

INSTITUTO TECNOLÓGICO SUPERIOR “CORDILLERA”

CARRERA DE ANÁLISIS DE SISTEMAS

LA ADMINISTRACIÓN DE PROYECTOS EN EL
CONTROL DE VERSIONES DE PRODUCTOS DE
SOFTWARE. SISTEMA DE RELEASE Y
CONTROL DE PRODUCTOS DE
SOFTWARE PARA EL ÁREA DE
DESARROLLO DE LA COMPAÑÍA
DE SEGUROS DE VIDA
COLVIDA S.A.

Proyecto de Trabajo de Graduación que se
Presenta como Requisito para Optar por el
Título de Tecnólogo Analista de Sistemas

Autor: MORALES Caicedo, Ricardo Xavier

Director de Trabajo de Graduación:
Ing. Jorge Tatayo

Quito Abril 2013



DECLARACIÓN DE AUTENTICIDAD

El abajo firmante, declara que los contenidos y los resultados obtenidos en el presente proyecto, como requerimiento previo para la obtención del Título de Tecnólogo Analista de Sistemas, son absolutamente originales, auténticos y personales y de exclusiva responsabilidad legal y académica del autor.

Ricardo Morales Caicedo
1719148288



APROBACIÓN DEL TUTOR

En mi calidad de Tutor del trabajo sobre el tema: "LA ADMINISTRACIÓN DE PROYECTOS EN EL CONTROL DE VERSIONES DE PRODUCTOS DE SOFTWARE. SISTEMA DE RELEASE Y CONTROL DE PRODUCTOS DE SOFTWARE PARA EL ÁREA DE DESARROLLO DE LA COMPAÑÍA DE SEGUROS DE VIDA COLVIDA S.A.", presentado por el ciudadano: Morales Caicedo Ricardo Xavier, estudiante de la Escuela de Sistemas, considero que dicho informe reúne los requisitos y méritos suficientes para ser sometido a la evaluación por parte del Tribunal de Grado, que el Honorable Consejo de Escuela designe, para su correspondiente estudio y calificación.

Quito, Marzo del 2013

Ing. Jorge Tatayo
TUTOR



APROBACIÓN DEL TRIBUNAL DE GRADO

Los miembros del Tribunal de Grado designado por el Honorable Consejo de la Escuela de Sistemas, aprueban el trabajo de investigación de acuerdo con las disposiciones reglamentarias emitidas por el Centro de Investigaciones Tecnológicas y Proyectos del Instituto Tecnológico Superior Cordillera para proyectos de grado de Tecnólogos Analistas de Sistemas del Sr: Morales Caicedo Ricardo Xavier.

Quito, Marzo 2013

Para constancia firman:

PRESIDENTE

VOCAL 1

VOCAL2



AGRADECIMIENTO

En primer lugar agradezco a Dios, porque siempre ha sido mi guía y fortaleza y gracias a su infinito amor me ha ayudado a cumplir con uno de mis más grandes anhelos.

Agradezco de manera muy especial a mi madre quien ha sido un pilar fundamental en mi vida la cual me ha enseñado a que con esfuerzo se logran grandes cosas.

A mis hermanos pues gracias a su apoyo he logrado cumplir con mis metas y aspiraciones.

A todos aquellos amigos en especial a ustedes: Daniel, Christian, Boris, que a pesar de todo siempre han estado ahí apoyándome y dándome ánimos para seguir adelante.

A todos los ingenieros del instituto quienes han contribuido con sus conocimientos en todo el transcurso de mi carrera.

A la compañía de seguros COLVIDA S.A. por abrirme las puertas para poder realizar el presente trabajo de titulación dentro de sus instalaciones.

De verdad muchas gracias a todos.



DEDICATORIA

Dedico este proyecto a todas las personas quienes de una u otra manera me han apoyado en todo el transcurso de mi carrera y de mi vida.

Y de manera muy especial quiero dedicarlo a Dios quien me ha guiado en cada paso de mi vida, a mi madre Blanca a quien admiro porque a pesar de todas las adversidades ha estado siempre apoyándome.

