OBSERVACIONES
1	ARUQUIPA CONDORI NELLY	6	5	3	0	0	EVENTUAL: Ingresó el: 2014-01-04, Se retiró el: 2014-01-09

FEBRERO-2014							
#	NOMBRE DEL FUNCIONARIO	HORAS DE TRABAJO	NUMERO DE DIAS LABORABLES	TOTAL DIAS TRABAJADOS	TOTAL ATRASOS	TOTAL FALTAS	OBSERVACIONES
1	ARUQUIPA CONDORI ANA MARIA	6	17	15	2	2	NUEVO: Ingresó el: 2014-02-12
2	AJURURO MAMANI GUILLERMO E.	4	22	20	1	2	NUEVO: Ingresó el: 2014-02-07
3	ANTEQUERA A. YURI VICTENE	4	20	17	2	3	NUEVO: Ingresó el: 2014-02-09

MARZO-2014							
#	NOMBRE DEL FUNCIONARIO	HORAS DE TRABAJO	NUMERO DE DIAS LABORABLES	TOTAL DIAS TRABAJADOS	TOTAL ATRASOS	TOTAL FALTAS	OBSERVACIONES
1	ARUQUIPA CONDORI ANA MARIA	6	8	7	2	1	RETIRADO: Se retiró el 2014-03-09
2	AJURURO MAMANI GUILLERMO E.	4	18	17	2	1	RETIRADO: Se retiró el 2014-03-19
3	ARIAS TICONA JESUS	4	9	6	2	1	NUEVO: Ingresó el: 2014-03-23
4	ANTEQUERA A. YURI VICTENE	4	31	30	1	1	Funcionario regular
5	MONROY CASTILLO JHONNY RUBEN	6	22	22	17	0	EVENTUAL: Ingresó el: 2014-03-05, Se retiró el: 2014-03-27

[Ver Registros anteriores](#)

LISTADO DE LAS ASISTENCIAS DEL CONTRATO: ENTEL - DATACOM

Copyright © 2013 - 2014 | TOTE'S Expertos en Limpieza, todos los derechos reservados | Developer: Jhony Monroy

Figura 3.26: Pantalla delo listado de asistencias registradas de los funcionarios de un contrato

Pruebas de aceptación para la Historia de usuario 5

- a. Identificar todos los posibles resultados de la historia:**
 - Interfaz para el listado de asistencias de los funcionarios de un contrato.
- b. Identificar los resultados que terminan la historia y los que permiten continuar dentro la historia:**
 - La historia termina cuando se lista las asistencias de los últimos tres meses.
 - La historia permite continuar si es que el usuario desea ver registros de asistencias anteriores.
- c. Identificar los caminos de ejecución posibles:**
 - Comienza la historia cuando el supervisor decide listar el estado del registro de las asistencias de un determinado contrato que este supervisando.
 - Si el Supervisor decide ver registros de meses anteriores, se muestra el listado correspondiente.
 - La historia termina cuando decide continuar con otro modulo.
- d. Asignar un conjunto de valores válidos y valores de entorno a cada camino de ejecución para obtener el resultado esperado:**
 - El conjunto de valores válidos está dado por la información ya almacenada de las asistencias de los funcionarios y los contratos abiertos.
- e. Eliminación de caminos redundantes:**
 - No se encontraron caminos redundantes.

Historia de usuario 6: Registrar asistencias de los funcionarios.

Tarea 6.1: Diseño del formulario para el registro de las asistencias.

REGISTRAR ASISTENCIAS DEL MES DE MARZO 2014 PARA EL CONTRATO: ENTEL - DATA COM									
FUNCIONARIOS QUE TRABAJARON EN EL MES DE MARZO-2014									
#	NOMBRE DEL FUNCIONARIO	HORAS DE TRABAJO	NUMERO DE DIAS LABORABLES*	NUMERO DE ATRASOS*	NUMERO DE FALTAS*	TOTAL DIAS TRABAJADOS	TOTAL ATRASOS	TOTAL FALTAS	OBSERVACIONES
1	ANTEQUERA A. YURI VICENTE	4 horas	31						Funcionario Regular
2	ARIUQUPA CONDORI ANA MARIA	6 horas	8						RETIRADO Se retiró el 2014-03-09
3	AJORURO MAMANI GUILLERMO E.	4 horas	18						RETIRADO Se retiró el 2014-03-19
4	ARIAS TICONA JESUS	4 horas	9						INGRESO Ingresó el: 2014-03-23
5	MONRROY CASILLO JHONNY RUBEN	6 horas	22						EVENTUAL Ingresó el: 2014-03-05 Se retiró el: 2014-03-27

Registrar Asistencias **Cancelar**

REGISTRAR ASISTENCIAS DEL MES DE MARZO-2014 PARA EL CONTRATO: ENTEL - DATA COM

• Se lista a los funcionarios que trabajaron en el contrato en el mes de MARZO de la gestión 2014.
• Completa la información del tipo de faltas y atrasos en el mes para cada funcionario.
• Presiona el botón "Registrar Asistencias" para proceder a registrar las asistencias.
• Presiona el botón "Cancelar" para cancelar esta acción.

Figura 3.27: Diseño del formulario para el registro de asistencias de los funcionarios que trabajaron en un respectivo contrato

Tarea 6.2: Modulo de registro de asistencias.

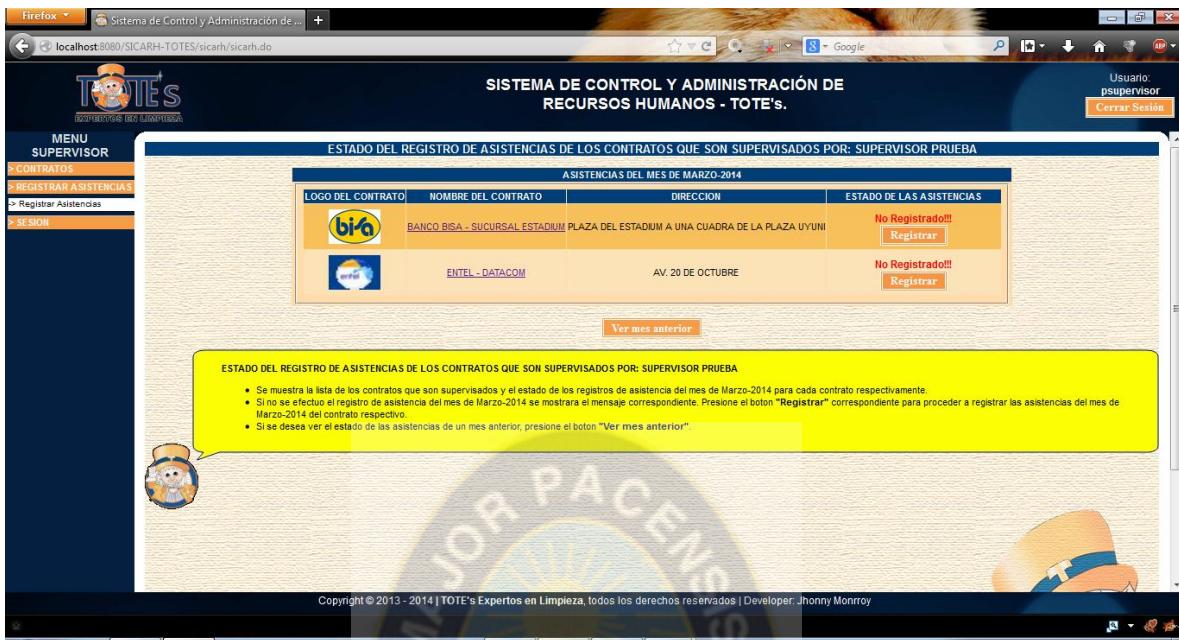


Figura 3.28: Pantalla de inicio mostrando el estado del registro de las asistencias de los contratos supervisados



Figura 3.29: Modulo del llenado de datos de las asistencias para un contrato en particular

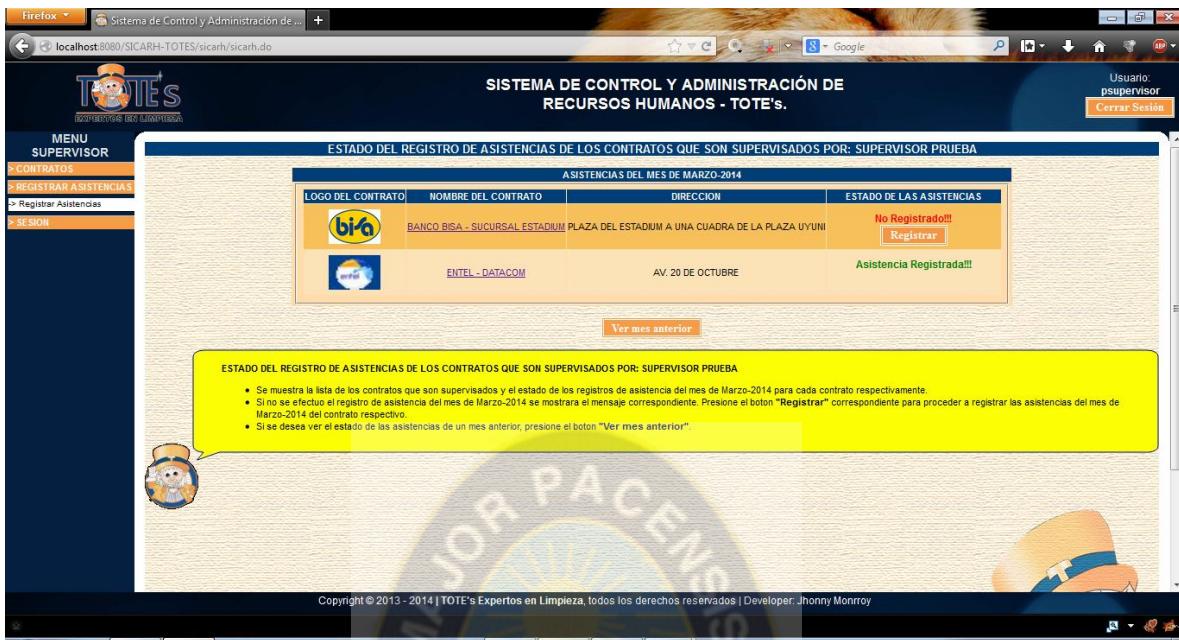


Figura 3.30: Estado del listado de asistencias después del registro de datos

Pruebas de aceptación para la Historia de usuario 6

- Identificar todos los posibles resultados de la historia:**
 - Interfaz para el listado contratos mostrando el estado de registros de las asistencias.
 - Interfaz para el llenado de datos de la asistencia de un contrato.
- Identificar los resultados que terminan la historia y los que permiten continuar dentro la historia:**
 - La historia termina cuando se registra las asistencias de un contrato.
- Identificar los caminos de ejecución posibles:**
 - Comienza la historia cuando el supervisor decide registrar las asistencias de un contrato que este supervisando.
 - La historia termina cuando el supervisor decide continuar con otro modulo.
- Asignar un conjunto de valores válidos y valores de entorno a cada camino de ejecución para obtener el resultado esperado:**
 - El conjunto de valores válidos está dado por la información ya almacenada de los funcionarios, cargos y contratos abiertos.
- Eliminación de caminos redundantes:**
 - No se encontraron caminos redundantes.

3.6.4 Cuarta Iteración

Historia de usuario 7: Generar reportes de planilla de sueldos por contrato.

Tarea 7.1: Diseño del reporte de la planilla por contratos.