Ricardo Morales



Índice de Contenidos

CAPÍTULO I	1
El Problema	1
1.1.Planteamiento del Problema	1
1.2.Formulación del Problema	2
1.3.Objetivos	2
1.3.1.General.....	2
1.3.2.Específicos	2
1.4.Justificación e Importancia	2
CAPÍTULO II	4
Marco Teórico.....	4
2.1.Antecedentes.....	4
2.2.Fundamentación Teórica	6
2.3.Marco Legal	7
2.4.Glosario de Términos.....	8
2.5.Caracterización de Variables	8
2.6.Preguntas Directrices.....	8
CAPÍTULO III	9
Marco Metodológico	9
3.1.Tipo de Investigación	9
3.1.1.Investigación Aplicada	9
3.2.Población y Muestra	10
3.3.Técnicas de Recolección de Información	10
3.3.1.La Encuesta	10
3.4.Operationabilidad de las Variables	11
CAPÍTULO IV	12
Análisis e Interpretación de Resultados	12
4.1.Análisis de Encuestas	12
4.2.Conclusión del Análisis de Datos	21
CAPÍTULO V	22
Conclusiones y Recomendaciones.....	22



5.1.Conclusiones	22
5.2.Recomendaciones.....	22
CAPÍTULO VI.....	24
La Propuesta	24
6.1.Introducción.....	24
6.1.1.Propósito	24
6.1.2.Objetivo	24
6.1.3.Alcance.....	24
6.1.4.Definiciones, Acrónimos, y Abreviaciones	25
6.2.Posicionamiento	26
6.2.1.Oportunidad de Negocio	26
6.2.2.Declaración del Problema.....	27
6.2.3.Declaración del Producto	27
6.3.Descripción de los Usuarios y Trabajadores del Negocio.....	27
6.3.1.Descripción de Usuarios	28
6.3.2.Descripción de Trabajadores del Negocio	29
6.3.3.Perfil de los Usuarios del Negocio	29
6.3.3.1.Usuario Solicitante.....	29
6.3.3.2.Asistente Desarrollo	29
6.3.3.3.Asistente de QA	29
6.3.3.4.Gerente de Sistemas.....	30
6.3.3.5.Jefatura Desarrollo	30
6.3.3.6.Asistente de Operaciones.....	30
6.3.4.Perfiles de los Trabajadores del Negocio	31
6.3.5.Necesidades Claves de los Trabajadores del Negocio y Usuarios	31
6.4.Perspectivas del Producto	31
6.4.1.Resumen de Capacidades.....	31
6.5.Definición de Reglas del Negocio	32
6.6.Arquitectura del Negocio	33
6.6.1.Representación Arquitectónica	33
6.6.1.1.Visión del Cliente	33
6.6.2.Visión de la Organización	33
6.6.2.1.Estructura de la Organización.....	33



6.6.2.2.Realización de los Casos de Uso del Negocio	34
6.6.2.3.Patrones Generales de Comportamiento.....	35
6.7.Especificación de Requerimientos.....	35
6.7.1.Casos de Uso y Actores.....	35
6.7.1.1.Lista de Actores	35
6.7.1.2.Lista de Casos de Uso	35
6.7.2.Requerimientos	36
6.7.2.1.Funcionales	36
6.7.2.2.No Funcionales	37
6.7.3.Requerimientos de Software.....	37
6.7.4.Requerimientos de Hardware	37
6.7.5.Interfaces de Usuario	37
6.7.6.Estándares Aplicados	40
6.7.6.1.Estándares Base de Datos	40
6.7.6.2.Estándares de Programación.....	40
6.7.7.Diseño de Casos de Uso.....	41
6.7.8.Especificación de Casos de Uso	41
6.8.Análisis y Diseño de la Propuesta	46
6.8.1.Arquitectura Física	46
6.8.1.1.Capa Presentación	46
6.8.1.2.Capa del Servidor de Aplicación y DBB.....	46
6.8.2.Arquitectura Lógica	46
6.8.2.1.Capa de Datos	46
6.8.2.2.Capa de Lógica del Negocio	47
6.8.2.3.Capa de Presentación	47
6.8.3.Modelo de Análisis	47
6.8.4.Modelo de Diseño	48
6.8.5.Modelo de Implementación	48
6.8.6.Modelo Lógico	49
6.8.7.Modelo Físico.....	49
6.8.8.Diseño de Interfaces	50
6.8.8.1.Login.....	50
6.8.8.2.Cambio de Clave	50



6.8.8.3. Página de Inicio o Menú Principal	51
6.8.8.4. Mantenimiento Empresa.....	51
6.8.8.5. Ingreso de Proyectos	52
6.9. Plan Maestro de Pruebas.....	52
6.10. Diseños de Casos de Prueba	53
6.11. Manuales y Entregables	53
6.12. Capacitación a Usuarios.....	54
Anexos	56
Anexo 01: Bibliografía Básica.....	56
Anexo 02: Net Gráfica	56
Anexo 03: Encuesta	56
Anexo 04: Diseño de Casos de Prueba	59
Anexo 05: Script Base de Datos	61
Anexo 06: Manual de Implementación	85
Anexo 07: Manual de Usuario	98
Anexo 08: Manual Técnico	108
Anexo 09: Cronograma Capacitación	117



Índice de Cuadros

Cuadro 1: Calculo de Población	10
Cuadro 2: Operacionabilidad de las Variables.....	11
Cuadro 3: Declaración del Problema	27
Cuadro 4: Declaración del Producto	27
Cuadro 5: Descripción de los Usuarios del Negocio	28
Cuadro 6: Usuario Solicitante	29
Cuadro 7: Asistente Desarrollo	29
Cuadro 8: Asistente de QA.....	29
Cuadro 9: Gerente de Sistemas	30
Cuadro 10: Jefatura Desarrollo	30
Cuadro 11: Asistente Operaciones	30
Cuadro 12: Necesidades Claves de los Trabajadores y Usuarios del Negocio	31
Cuadro 13: Resumen de Capacidades	31
Cuadro 14: Requerimientos de Software	37
Cuadro 15: Requerimientos Hardware	37
Cuadro 16: Caso de Uso UC001	41
Cuadro 17: Caso de Uso UC002	42
Cuadro 18: Caso de Uso UC003	42
Cuadro 19: Caso de Uso UC004	43
Cuadro 20: Caso de Uso UC005	43
Cuadro 21: Caso de Uso UC006	44
Cuadro 22: Caso de Uso UC007	44
Cuadro 23: Caso de Uso UC008	45



Índice de Tablas

Tabla 1: Tabulación resultados pregunta 1	12
Tabla 2: Tabulación resultados pregunta 2	13
Tabla 3: Tabulación resultados pregunta 3	14
Tabla 4: Tabulación resultados pregunta 4	15
Tabla 5: Tabulación resultados pregunta 5	16
Tabla 6: Tabulación resultados pregunta 6	16
Tabla 7: Tabulación resultados pregunta 7	17
Tabla 8: Tabulación resultados pregunta 8	18
Tabla 9: Tabulación resultados pregunta 9	19
Tabla 10: Tabulación resultados pregunta 10	20



Índice de Gráficos

Ilustración 1: Grafico estadístico resultados pregunta 1.....	13
Ilustración 2: Grafico estadístico resultados pregunta 2.....	14
Ilustración 3: Grafico estadístico resultados pregunta 3.....	15
Ilustración 4: Grafico estadístico resultados pregunta 4.....	15
Ilustración 5: Grafico estadístico resultados pregunta 5.....	16
Ilustración 6: Grafico estadístico resultados pregunta 6.....	17
Ilustración 7: Grafico estadístico resultados pregunta 7.....	18
Ilustración 8: Grafico estadístico resultados pregunta 8.....	19
Ilustración 9: Grafico estadístico resultados pregunta 9.....	19
Ilustración 10: Grafico estadístico resultados pregunta 10.....	20
Ilustración 11: Organigrama COLVIDA S.A.....	34
Ilustración 12: Diagrama de Casos de Uso	41
Ilustración 13: Diagrama Arquitectura Física	46
Ilustración 14: Diagrama de Secuencia UC001	47
Ilustración 15: Diagrama Colaboración UC0010.....	48
Ilustración 16: Diagrama de clases	48
Ilustración 17: Diagrama de Implementación	48
Ilustración 18: Diagrama Modelo Lógico.....	49
Ilustración 19: Diagrama Modelo Físico	49
Ilustración 20: Login - Ricardo Morales	50
Ilustración 21: Cambio de Clave	50
Ilustración 22: Página de Inicio	51
Ilustración 23: Mantenimiento de Empresa	51
Ilustración 24: Ingreso de Clientes	52



Resumen

Los conceptos de calidad y mejoramiento continuo que rigen dentro de seguros COLVIDA S.A. son tan necesarios ya que están enfocados en brindar un buen servicio tanto al cliente interno y con mayor razón al cliente externo. Lo cual ha llevado a las diferentes áreas de la compañía a mejorar día a día sus procesos. En tal razón el área de desarrollo se avisto afectada al no contar con un buen método para la administración de sus proyectos. En vista de esto el presente trabajo tiene como objetivo realizar un estudio de la administración de proyectos en el control de versiones de productos de software para concluir con el desarrollo de un sistema que automatice los procesos de administración de proyectos y versionamiento de software el mismo que se acople a las necesidades del área. Dentro del presente escrito podremos encontrar el estudio realizado a los procesos que maneja el área de desarrollo, así como también los objetivos que se pretende alcanzar, de igual manera se encuentran detallados todos los estudios y las fases de desarrollo del sistema a implementarse.