Sistema de Control y Administración ... Sistema de Control y Administración ... +

file:///C:/Users/BreakDark/Desktop/Sistema de Control y Administración de Recursos Humanos TOTES_LTDA.htm

Firefox

**PLANILLA PRELIMINAR DEL CONTRATO ENTEL - DATACOM
MARZO-2014**

Total Días Laborables: 31

#	NOMBRE DEL FUNCIONARIO	CARGO	HORAS DE TRABAJO	DÍAS TRABAJADOS	TOTAL ATRASOS	TOTAL FALTAS	SUELDO BÁSICO	SUELDO POR DIA	DESCUENTO POR ATRASO % Dia	DESCUENTO POR FALTA 2 Días	TOTAL GANADO	TOTAL DESCUENTO POR ATRASOS	TOTAL DESCUENTO POR FALTAS	LIQUIDO PAGABLE
1	ARUQUIPA CONDORI ANA MARIA	OPERARIO DE LIMPIEZA	6	7	2	1	900,00 Bs	29,03 Bs	14,52 Bs	58,06 Bs	203,23 Bs	-29,03 Bs	-58,06 Bs	116,13 Bs
2	AJURURO MAMANI GUILLERMO E.	OPERARIO DE LIMPIEZA	4	16	2	2	600,00 Bs	19,35 Bs	9,68 Bs	38,71 Bs	309,68 Bs	-19,35 Bs	-77,42 Bs	212,90 Bs
3	ARIAS ICONA JESUS	OPERARIO DE LIMPIEZA	4	8	3	1	600,00 Bs	19,35 Bs	9,68 Bs	38,71 Bs	154,84 Bs	-29,03 Bs	-38,71 Bs	87,10 Bs
4	ANTEQUERA A. YURI VICENTE	OPERARIO DE LIMPIEZA	4	31	1	0	600,00 Bs	19,35 Bs	9,68 Bs	38,71 Bs	600,00 Bs	-9,68 Bs	-0,00 Bs	590,32 Bs
5	MONROY CASILLO JHONNY RUBEN	OPERARIO DE LIMPIEZA	8	19	8	3	1200,00 Bs	38,71 Bs	19,35 Bs	77,42 Bs	735,48 Bs	-154,84 Bs	-232,26 Bs	348,39 Bs

Confirmar Planilla

Cancelar

**PLANILLA PRELIMINAR DEL CONTRATO ENTEL - DATACOM
MARZO-2014**

- Se muestra una planilla elaborada preliminarmente con los datos de las asistencias de cada funcionario.
- Si deseas confirmar la planilla, presiona el botón "Confirmar Planilla" para proceder a registrarla en la base de datos.
- NOTA: Si confirma el registro, no podra modificarlo mas.
- Si existe algun error puede cancelar el proceso presionando el botón "Cancelar".
- Para modificar esta planilla se tiene que modificar antes el registro de las asistencias que realizo el supervisor de este contrato.

Figura 3.31: Diseño preliminar del reporte de la planilla por contratos

Tarea 7.2: Modulo de generación de reportes de planilla de sueldos.

Sistema de Control y Administración ... Sistema de Control y Administración ... +

localhost:8080/SICARH-TOTES/sicarh/sicarh.do

SISTEMA DE CONTROL Y ADMINISTRACIÓN DE RECURSOS HUMANOS - TOTES.

Usuario: jecursos Cerrar Sesión

GENERAR PLANILLA PARA LOS CONTRATOS DE LA REGIONAL: LA PAZ

Marzo-2014

NOMBRE DEL CONTRATO	ESTADO	OBSERVACIONES	VER PLANILLA
COMIDOL ARCHIVOS	CONTRATO CERRADO Se cerro el contrato en fecha: 2014-03-28		Ver Planilla
BANCO BISA - CENTRAL LA PAZ	CONTRATO ABIERTO		Ver Planilla
BANCO BISA - SUCURSAL ESTADIUM	CONTRATO ABIERTO		Ver Planilla
ENTEL - DATACOM	CONTRATO ABIERTO		Ver Planilla
TOTES - LA PAZ	CONTRATO ABIERTO		Ver Planilla

Ver Registros anteriores

LISTADO DE LAS ASISTENCIAS DEL CONTRATO:
DE - A -

- Se muestra un listado del estado de los registros de asistencias de los meses: - - -
- Si no se registra la asistencia de un mes, presione el botón "Registrar" correspondiente para proceder a efectuar el registro.
- Para ver un registro de asistencia anterior a - presione el botón "Ver Registros anteriores"

Copyright © 2013 - 2014 | TOTES Expertos en Limpieza, todos los derechos reservados | Developer: Jhony Monroy

Figura 3.32: Modulo para generar planillas por contratos

The screenshot shows a Firefox browser window displaying the 'SISTEMA DE CONTROL Y ADMINISTRACIÓN DE RECURSOS HUMANOS - TOTES' application. The URL is 'localhost:8080/SICARH-TOTES/sicarh/sicarh.do'. The page title is 'SISTEMA DE CONTROL Y ADMINISTRACIÓN DE RECURSOS HUMANOS - TOTES.' A user menu at the top right shows 'Usuario: jecurus' and 'Cerrar Sesión'. On the left, a sidebar menu includes 'MENU JEFE RECURSOS HUMANOS', 'FUNCIONARIOS', 'CONTRATOS', 'CARGOS', 'PUESTOS', 'PLANILLAS' (with options 'Listar Planillas' and 'Generar Planilla'), and 'SESION'. The main content area displays a table titled 'PLANILLA PRELIMINAR DEL CONTRATO ENTEL - DATACOM MARZO 2014'. The table has 13 columns: '#', 'NOMBRE DEL FUNCIONARIO', 'CARGO', 'HORAS DE TRABAJO', 'DIAS TRABAJADOS', 'TOTAL ATRASOS', 'TOTAL FALTAS', 'SUELDO BASICO', 'SUELDO POR DIA', 'DESCUENTO POR ATRASO % Dia', 'DESCUENTO POR FALTA 2 Dias', 'TOTAL GANADO', 'TOTAL DESCUENTO POR ATRASOS', and 'TOTAL DESCUENTO POR FALTAS'. Below the table is a yellow confirmation dialog box with the title 'PLANILLA PRELIMINAR DEL CONTRATO ENTEL - DATACOM MARZO-2014'. It contains the following text: '• Se muestra una planilla elaborada preliminarmente con los datos de las asistencias de cada funcionario. • Si desea confirmar la planilla, presione el botón "Confirmar Planilla" para proceder a registrarla en la base de datos. NOTA.- Si confirma el registro, no podrá modificarlo mas. • Si existe algún error puede cancelar el proceso presionando el botón "Cancelar".' Buttons for 'Confirmar Planilla' and 'Cancelar' are shown.

#	NOMBRE DEL FUNCIONARIO	CARGO	HORAS DE TRABAJO	DIAS TRABAJADOS	TOTAL ATRASOS	TOTAL FALTAS	SUELDO BASICO	SUELDO POR DIA	DESCUENTO POR ATRASO % Dia	DESCUENTO POR FALTA 2 Dias	TOTAL GANADO	TOTAL DESCUENTO POR ATRASOS	TOTAL DESCUENTO POR FALTAS	LIQUIDO PAGABLE
1	ARUQUIPA CONDORI ANA MARIA	OPERARIO DE LIMPIEZA	6	7	2	1	900,00 Bs	29,03 Bs	14,52 Bs	58,06 Bs	203,23 Bs	-29,03 Bs	-58,06 Bs	116,13 Bs
2	AJORUJO MAMANI GUILLERMO E.	OPERARIO DE LIMPIEZA	4	16	2	2	600,00 Bs	19,35 Bs	9,68 Bs	38,71 Bs	309,68 Bs	-19,35 Bs	-77,42 Bs	212,90 Bs
3	ARIAS TICONA JESUS	OPERARIO DE LIMPIEZA	4	8	3	1	600,00 Bs	19,35 Bs	9,68 Bs	38,71 Bs	154,84 Bs	-29,03 Bs	-38,71 Bs	87,10 Bs
4	ANTEQUEURA A. YURI OPERARIO DE VICENTE	OPERARIO DE LIMPIEZA	4	31	1	0	600,00 Bs	19,35 Bs	9,68 Bs	38,71 Bs	600,00 Bs	-9,68 Bs	-0,00 Bs	590,32 Bs
5	MONROY CASILLO JHONNY RUBEN	OPERARIO DE LIMPIEZA	8	19	8	3	1200,00 Bs	38,71 Bs	19,35 Bs	77,42 Bs	738,48 Bs	-154,84 Bs	-232,26 Bs	348,39 Bs

Figura 3.33: Reporte de la planilla del mes

Pruebas de aceptación para la Historia de usuario 7

a. Identificar todos los posibles resultados de la historia:

- Interfaz para el listado contratos mostrando el estado de registros de las planillas.
- Reporte de la planilla de un contrato de un respectivo mes y gestión.

b. Identificar los resultados que terminan la historia y los que permiten continuar dentro la historia:

- La historia termina cuando se generar una planilla de un contrato.

c. Identificar los caminos de ejecución posibles:

- Comienza la historia cuando el jefe de recursos humanos decide generar la planilla de un contrato que pertenezca a su regional.
- La historia termina cuando el jefe de recursos humanos decide continuar con otro modulo.

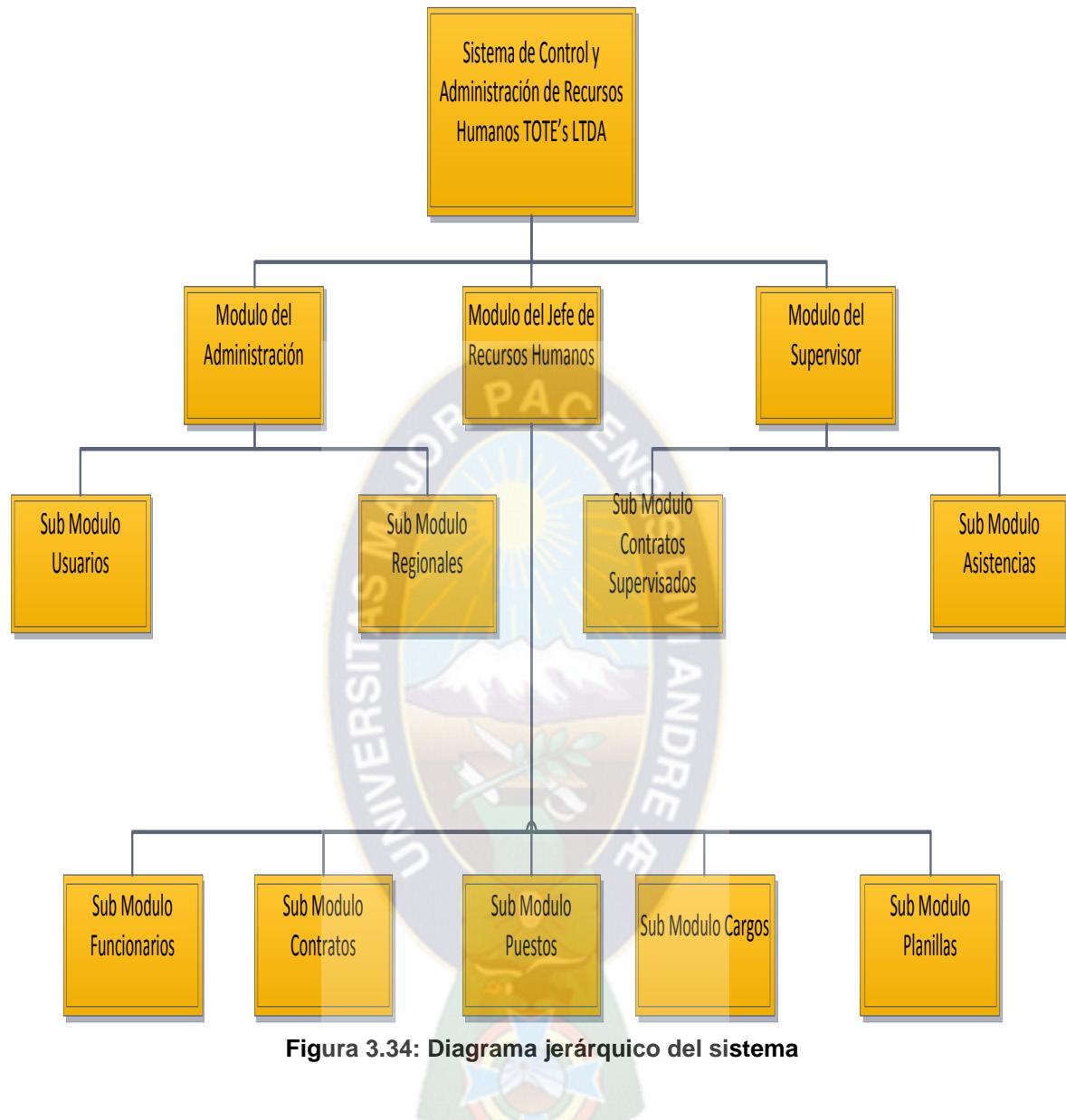
d. Asignar un conjunto de valores válidos y valores de entorno a cada camino de ejecución para obtener el resultado esperado:

- El conjunto de valores válidos está dado por la información ya almacenada de las asistencias de los funcionarios a sus respectivos contratos.

e. Eliminación de caminos redundantes:

No se encontraron caminos redundantes.