CAPÍTULO I

El Problema

1.1. Planteamiento del Problema

Seguros COLVIDA S.A. es una compañía con más de 10 años en el mercadeo masivo de seguros de vida de personas en Ecuador, Seguros COLVIDA S.A. a través de los productos y programas de mercadeo, ofrece en forma personalizada un seguro de vida voluntario acorde a las necesidades individuales y familiares de los ecuatorianos.

Actualmente la compañía de seguros de vida COLVIDA S.A. cuenta con el departamento de sistemas el cual se encuentra dividido en dos áreas: Operaciones y Desarrollo de Software.

El área de operaciones es la encargada de dar soporte tecnológico a usuarios, así como también velar por el buen funcionamiento de los sistemas de información y servidores que existen dentro de la compañía.

El área de desarrollo de software es la encargada de realizar las mejoras y adecuaciones a los sistemas de información que se manejan dentro de la compañía, las mismas que se presentan en el día a día con la realización de sus actividades cotidianas.

Dentro del área de desarrollo se maneja un proceso en el cual se da seguimiento a las fases del ciclo de vida de desarrollo de software para las mejoras y adecuaciones a los sistemas de información que solicitan los usuarios de mismos.

Dicho proceso presenta varios inconvenientes en cuanto a la administración de proyectos ya que la forma de archivado de la documentación de estos se la lleva de manera física, lo cual genera retrasos en la entrega de información cuando se requiere conocer datos específicos de un proyecto, y que se tarda demasiado tiempo buscando en los archivos físicos.

El versionamiento de software es llevado en un archivo Excel el cual es



manejado por varias personas, por esta razón se ha visto afectado este proceso ya que se cometen errores como la falta de información en el archivo y al momento de asignar una nueva versión a los sistemas de información, lo cual con lleva un retraso a la productividad del área.

Otro de los inconvenientes es cuando un proyecto tiene que ser reprocesado, lo cual en muchas ocasiones ha generado que se duplique la información y no se pueda dar un correcto seguimiento al mismo.

1.2. Formulación del Problema

¿De qué manera la administración de proyectos influirá en el control de versiones de productos de software?

1.3. Objetivos

1.3.1. General

Analizar de qué manera la administración de proyectos ayudara al control de versiones de productos de software del área de desarrollo de la compañía de seguros de vida COLVIDA.

1.3.2. Específicos

- Estudiar los procesos que maneja el área de desarrollo de software para la administración y versionamiento de proyectos y automatizarlos.
- Mejorar los procesos de administración y versionamiento de software ayudando al manejo seguro de la información concerniente a cada proyecto asignado al área.
- Incluir mecanismos que permitan al usuario solicitante conocer el estado de cada uno de los proyectos que solicite al área.
- Implementar un sistema para la administración, versionamiento y control de productos de software.

1.4. Justificación e Importancia

La funcionalidad del área de desarrollo dentro de la compañía de seguros de vida COLVIDA S.A. depende de los servicios oportunos y eficientes que de la misma, para lograr satisfacer las necesidades que se presentan día a día en los sistemas de información que se manejan dentro de la compañía.

Para lograr sus propósitos el área de desarrollo necesita mejorar la forma de administrar los proyectos asignados, al igual que de un sistema que le permita llevar de mejor manera los procesos de control de versiones de



productos de software.

Un aspecto clave para la implementación del sistema en el área de desarrollo, es la visión de brindar un servicio de calidad al usuario final, ya que los buenos servicios brindados sobre todo la buena atención y la eficiencia en el manejo de la información harán que el área de desarrollo se vea bien dentro de la compañía.