3.7 Diagrama Jerárquico del sistema



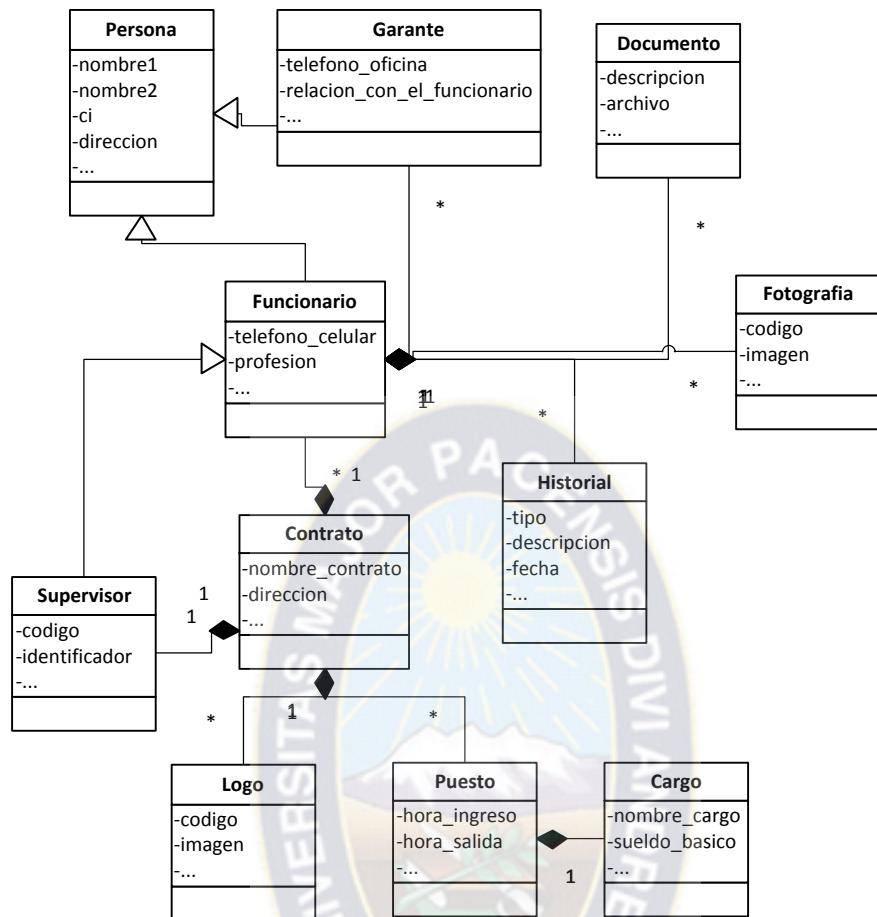


Figura 3.35: Diagrama de clases del modelo

Tabla 3.14 Acceso de los usuarios a los diferentes módulos del sistema

TIPO DE USUARIO	MODULO AL QUE TIENE ACCESO	SUB MODULO AL QUE PUEDE ACCESAR
Administrador	Usuarios	Buscar usuario Listar usuarios Registrar usuarios Mostrar información de los usuarios Editar usuario
	Regionales	Listar regionales Registrar regional Mostrar información de la regional Editar regional
	Sesión	Cambiar su propia contraseña
Jefe de Recursos Humanos	Funcionarios	Buscar funcionario Listar funcionarios Registrar funcionario Mostrar información de un

		funcionario Editar datos personales Editar documentos personales Editar fotografía Editar regional Adicionar cargo Cambiar de cargo Eliminar cargo Dar de baja de la empresa Administrar garantes del funcionario Administrar documentos digitales del funcionario
	Contratos	Buscar contrato Listar contratos Registrar contrato Mostrar información de un contrato Actualizar supervisor Editar datos del contrato Editar logo de la empresa a la cual pertenece el contrato Cerrar contrato Administrar puestos del contrato
	Cargos	Listar cargos Registrar cargos Mostrar datos del contrato Editar datos del contrato
	Puestos	Listar puestos vacantes
	Planillas	Listar planillas Generar planillas Mostrar planillas generadas
	Sesión	Cambiar su propia contraseña
Supervisor	Contratos	Buscar contrato Listar contratos supervisados Mostrar información del contrato Administrar asistencias del contrato
	Asistencias	Registrar asistencias
	Sesión	Cambiar su propia contraseña

3.8 Modelado del sistema

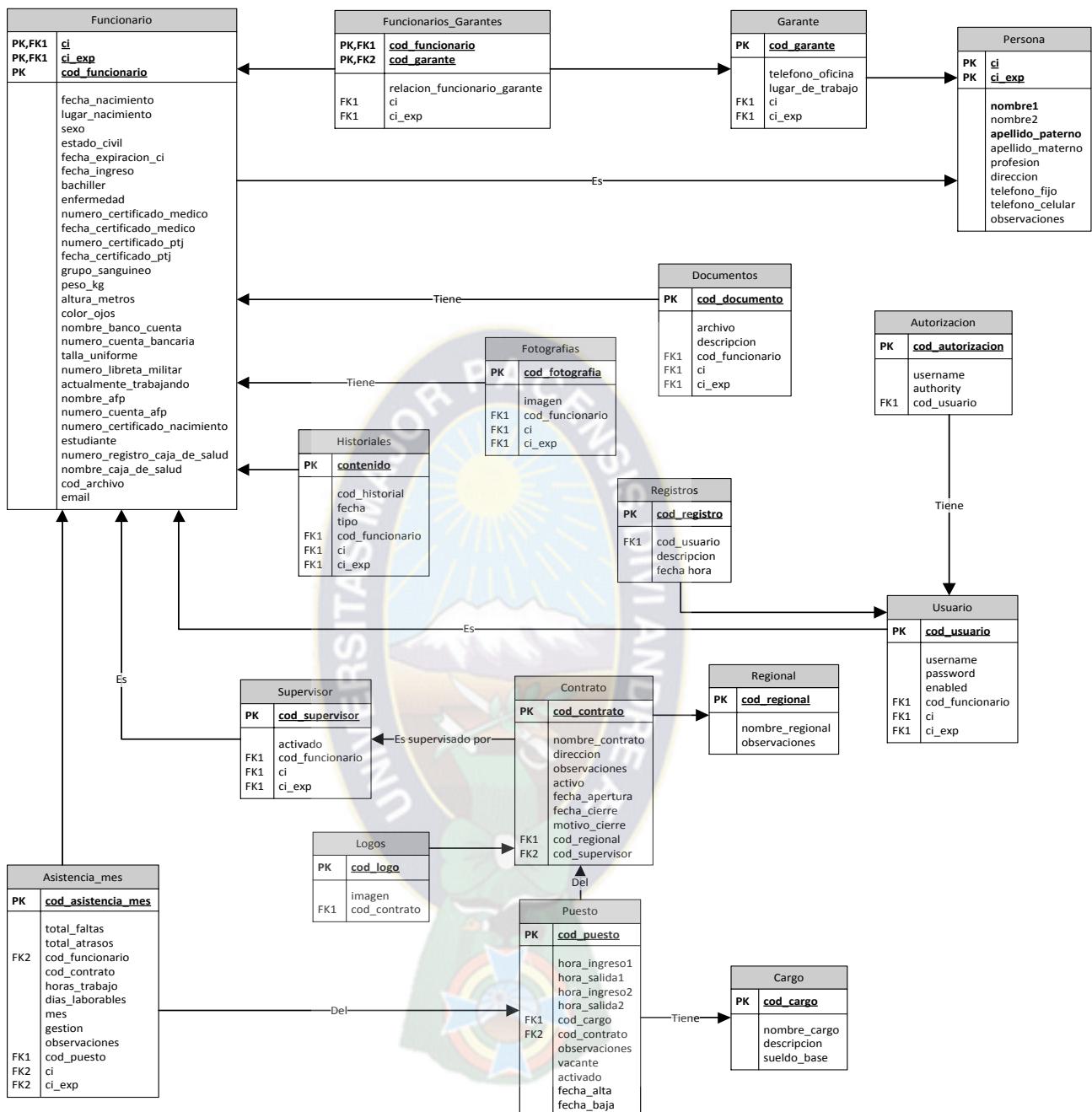


Figura 3.36 Modelo estructural⁴

⁴ Ver Marco teórico, punto 2.7.1 Modelo estructural

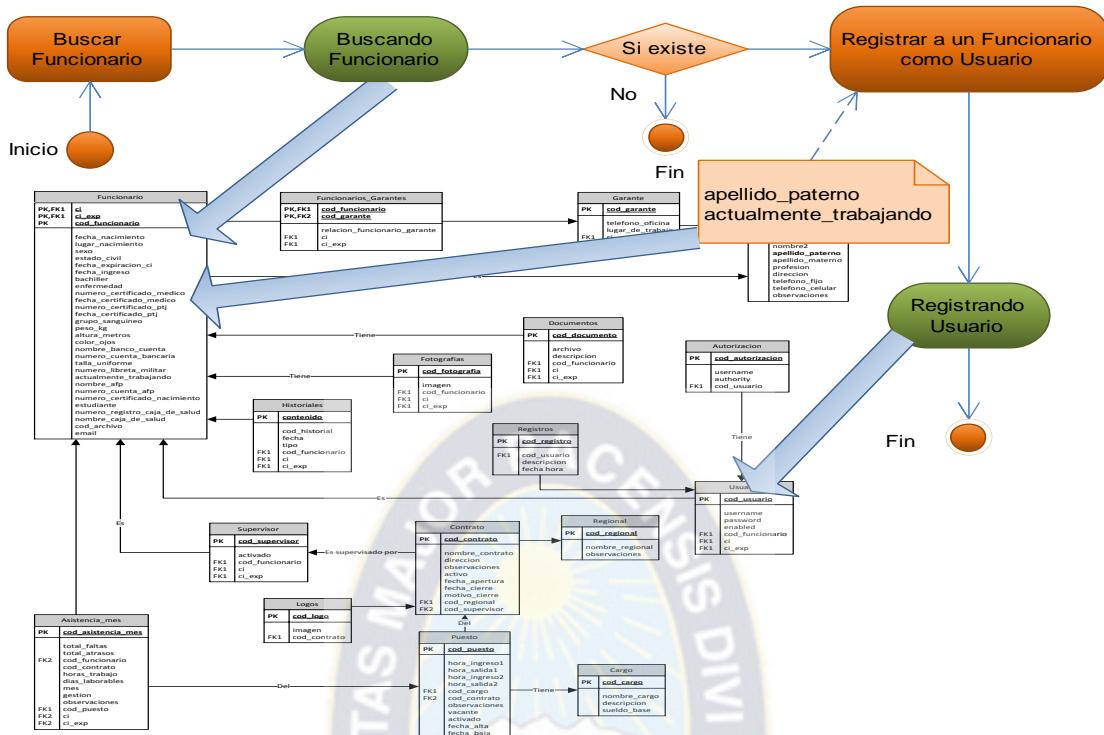


Figura 3.37 Modelo de derivación⁵ (Registrar Usuario)

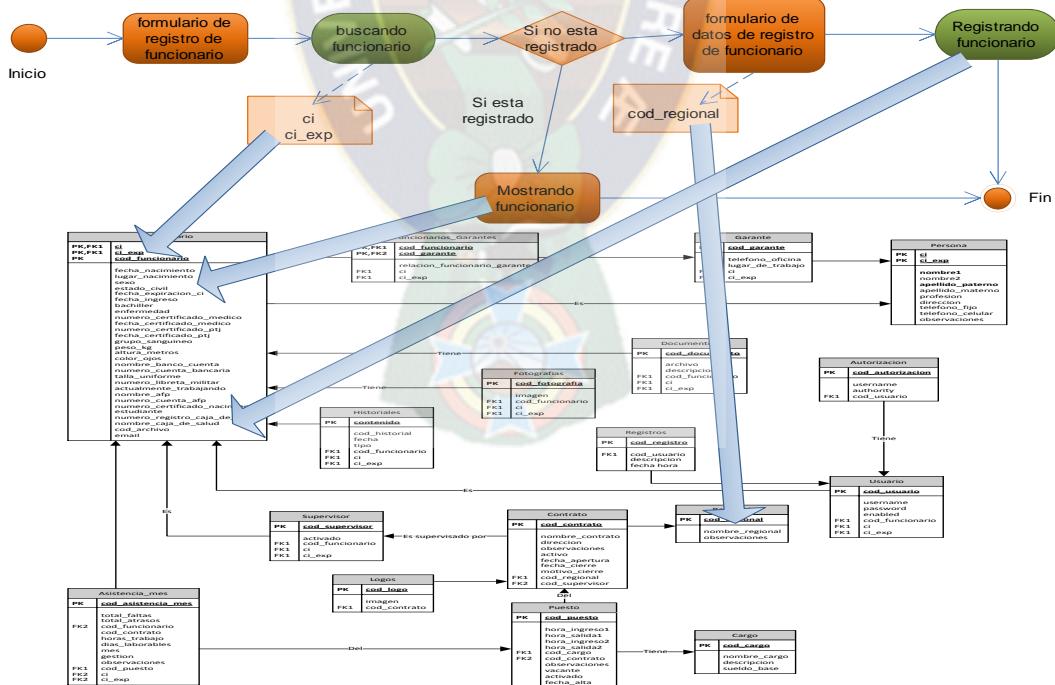


Figura 3.38 Modelo de derivación (Registro de un funcionario)

⁵ Ver Marco teórico, punto 2.7.2 Modelo de Derivación

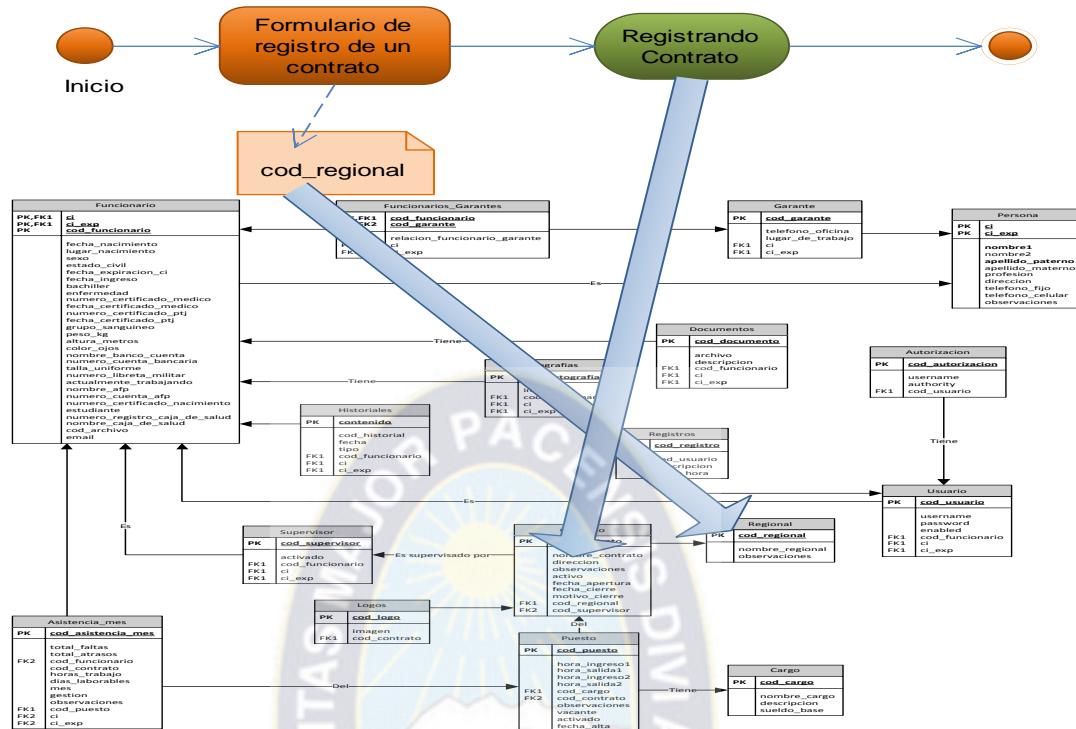


Figura 3.39 Modelo de derivación (registrar contrato)

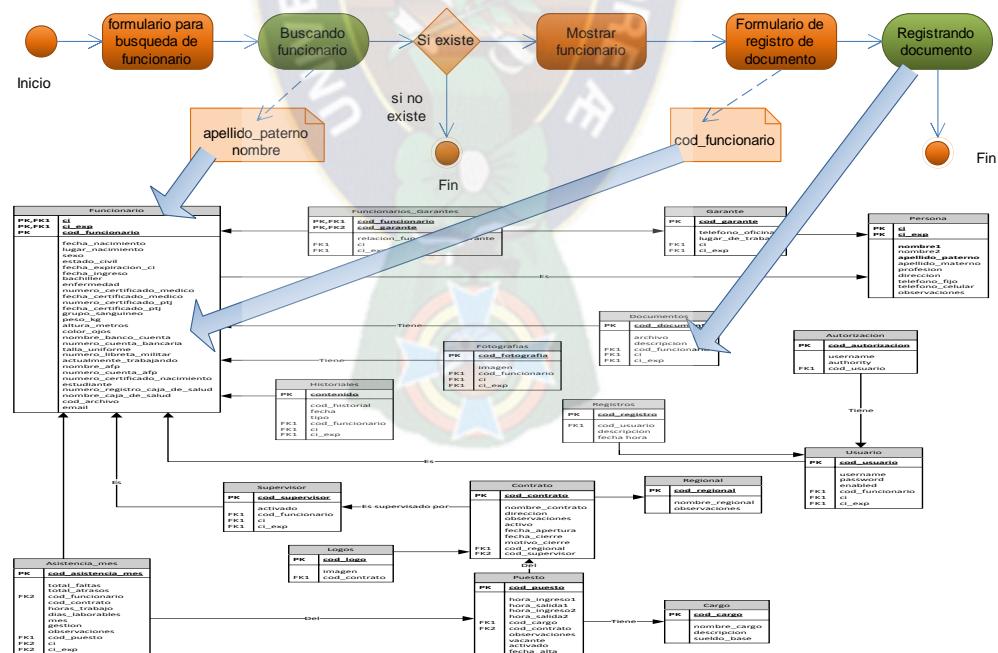


Figura 3.40 Modelo de derivación (Registro de documentación de un funcionario)

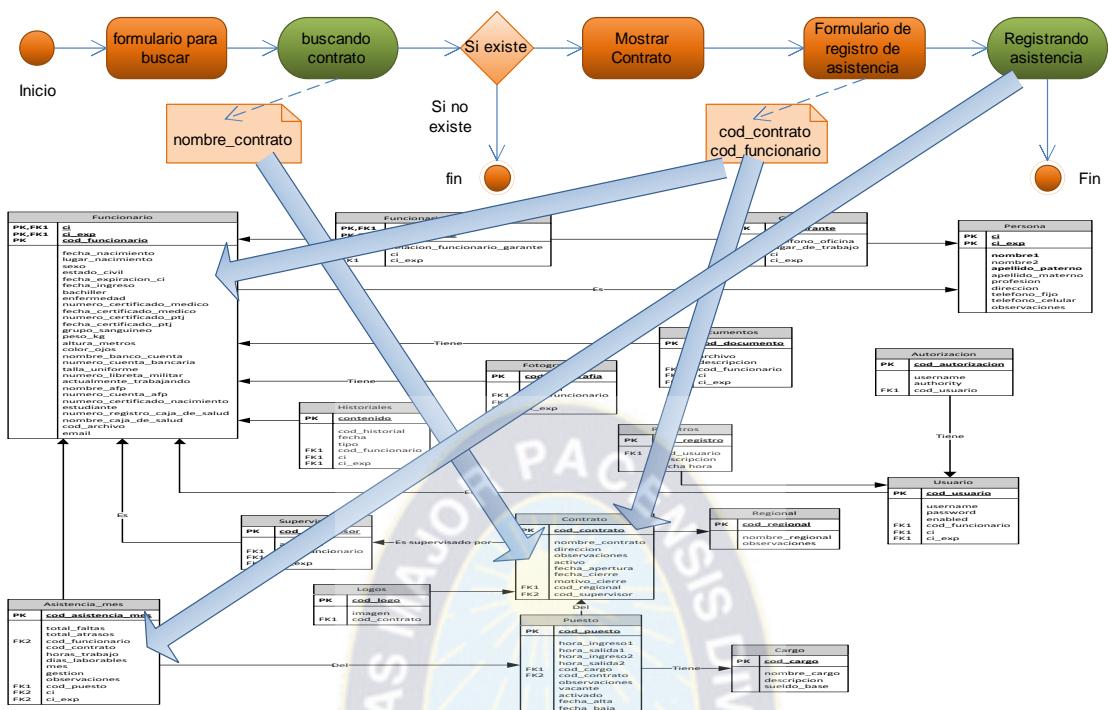


Figura 3.41 Modelo de derivación (Registro de asistencias)

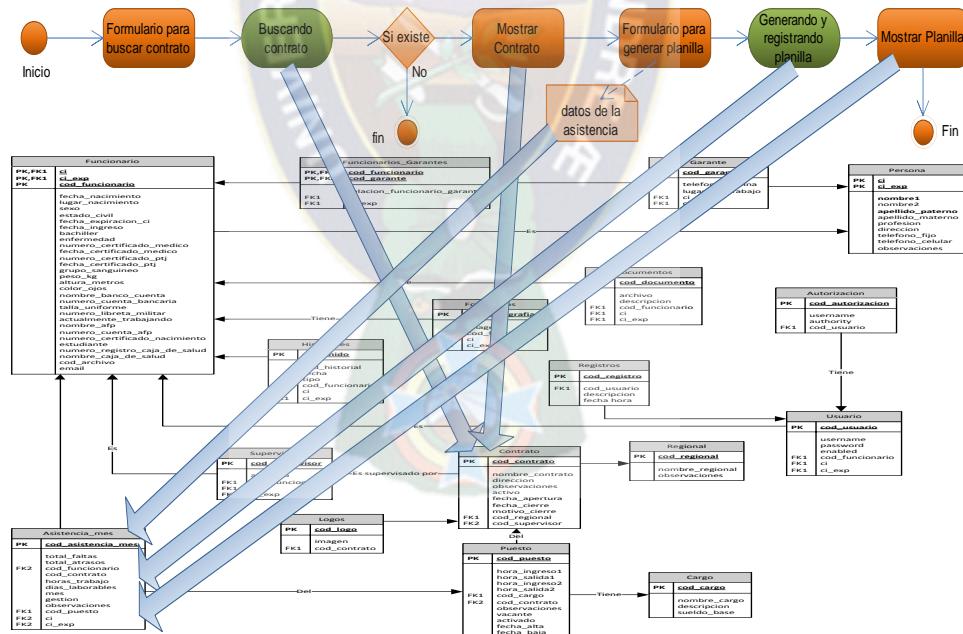


Figura 3.42 Modelo de derivación (Generación de planillas)

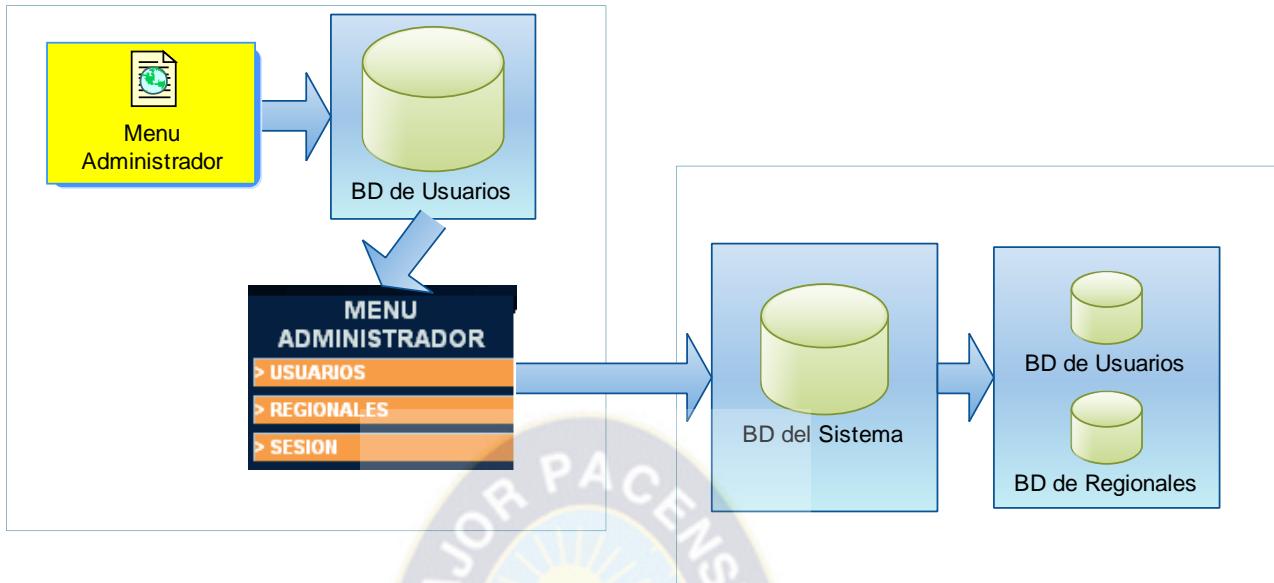


Figura 3.43 Modelo de hipertexto⁶ (Administrador)

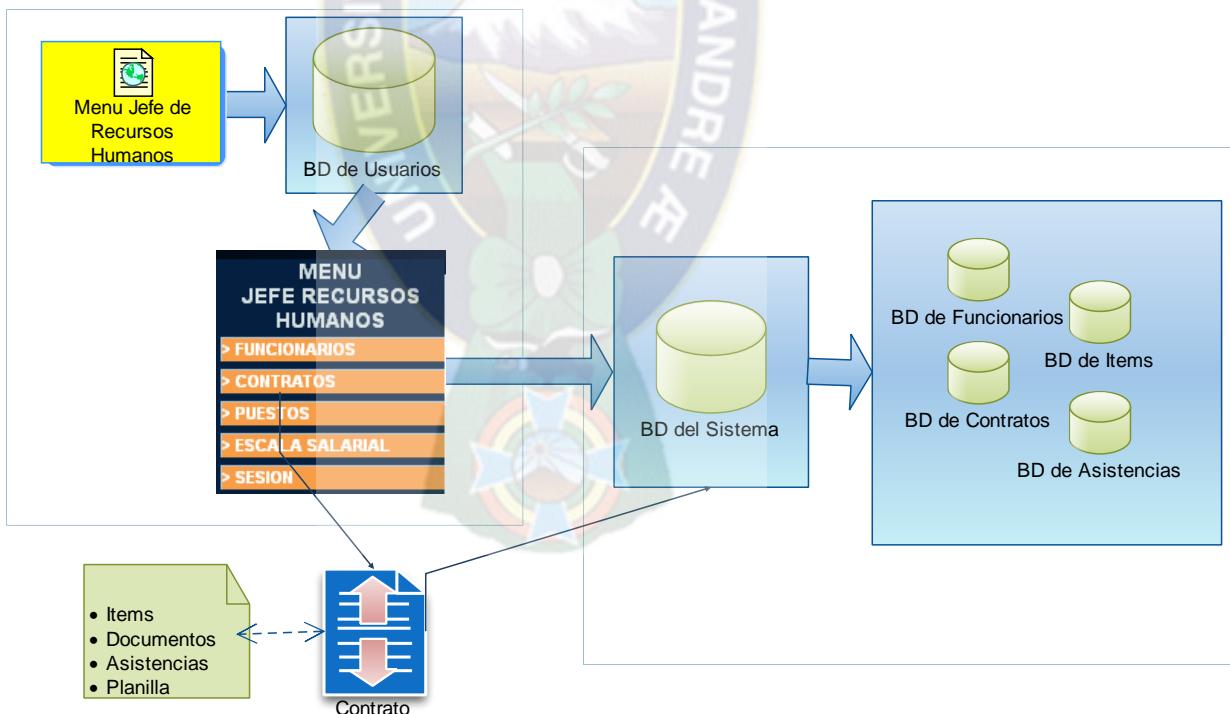


Figura 3.44 Modelo de hipertexto (Jefe de Recursos Humanos)

⁶ Ver punto 2.7.3 Modelo de Hipertexto (Marco Teórico)

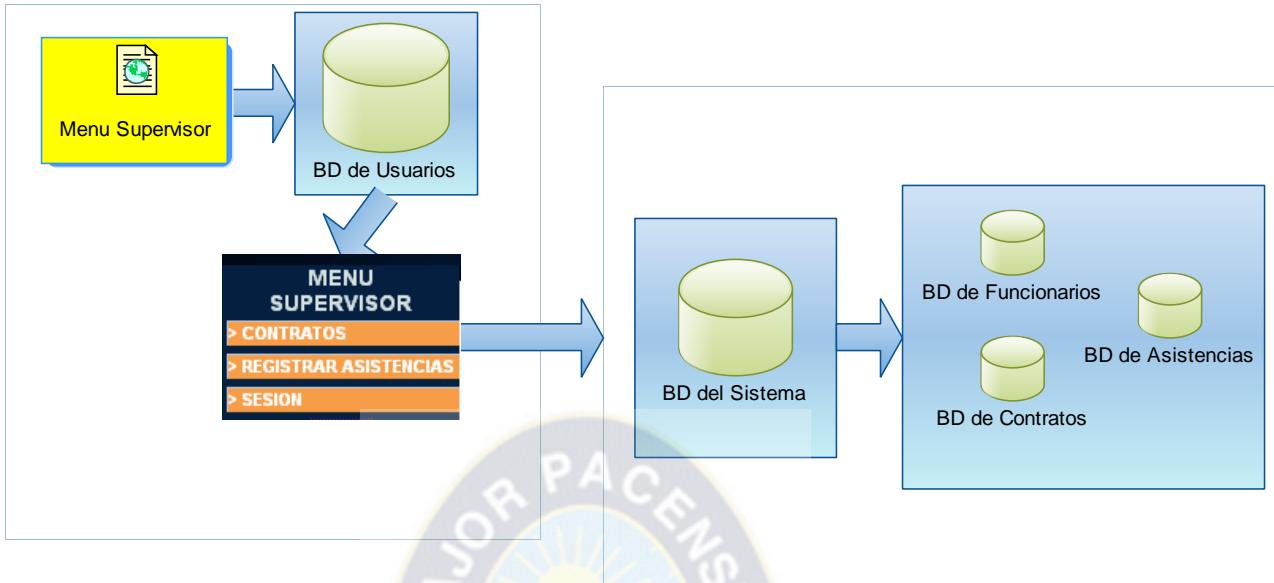


Figura 3.45 Modelo de hipertexto (Supervisor)

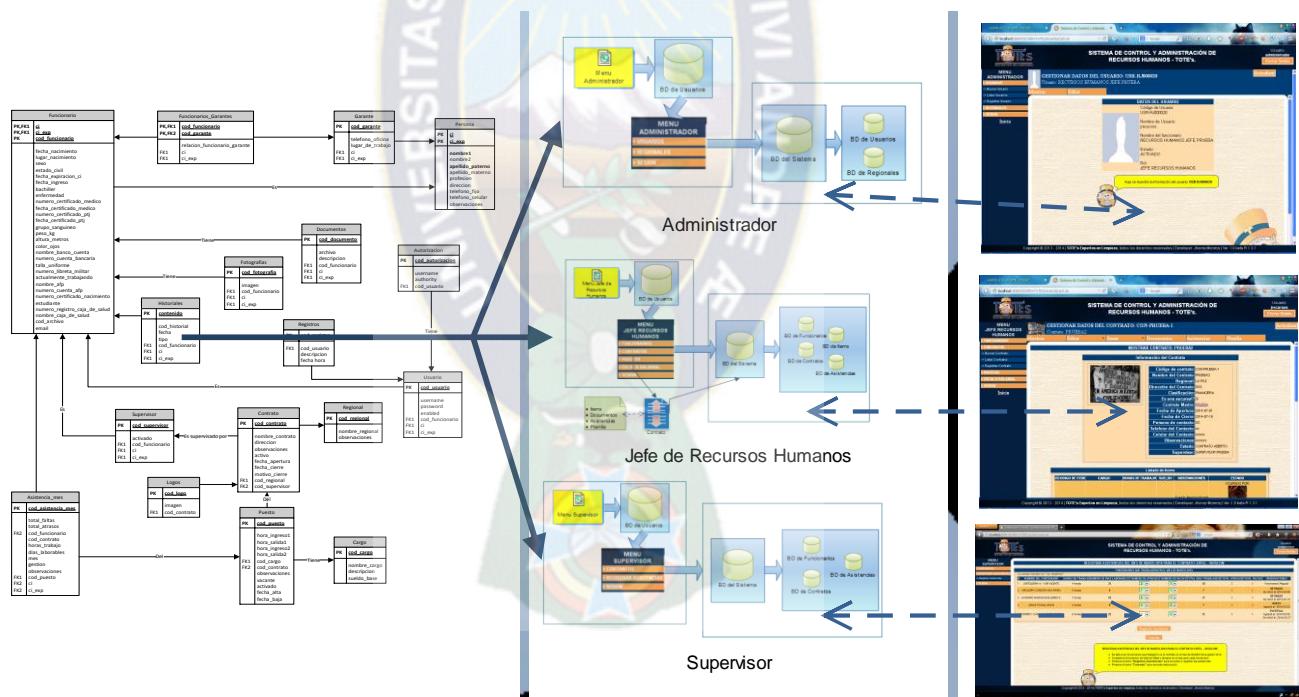


Figura 3.46 Modelo de Presentación⁷

⁷ Ver punto 2.7.3.4 Modelo de Presentación (Marco Teórico)

4 CAPÍTULO IV

MÉTRICAS DE CALIDAD

En este capítulo se tiene como objetivo el de aplicar las métricas de calidad para evaluar el proyecto.

4.1 Introducción

Para evaluar este proyecto se aplican las métricas de calidad según el estándar ISO 9126, evaluándose la fiabilidad, funcionalidad, portabilidad, usabilidad y mantenimiento del sistema.

4.2 Fiabilidad

Durante la entrega de los prototipos realizados en las cuatro iteraciones se encontraron las siguientes deficiencias. A los mismos que necesitaron de un tiempo de servicio determinado. Se logra levantar la siguiente información que se muestra en la siguiente tabla:

Tabla 4.1 Fiabilidad

Tiempo de servicio	Numero de peticiones	Fallas encontradas	Probabilidad de fallo	Tiempo medio entre fallos
8 horas	25	1	0,04	8
16 horas	50	2	0,04	8
32 horas	80	3	0,0375	10,7
64 horas	160	5	0,03125	12,8
TOTAL			0,149	39,5

Por lo tanto el valor promedio de fallas producidas en un tiempo de servicio (PFTS) es de:

$$PFTS = \frac{0.04 + 0.04 + 0.0375 + 0.03125}{4} = \frac{0.149}{4} = 0.03725$$

Lo que indica que el sistema es promedio puede presentar aproximadamente 37 fallas cada mil peticiones.

El tiempo promedio en presentarse estas fallas (TMF) está dado por:

$$TMF = \frac{8 + 8 + 10.7 + 12.8}{4} = \frac{39.5}{4} = 9.875$$

Lo cual indica que el sistema presenta una falla en promedio cada 9.875 horas en el que se hace el uso del sistema.

Entonces se tiene:

$$(1 - PFTS)\% = (1 - 0.03725)\% = (0.9628) = 96.28\%$$

Lo que nos indica que el sistema de control y administración de recursos humanos TOTE's LTDA tiene la capacidad de ser utilizado libre de errores con un 96.28% de seguridad y con una probabilidad de que suceda una falla cada 25 peticiones.

Tabla 4.2 Análisis de la fiabilidad

CARACTERISTICA	VALOR	OBSERVACION	LO QUE SE ESPERA
Fiabilidad	96.28%	Resultado obtenido durante la etapa de las iteraciones	Con el transcurso del tiempo de uso del sistema en las futuras versiones mejoradas se pretende incrementar aún más la fiabilidad.

Es conveniente hacer notar que estos datos fueron levantados en la etapa de las iteraciones y con el transcurso del tiempo el mismo tenderá a no presentar fallas haciendo de la fiabilidad una calidad del software, es así que se vuelve a levantar los datos en la finalización del proyecto según la metodología y se obtienen los siguientes datos:

Tabla 4.3 Fiabilidad a la finalización del proyecto

Tiempo de servicio	Numero de peticiones	Fallos encontrados	Propabilidad de fallo	Tiempo medio entre fallos
8 horas	25	1	0,04	8
16 horas	50	1	0,02	16
32 horas	80	1	0,0125	32
64 horas	160	2	0,0125	32
TOTAL			0,085	88

$$PFTS = \frac{0.04 + 0.02 + 0.0125 + 0.125}{4} = \frac{0.085}{4} = 0,02125$$

$$(1 - PFTS)\% = (1 - 0.02125)\% = (0,97875)\% = 97.87\%$$

Lo que nos indica que el sistema web de control y administración de recursos humanos para la empresa TOTE's LTDA tiene la capacidad de ser utilizado libre de errores con un 97.87% de seguridad.

Tabla 4.4 Fiabilidad a la finalización del proyecto

CARACTERISTICA	VALOR EN LAS ITERACIONES	VALOR FINALIZACION	OBSERVACION
Fiabilidad	96.28%	97.87%	Se puede ver que existe una mejora en la fiabilidad

4.3 Funcionabilidad

Se hace uso de las métricas de adecuación funcional y completitud de la implementación funcional que nos ayuda a medir la implementación funcional y que tan adecuadas son las funciones evaluadas.

4.3.1 Completitud de la implementación funcional

En total el presente proyecto cuenta con 7 historias de usuario. El desarrollo del proyecto se hizo mediante iteraciones, de tal manera que podemos obtener la completitud de la implementación funcional de la siguiente manera:

En la primera iteración se implementó 3 historias de usuario que son: Registro de usuarios, registro de funcionarios, registro de contratos.

$$X = 1 - \frac{A(HU \text{ implementadas})}{B(HU \text{ definidas en la fase de exploracion})} = 1 - \frac{3}{7} = 0.57$$

Lo que nos indica que falta un 57% del sistema por implementar.

En la segunda iteración se implementó 1 historia de usuario: Verificar el estado de la documentación de un funcionario. En total sumadas con las de la primera iteración tenemos hasta aquí 4 historias de usuario.

$$X = 1 - \frac{A(HU \text{ implementadas})}{B(HU \text{ definidas en la fase de exploracion})} = 1 - \frac{4}{7} = 0.43$$

Durante la tercera iteración se implementó otras 2 historias de usuario: Verificar el estado de las asistencias de los funcionarios, registrar asistencias de los funcionarios. Sumadas más las de las iteraciones anteriores tenemos 6 en total.

$$X = 1 - \frac{A(HU \text{ implementadas})}{B(HU \text{ definidas en la fase de exploracion})} = 1 - \frac{6}{7} = 0.14$$

En la cuarta iteración se implementó la restante historia de usuario: Generar reportes de planilla de sueldos por contrato. Con esta historia el resultado final de X es cero. Lo que indica que las historias de usuario descritas en la fase de exploración fueron resueltas al finalizar el proyecto.

4.3.2 Adecuación funcional

Se detectaron problemas en la etapa de evaluación en las historias de usuario: Registro de usuarios y registro de funcionarios que se habilitaba usuarios que ya no estaban trabajando en la empresa y los funcionarios podían garantizarse a sí mismos.

$$X = 1 - \frac{A(HU \text{ donde se detectaron problemas})}{B(HU \text{ evaluadas})} = 1 - \frac{2}{7} = 0.714$$

Lo que nos indica que existe un 71,4% de adecuación de las historias de usuario evaluadas en la fase de producción.

Tabla 4.5 Análisis de la funcionabilidad

CARACTERISTICA	VALOR	OBSERVACION	LO QUE SE ESPERA
Funcionabilidad	71,4%	Resultado obtenido durante la etapa de las iteraciones.	Con el transcurso del tiempo en las futuras versiones se pretende mejorar la adecuación funcional de las historias de usuario.

A la finalización del proyecto se vuelve a medir la adecuación funcional y se obtiene un solo problema de que los funcionarios deberían poder ser garantes de otros funcionarios, el nuevo resultado es:

$$X = 1 - \frac{A(HU \text{ donde se detectaron problemas})}{B(HU \text{ evaluadas})} = 1 - \frac{1}{7} = 0.857$$

Tabla 4.6 Mejora de la funcionabilidad a la finalización del proyecto

CARACTERISTICA	VALOR EN LAS ITERACIONES	VALOR FINALIZACION	OBSERVACION
Funcionabilidad	71,4%	85,7%	Se puede observar que existe una mejora en la funcionabilidad.

Al completar la implementación funcional al finalizar el proyecto se logra llegar a cumplir la implementación de todas las historias de usuario propuestas.

4.4 Usabilidad

Entendido como la operabilidad y la capacidad de ser comprendido, cuyas métricas son la consistencia operacional en el uso y la completitud de la descripción.

Consistencia operacional

Viene dada por la siguiente formula.

$$X = 1 - \frac{A(\# \text{ instancias operacionales comportamiento inconsistente})}{B(\# \text{ total de operaciones})} = 1 - \frac{2}{42} = 0.952$$

Por lo tanto tenemos que un 95,2% del sistema no tiene instancias de operaciones con comportamiento inconsistente.

Consistencia operacional en el uso

Viene dado de la siguiente manera:

$$X = 1 - \frac{A(\# \text{funciones que encontraron inconsistencias})}{B(\# \text{usos por usuario en el periodo de prueba})} = 1 - \frac{2}{15} = 0.867$$

Existe un 86,7% de satisfacción del usuario en la consistencia operacional de uso del sistema.

Tabla 4.7 Análisis de los datos de la usabilidad

USABILIDAD	VALOR	OBSERVACION	LO QUE SE ESPERA
Consistencia operacional	95,2%	Resultado obtenido durante la etapa de las iteraciones.	Con el transcurso del tiempo y uso del sistema en las futuras versiones se pretende que se tienda a incrementar el valor de la consistencia operacional.
Consistencia operacional en el uso	86,7%	Resultado obtenido durante la etapa de las iteraciones.	Se pretende incrementar con el transcurso del tiempo y uso del sistema en las futuras versiones la satisfacción del usuario en la consistencia.

A la finalización del proyecto se procede a medir nuevamente la usabilidad, obteniéndose los siguientes resultados:

Consistencia operacional

$$X = 1 - \frac{A(\# \text{instancias operacionales comportamiento inconsistente})}{B(\# \text{total de operaciones})} = 1 - \frac{1}{42} = 0.976$$

Consistencia operacional en el uso

$$X = 1 - \frac{A(\# \text{funciones que encontraron inconsistencias})}{B(\# \text{usos por usuario en el periodo de prueba})} = 1 - \frac{1}{15} = 0.933$$

Tabla 4.8 Mejora en la usabilidad del sistema

CARACTERISTICA	VALOR EN LAS ITERACIONES	VALOR FINALIZACION	OBSERVACION
Consistencia operacional	95,2%	97,6%	Se puede observar que existe una mejora en la consistencia operacional.
Consistencia operacional en el uso	87%	93,3%	Se puede observar que existe una mejora en la consistencia operacional en el uso.

4.5 Portabilidad

La portabilidad del sistema viene dada de la siguiente manera:

$$X = \frac{A(\# \text{casos exitosos de instalación})}{B(\# \text{de instantes de instalacion})} = \frac{3}{4} = 0,75$$

Lo cual indica que el sistema web tiene una facilidad de instalación de un 75% un valor satisfactorio para el factor de portabilidad.

Tabla 4.9 Análisis de los datos de portabilidad

CARACTERISTICAS	VALOR	OBSERVACION	LO QUE SE ESPERA
Portabilidad	75%	Resultado obtenido durante la etapa de las iteraciones	Este valor se debe a que se realizó la medición en casos donde se instala el sistema web en un servidor local en la empresa.

A la conclusión del proyecto la portabilidad mejora notablemente, y con una nueva evaluación se tiene:

$$X = \frac{A(\# \text{ casos exitosos de instalación})}{B(\# \text{ de instantos de instalacion})} = \frac{4}{4} = 1$$

Tabla 4.10 Análisis de los datos de portabilidad a la finalización del proyecto

CARACTERISTICA	VALOR EN LAS ITERACIONES	VALOR FINALIZACION	OBSERVACION
Portabilidad	75%	100%	Se puede ver que existe una mejora en la portabilidad.

4.6 Mantenimiento

Capacidad para ser analizado: El sistema web tiene la capacidad para ser diagnosticado, para identificar las partes que han de ser modificadas si fuera el caso.

Cambiabilidad: Debido a su arquitectura MVC (Modelo Vista Controlador) y la capacidad del software que permite que una determinada modificación sea implementada, es posible con la el código fuente debidamente documentado siendo posible ampliar el sistema con futuras funciones.

Obteniendo un resumen de todas las mediciones se tiene

Tabla 4.11 Resumen de Métricas de Calidad

Medida	Valor
Fiabilidad	97,87%
Funcionabilidad	85,70%
Usabilidad	93,30%
Portabilidad	100%
PROMEDIO	94,22%

Todos los valores obtenidos están dentro de los márgenes de los estándares de calidad ISO 9126.⁸. Por lo tanto se concluye que la valoración que se obtuvo alcanzo una calidad del 94.22% lo cual representa un grado de satisfacción.

4.7 Análisis de Costos

Los costos que incidieron en la concreción y finalización del Sistema Web se especifican de la siguiente manera:

Costo de desarrollo del software, costo de capacitación del usuario y costo del software base para la construcción del sistema.

Costo de desarrollo del software

⁸ Ver Marco Teórico, punto 2.9.1 ISO 9126

La obtención de este precio se realizó sobre la base del método de estimación empírica COCOMO⁹ básico, que estima el esfuerzo E expresada en personas-mes y el tiempo de duración del desarrollo del software T_d , mediante las siguiente ecuaciones.

$$K_m = E = a_i S_k^{b_i}$$

$$T_d = 2.5 K_m^{0.35}$$

Donde E y K_m son el número de personas necesarias por mes, S_k el tamaño expresado en miles de líneas de código fuente, y a_i , b_i son obtenidos mediante una tabla de coeficientes¹⁰.

Dadas las características del software, se pueden afirmar que el mismo está comprendido entre el grupo de sistemas Semi encajado.

El número estimado de líneas de código se expresa en la siguiente tabla

Tabla 4.12 Conteo estimado del Total de líneas de código durante el desarrollo

Modelos	Vistas	Controladores	TOTAL
1829	1690	1938	5457

En este caso se toma un valor de 5 mil líneas de código estimadas

Dado estos valores se tiene:

$$\begin{aligned} K_m &= E = 3.0 * 5^{1.12} \\ &= 3.0 * 6.07 \\ &= 18.21 \end{aligned}$$

Luego

$$\begin{aligned} T_d &= 2.5 * 18.21^{0.35} \\ &= 2.5 * 2.76 \\ &= 6.9 \end{aligned}$$

De aquí se obtiene una estimación de esfuerzo de 18 personas-mes, en un tiempo de 6 meses, se obtiene el total de personas necesarias.

$$P = \frac{K_m}{T_d} = \frac{18.21}{6.9} = 2.64 \cong 3$$

Considerando que el sueldo de un desarrollador depende de la experiencia del mismo, se da un valor según la oferta de los programadores en el mercado actual boliviano, 2500 Bs al mes. El costo estimado del software es:

$$\begin{aligned} COSTO\ DE\ DESARROLLO &= 3 * 2500 \\ &= 7500\ Bs \end{aligned}$$

⁹ Ver Marco teórico, punto 2.11 El modelo COCOMO

¹⁰ Ver Marco teórico, Tabla 2.2 Coeficientes

Costo de la capacitación de los usuarios

Este costo está dado por producto del tiempo de capacitación, dos horas a 50 Bs (25 Bs la hora), por el número de días de capacitación que en este caso fueron 5, hacen un total de:

$$COSTO DE CAPACITACION = 5 * 50$$

$$= 250 \text{ Bs}$$

Costo del software base para el desarrollo del sistema

Los precios de licencia del lenguaje de programación y el motor de base de datos no se contemplan ya que son libres.

$$COSTO DEL SOFTWARE BASE = 0 \text{ Bs}$$

Haciendo un resumen de costos se obtiene el costo estimado del proyecto

Tabla 4.13 Costo del proyecto

COSTO DE DESARROLLO	7500
COSTO DE CAPACITACION	250
COSTO DEL SOFTWARE BASE	0
TOTAL	7750

Es decir que el costo estimado del proyecto es de 7750 Bs.

5 CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Después de terminar el desarrollo del sistema para el departamento de Recursos Humanos se llegaron a las siguientes conclusiones

- Con la implementación del Sistema Web para el Control y Administración de Recursos Humanos se logró centralizar la información de todos los funcionarios para que esta pueda ser accedida de forma inmediata sin necesidad de estar recurriendo a los archivos centrales de la institución.
- Se logró diseñar la base de datos para la Administración de personal de acuerdo a las necesidades y requerimientos de los usuarios que harán uso del sistema.
- Se implementó el módulo de registro de horarios de entrada y salida, adecuándolos a los horarios de los contratos en donde la Institución presta sus servicios, así como también se implementaron herramientas para la emisión de reportes de asistencia del personal operativo, puestos vacantes y planillas.
- Se implementó triggers en el gestor de base de datos que deshabilitan a un usuario que no esté trabajando en la institución mejorando así la seguridad e integridad de la información almacenada.

El Sistema Web para el control y administración de Recursos Humanos para la Empresa de Limpieza Industrial “TOTES LTDA” ha sido desarrollado bajo las especificaciones planteadas cumpliendo así con el objetivo general.

Cabe hacer notar que los resultados obtenidos en las métricas de calidad fueron tomados en la etapa de las iteraciones siendo así de manera preliminar, y se espera una mejora en los datos obtenidos a partir del conjunto de datos que serán registrados en la base de datos del sistema.

5.2 Recomendaciones

Las recomendaciones que se pueden realizar para futuros proyectos basados en este están referidas a continuación:

Se puede ampliar el sistema de tal forma que se pueda implementar un sistema que utilice los datos especialmente de asistencia para sugerir asensos futuros y dar baja de funcionarios que son informales en sus asistencias.

El sistema web está orientado según las necesidades de la empresa en la cual los horarios de trabajo son diversos, habiendo horarios de 2, 4, 8, incluso 12 horas de trabajo y una complejidad es que los funcionarios no están en solo ambiente de trabajo (contrato), se sugiere que en un futuro se adicue el uso de relojes biométricos inalámbricos y que se

recupere la información de este y se los suba vía web mediante el sistema para que las asistencias sean mejor controladas.

Viendo las necesidades de la empresa también se puede automatizar otras actividades en otras áreas como es la postulación de nuevos empleados por vía web, obtener sus certificados de trabajo automáticamente vía web sin necesidad de estar recurriendo constantemente a la oficina de recursos humanos.

El sistema está orientado a la Web y puede ser ejecutado por dispositivos móviles como ser celulares inteligentes con navegadores, pero se sugiere que en un futuro se realice una aplicación exclusiva para estos dispositivos para mejorar la conectividad y la interfaz que existe con el usuario y el sistema en conjunto.



6 BIBLIOGRAFÍA

- Anaya, A. (s.f.). *A propósito de programación extrema XP (eXtreme Programming)*. Popayán – Cauca - Colombia.
- Ceballos, F., Arboleda, H., & Casallas, R. (30 de Noviembre de 2008). *Un Enfoque para Desarrollar Aplicaciones WEB Basado en Líneas de Producto Dirigidas por Modelos*. Recuperado el 14 de Septiembre de 2014, de Revista electrónica Paradigma en construcción de software: http://paradigma.uniandes.edu.co/index.php?option=com_content&view=article&catid=47%3Aarticulos&id=110%3Aun-enfoque-para-desarrollar-aplicaciones-web-basado-en-lineas-de-producto-dirigidas-por-modelos&lang=es&showall=1
- Chávez, F. (2007). *Sistema de Administración de Recursos Humanos para el CEMSE*. La Paz - Bolivia.
- Delgado, J. L. (2008). *Sistema de Administración de personal Instituto Nacional de Reforma Agraria Dirección Nacional de Recursos Humanos*. La Paz - Bolivia.
- Dolado, J. (s.f.). *El Modelo COCOMO*. Recuperado el 14 de Septiembre de 2014, de Campus de Gipuzkoa - Campus Gipuzkoa - UPV_EHU: <http://www.sc.ehu.es/jiwdocj/mmis/cocomo.htm>
- Fernández, G. (2002). *Introducción a Extreme Programming*.
- Llanque, J. V. (2009). *Sistema biométrico dactilar para el control de personal y planillas de pago*. La Paz - Bolivia.
- Mamani, E. (2011). *Sistema de administración de Recursos Humanos CASO:File Archivos S.A*. La Paz - Bolivia.
- Marconi, J. (2012). *Administración de Personal*. La Ceiba - Honduras.
- Martinez, R. (10 de Octubre de 2010). *Sobre PostgreSQL*. Recuperado el 14 de Septiembre de 2014, de Portal en español sobre PostgreSQL: http://www.postgresql.org.es/sobre_postgresql
- Mendez, A. (Agosto de 2013). Jefe Administrativo TOTES's LTDA. (J. Monroy, Entrevistador)
- Nina, G. O. (s.f.). *Sistema de administración de personal Unidad de talento humano Hospital Agramont*. La Paz - Bolivia.
- Noriega Quintana, D. J. (2007). *Calidad de Software*.

Oracle y Java _ Tecnologías _ Oracle ES. (s.f.). Recuperado el 14 de Septiembre de 2014, de Oracle España _ Hardware and Software, Engineered to Work Together: www.oracle.com/es/technologies/java/overview/index.html#readmore

Rodriguez, J. (2006). *Pruebas Unitarias*.

Sevilla Sánchez, J. (s.f.). *Spring Framework*.

Smartsys Cia Ltda. (1 de Septiembre de 2011). *Norma ISO-9126 para análisis de software*. Recuperado el 14 de Septiembre de 2014, de BEMUS ERP - Su plataforma PYME Norma ISO-9126 para análisis de software: <http://bemuserp.blogspot.com/2011/09/norma-iso-9126-para-analisis-de.html>

Sommerville, I. (2005). *Ingeniería del Software* (Septima ed.). Madrid - España: Pearson Education S. A.

SpringFramework. (s.f.). *Introduction to Spring Framework*. Recuperado el 1 de Octubre de 2014, de Spring Framework: <http://docs.spring.io/spring-framework/docs/current/spring-framework-reference/html/overview.html#overview-modules>

tecnologimerlin. (4 de Septiembre de 2007). *Tecnología es desarrollo*. Recuperado el 14 de Septiembre de 2014, de <http://tecnologimerlin.blogspot.com/>

Tola, E. (2007). *Sistema biométrico de control de asistencia y planillas de pago*. La Paz - Bolivia.

TOTE's. (2011). *TOTE's - Expertos en Limpieza*. Recuperado el 20 de Agosto de 2014, de sitio web de A. TOTE's LTDA: <http://totes.com.bo>

webml.org. (s.f.). Recuperado el 14 de Septiembre de 2014, de <http://www.webml.org>

Wikipedia-IngenieríaWeb. (8 de Agosto de 2014). *Ingeniería web*. Recuperado el 26 de Agosto de 2014, de Wikipedia: http://es.wikipedia.org/wiki/Ingenier%C3%ADA_web

Wikipedia-Pruebas. (19 de Agosto de 2014). *Pruebas de software*. Recuperado el 26 de Agosto de 2014, de Wikipedia, la enciclopedia libre: http://es.wikipedia.org/wiki/Pruebas_de_software

Wikipedia-WebML. (4 de Agosto de 2013). *WebML - Wikipedia, the free encyclopedia*. Recuperado el 26 de Agosto de 2014, de <http://en.wikipedia.org/wiki/WebML>

ANEXOS



Historia 3

Tabla 6.1 Historia de usuario - Registro de contratos

Historia de Usuario	
Número: 3	Usuario: Jefe de Recursos Humanos
Nombre de Historia: Registro de contratos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta (Alta / Media / Baja)
Puntos estimados: 2	Iteración asignada: 1
Descripción: Diseñar el módulo de registro de contratos	
Observaciones: CONFIRMADO por el cliente	

Tareas

Tabla 6.2 Tarea - Diseño de la base de datos para la información de un contrato

Tarea	
Número de tarea: 3.1	Número de historia: 3
Nombre de tarea: Diseño de la base de datos para la información del contrato	
Tipo de tarea: Diseño	Puntos estimados: ½
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monroy Castillo	
Descripción: Se diseña la tabla que corresponde a la información del contrato.	

Tabla 6.3 Tarea - Modulo de registro de información de un contrato

Tarea	
Número de tarea: 3.2	Número de historia: 3
Nombre de tarea: Modulo de registro de información de un contrato	

Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monrroy Casillo	
Descripción: Se desarrolla el modulo para el registro y habilitación de un nuevo contratos.	

Tabla 6.4 Tarea - Modulo de re apertura de un contrato cerrado

Tarea	
Número de tarea: 3.3	Número de historia: 2
Nombre de tarea: Modulo de re apertura de un contrato cerrado	
Tipo de tarea: Desarrollo	Puntos estimados: ½
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monrroy Casillo	
Descripción: Se desarrolla el modulo para re abrir un contrato que ya estaba registrado en el sistema pero que se cerró con anterioridad.	

Historia 4

Tabla 6.5 Historia de usuario - Verificar el estado de la documentación de un funcionario

Historia de Usuario	
Número: 4	Usuario: Jefe de Recursos Humanos
Nombre de Historia: Verificar el estado de la documentación de un funcionario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta (Alta / Media / Baja)
Puntos estimados: 3 ½	Iteración asignada: 2
Descripción: Diseñar el módulo de registro de documentación del funcionario	

Observaciones:

CONFIRMADO por el cliente

Tareas**Tabla 6.6 Tarea - Diseño de la base de datos para la información de la documentación de un funcionario**

Tarea	
Número de tarea: 4.1	Número de historia: 4
Nombre de tarea: Diseño de la base de datos para la información de la documentación de un funcionario	
Tipo de tarea: Diseño	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monrroy Casillo	
Descripción: Se diseña las tabla que corresponden a la información de la documentación del funcionario, como ser copias digitales (copias escaneadas) almacenadas en la base de datos.	

Tabla 6.7 Tarea - Modulo de registro de documentos del funcionario

Tarea	
Número de tarea: 4.2	Número de historia: 4
Nombre de tarea: Modulo de registro de documentos del funcionario	
Tipo de tarea: Desarrollo	Puntos estimados: 1 ½
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monrroy Casillo	
Descripción: Se desarrolla el modulo para el registro y almacenamiento en la base de datos de los documentos digitalizados de un funcionario.	

Tabla 6.8 Tarea - Modulo de descarga de documentos digitalizados

Tarea	
Número de tarea: 4.3	Número de historia: 4
Nombre de tarea: Modulo de descarga de documentos digitalizados	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monrroy Castillo	
Descripción: Se desarrolla el modulo para descargar documentos que ya están almacenados en la base de datos.	

Administración y control de asistencias

El módulo se puede dividir en las siguientes historias de usuario y sus respectivas tareas:

Historia 5

Tabla 6.9 Historia de usuario - Verificar el estado de las asistencias de los funcionarios

Historia de Usuario	
Número: 5	Usuario: Jefe de Recursos Humanos
Nombre de Historia: Verificar el estado de las asistencias de los funcionarios	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta (Alta / Media / Baja)
Puntos estimados: 3	Iteración asignada: 3
Descripción: Diseñar el módulo para reportar asistencias de los funcionarios	
Observaciones: CONFIRMADO por el cliente	

Tareas

Tabla 6.10 Tarea - Diseño de la base de datos para la información de las asistencias de un funcionario

Tarea	
Número de tarea: 5.1	Número de historia: 5
Nombre de tarea: Diseño de la base de datos para la información de la asistencia de un funcionario	
Tipo de tarea: Diseño	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monrroy Casillo	
Descripción: Se diseña las tablas que corresponden a la información de la asistencia del funcionario.	

Tabla 6.11 Tarea - Modulo de verificación de asistencias de los funcionarios por contratos y general

Tarea	
Número de tarea: 5.2	Número de historia: 5
Nombre de tarea: Modulo de verificación de asistencia del funcionario por contratos y general	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monrroy Casillo	
Descripción: Se desarrolla el modulo para mostrar las asistencias de un funcionario por contrato y su asistencia general.	

Historia 6

Tabla 6.12 Historia de usuario - Registrar asistencias de los funcionarios

Historia de Usuario	
Número: 6	Usuario: Supervisor

Nombre de Historia: Registrar asistencias de los funcionarios	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta (Alta / Media / Baja)
Puntos estimados: 3	Iteración asignada: 3
Descripción: Diseñar el módulo para registrar las asistencias de los funcionarios	
Observaciones: CONFIRMADO por el cliente	

Tareas

Tabla 6.13 Tarea - Diseño del formulario para el registro de las asistencias

Tarea	
Número de tarea: 6.1	Número de historia: 6
Nombre de tarea: Diseño del formulario para el registro de las asistencias	
Tipo de tarea: Diseño	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monroy Casillo	
Descripción: Se diseña el formulario para que el supervisor registre la asistencia del funcionario.	

Tabla 6.14 Modulo de verificación de asistencias de los funcionarios por contrato y general

Tarea	
Número de tarea: 6.2	Número de historia: 6
Nombre de tarea: Modulo de registro de asistencias	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monroy Casillo	

Descripción:

Se desarrolla el modulo para registrar la asistencia de un funcionario.

Generación de Reportes

El módulo se puede dividir en las siguientes historias de usuario y sus respectivas tareas:

Historia 7**Tabla 6.15 Historia de usuario - Generar reportes de planilla de sueldos por contrato**

Historia de Usuario	
Número: 7	Usuario: Jefe de Recursos Humanos
Nombre de Historia: Generar reportes de planilla de sueldos por contrato	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta (Alta / Media / Baja)
Puntos estimados: 3	Iteración asignada: 4
Descripción: Diseñar el módulo para generar las planillas de sueldo por contrato	
Observaciones: CONFIRMADO por el cliente	

Tareas**Tabla 6.16 Tarea - Diseño del reporte de la planilla por contratos**

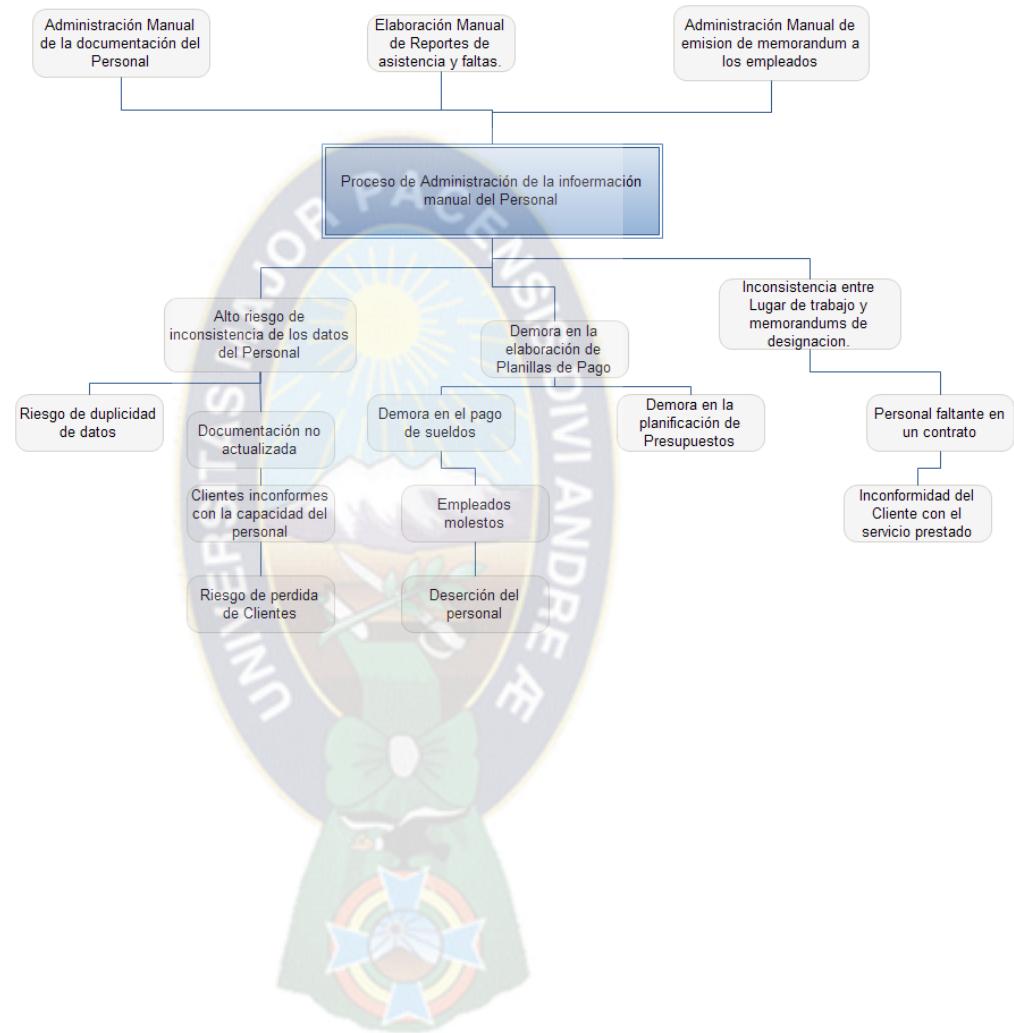
Tarea	
Número de tarea: 7.1	Número de historia: 7
Nombre de tarea: Diseño del reporte de la planilla por contratos	
Tipo de tarea: Diseño	Puntos estimados: 1
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monroy Casillo	
Descripción: Se diseña el formato del reporte que se generara.	

Tabla 6.17 Tarea - Modulo de generación de reportes de planilla de sueldos

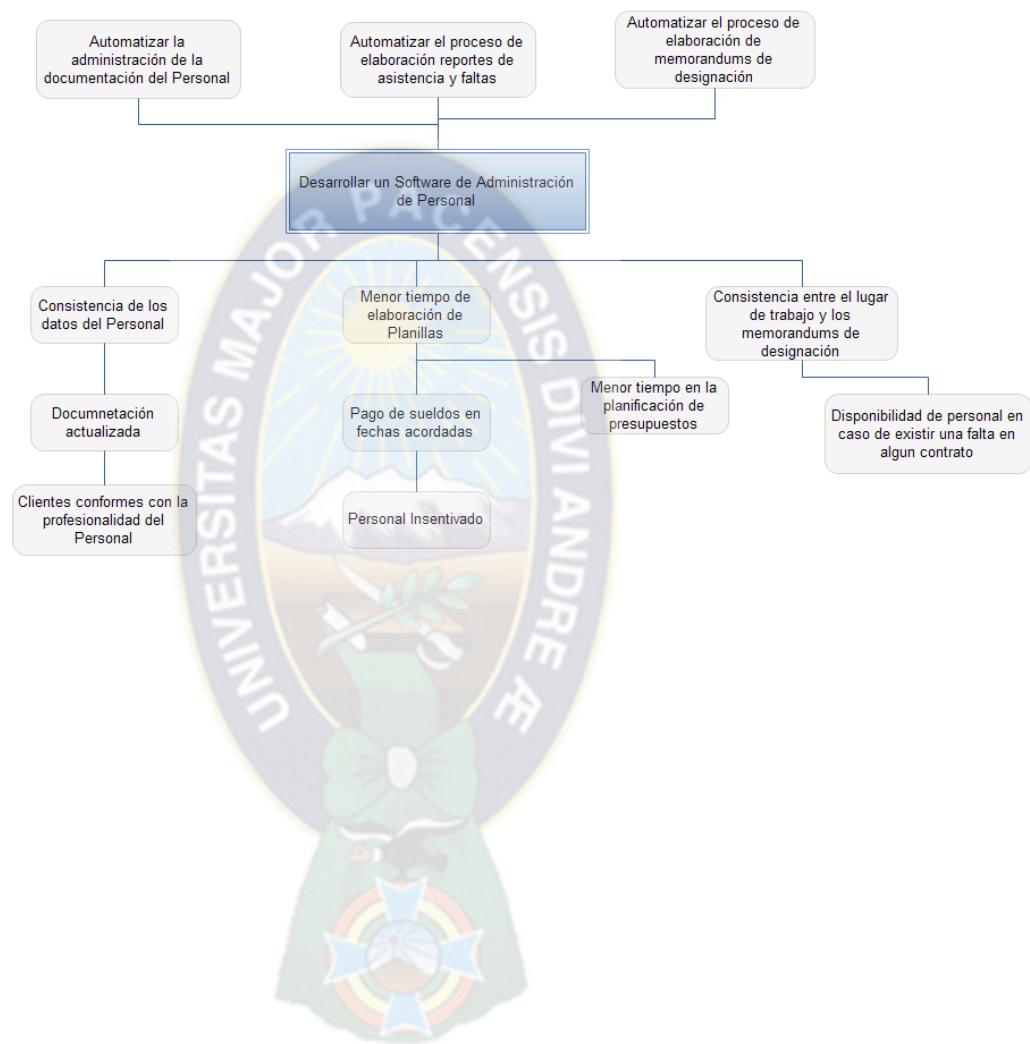
Tarea	
Número de tarea: 7.2	Número de historia: 7
Nombre de tarea: Modulo de generación de reportes de planilla de sueldos	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio:	Fecha fin:
Responsable: Jhonny Rubén Monroy Castillo	
Descripción: Se desarrolla el modulo para generar la planilla preliminar de sueldos por contratos.	



Arbol de Problemas



Arbol de Objetivos



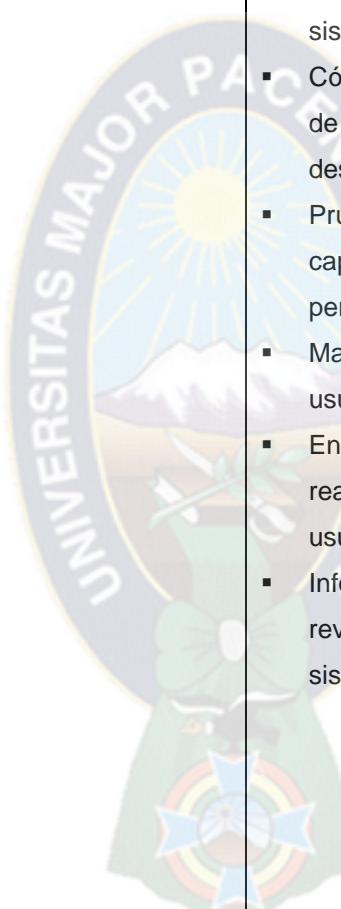
Matriz del Marco Lógico

Título: Sistema Web para el Control y Administración de Recursos Humanos.

Caso: Empresa de Limpieza Industrial “TOTES LTDA”.

Resumen Narrativo de Objetivos	Identificadores verificables objetivamente	Medios de Verificación	Supuestos
FIN: Automatizar el proceso de administración del personal en la Empresa de Limpieza Industrial “TOTES LTDA”	Habrá un mejor manejo de la información del Personal.	<ul style="list-style-type: none"> ▪ Estadísticas anuales del funcionamiento del sistema de información. ▪ Resultados de encuestas a los usuarios. 	Buena administración del Sistema de Información y conocimiento de la importancia que corresponde a la información que será manejada.
PROPOSITO: <ul style="list-style-type: none"> ▪ Desarrollar un Sistema Web para el Control y Administración de Recursos Humanos. 	<ul style="list-style-type: none"> ▪ Información coherente sobre el registro del personal. 	<ul style="list-style-type: none"> ▪ Reportes de la información del Personal con un formato estándar. ▪ Reportes de asistencia del Personal con un formato estándar. 	<ul style="list-style-type: none"> ▪ La veracidad de la información resultante es controlada constantemente por los usuarios. ▪ Personal Dispuesto y capacitado para operar con el Sistema de Información. ▪ Aprobación del Gerente General por los resultados obtenidos.
PRODUCTO: <ul style="list-style-type: none"> ▪ Elaborar herramientas que realicen el control de la 	<ul style="list-style-type: none"> ▪ Entrega del Sistema de Información en funcionamiento. ▪ Un Sistema 	<ul style="list-style-type: none"> ▪ Documento de entrega y conformidad del Sistema de Información. 	<ul style="list-style-type: none"> ▪ Tecnología y Equipos de computación disponibles en el mercado local.

<p>documentación del personal, así como también controle los permisos y faltas del personal operativo y administrativo.</p> <ul style="list-style-type: none"> ▪ Elaborar un módulo que generar reportes, informes e historial de asistencias del personal operativo y administrativo. ▪ Elaborar un módulo que genere reportes e informes sobre el estado de los contratos y el personal existente y faltante en cada una de las instituciones a las que se brinda el servicio de limpieza industrial. 	<p>orientado a la Web capaz de realizar reportes diarios de personal.</p> <ul style="list-style-type: none"> ▪ Incremento en la efectividad del personal. ▪ Personal capacitado para el manejo del sistema de información. 	<ul style="list-style-type: none"> ▪ Encuestas con el personal para conocer el grado de satisfacción de uso del Sistema. ▪ Documentación del Sistema. ▪ Evaluaciones periódicas. 	<ul style="list-style-type: none"> ▪ Personal capacitado y calificado. ▪ Seguridad y resguardo del servidor de la Base de Datos.
<p>PLAN DE ACTIVIDADES:</p> <ul style="list-style-type: none"> ▪ Identificar el problema principal. 	<ul style="list-style-type: none"> ▪ 7 meses para la realización del proyecto. ▪ Gastos de transporte al 	<ul style="list-style-type: none"> ▪ Presentación del perfil del proyecto de grado realizado durante tres meses. 	<ul style="list-style-type: none"> ▪ El Personal es receptivo a la implementación del Sistema de Información en la

<ul style="list-style-type: none"> ▪ Elaborar el árbol de problemas. ▪ Elaborar el árbol de objetivos. ▪ Elaborar la Matriz del marco Lógico. ▪ Elaborar del Perfil de proyecto de grado. ▪ Determinación de Requerimientos (1ra iteración). ▪ Diseñar el módulo de interacción. ▪ Desarrollar el Sistema de Información para la Administración de Personal (1ra iteración). ▪ Codificar el software para el sistema de información (1ra iteración). ▪ Implementar el sistema y utilizar los casos de prueba (1ra iteración). ▪ Determinación de Requerimientos (2da iteración). ▪ Desarrollar el Sistema de Información para 	<p>visitar a la Empresa de Limpieza Industrial “TOTES LTDA”.</p> 	<ul style="list-style-type: none"> ▪ Aval del Asesor aprobando el contenido del perfil de proyecto de grado. ▪ Aval de la Empresa aprobando el desarrollo del sistema. ▪ Códigos fuentes de los módulos desarrollados. ▪ Pruebas para la capacitación al personal. ▪ Manuales de usuario. ▪ Encuestas realizadas a los usuarios. ▪ Informe del revisor del sistema. 	<p>Empresa.</p> <ul style="list-style-type: none"> ▪ La Empresa mantiene interés en el desarrollo del sistema. ▪ La gerencia aprueba el presupuesto del proyecto. ▪ Las terminales y el servidor trabajen adecuadamente.
--	---	--	---

<p>el Control de Personal (2da iteración).</p> <ul style="list-style-type: none"> ▪ Codificar el software del Sistema de Información (2da iteración). ▪ Implementar el sistema de información y utilizar los casos de prueba (2da iteración). ▪ Elaboración de los manuales de Usuario. ▪ Capacitación del personal para la puesta en marcha del sistema de información. ▪ Entrega Final del Sistema de Información a la Empresa. 			
--	--	--	--

DOCUMENTOS