CAPÍTULO II

Marco Teórico

2.1. Antecedentes

Según el artículo Revisión de los Sistemas de Control de Versiones Utilizados en el Desarrollo de Software, escrito por Edgar Tello, Claudia M. Rosa y Diego A. Tello:

En la actualidad, la industria del software juega un papel cada vez más importante para la economía global. El software ha transformado los procesos de control de la mayoría de los servicios de los cuales dependemos. Cada día surgen más y mejores tecnologías y con ellas novedosas aplicaciones, generando nuevos retos para los implicados en los procesos de software.

En el área de desarrollo de software el disponer de una herramienta de control de versiones que permita el registro de las modificaciones que se hayan realizado a los programas o documentos, y que proporcione las versiones previas del software, puede evitar conflictos al momento de generar un producto final.

El control de versiones es el proceso de almacenar y recuperar cambios de un proyecto de desarrollo. Los sistemas de control de versiones SCV, permiten disponer de una versión anterior para corregir errores o actualizar funciones. Dentro de sus funcionalidades está el conservar las versiones que se hayan generado a través del tiempo, así como los diferentes archivos que integran el proyecto en cuestión, uniendo en forma automática las aportaciones de los integrantes de un equipo de trabajo. Los SCV pueden ser utilizados en muchos entornos, tales como el gestionar las versiones de documentos generados por procesadores de texto, presentaciones multimedia, archivos del sistema, correo electrónico y código fuente, por mencionar algunos.

Los SCV también son identificados como sistemas de control de código fuente (Source Control Code Systems, SCCS), sistemas de gestión de



control de código (Source Control Management Systems, SCMS), sistemas de control de revisiones (Revision Control Systems, RCS) o gestión de configuración de software (Software Configuration Management SCM). En el presente trabajo de investigación se hace referencia a los SCV enfocándolos exclusivamente a los sistemas, herramientas o aplicaciones desarrolladas para el control de versiones en el ambiente de desarrollo de software.

Según el artículo Herramientas de Control de Código Fuente, escrito por el Ing. Armando Cabrera:

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas (por ejemplo, para algún cliente específico).

El control de versiones se realiza principalmente en la industria informática para controlar las distintas versiones del código fuente. Sin embargo, los mismos conceptos son aplicables a otros ámbitos como documentos, imágenes, sitios web, etcétera.

Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión (CVS, Subversion, SourceSafe, ClearCase, Darcs, Plastic SCM, Git, Mercurial, etc.).

Un sistema de control de versiones debe proporcionar:

- Mecanismo de almacenaje de los elementos que deba gestionar (ej. archivos de texto, imágenes, documentación, etc.).
- Posibilidad de realizar cambios sobre los elementos almacenados (ej. modificaciones parciales, añadir, borrar, renombrar o mover elementos).
- Registro histórico de las acciones realizadas con cada elemento o conjunto de elementos (normalmente pudiendo volver o extraer un estado anterior del producto).



2.2. Fundamentación Teórica

La administración de proyectos es un proceso continuo. Este proceso requiere de una estrategia global, apoyada por herramientas de trabajo que incrementen la productividad. El propósito de planificar y controlar es proveer una propuesta uniforme para el desarrollo y la administración de los proyectos. Los planes deben apoyar los niveles estratégicos, tácticos y operacionales de las organizaciones con el fin de alcanzar las metas corporativas de largo, mediano y corto plazo.

A través del ciclo de vida de un proyecto, se conforman dos categorías de actividades a realizar y que se encuentran directamente relacionadas: las actividades de gestión y las actividades de desarrollo del sistema.

Las actividades de gestión son aquellas relacionadas con la administración de las organizaciones, personas, sistemas y procedimientos comprometidos en el proceso de planificación y construcción del sistema. La planificación del proyecto, junto con las actividades de control, es iterada para cada fase del proyecto y proveen de la estrategia de administración con la cual las actividades de desarrollo del sistema son estimadas, programadas y ejecutadas.

Las actividades de desarrollo del sistema se centran en el desarrollo mismo. Las metodologías de desarrollo están típicamente organizadas en distintas fases, agrupadas en áreas funcionales de estudio, diseño y construcción, basadas en una estructura de partición del trabajo.

Se llama control de versiones o versionamiento a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra dicho producto en un momento dado de su desarrollo o modificación.

El versionamiento de productos de software es el proceso de asignación de un nombre o número único a un software para indicar su nivel de desarrollo. Generalmente se asigna dos o tres números que van incrementando conforme el desarrollo del software aumenta y se requiera la asignación de un nuevo nombre o número único. Una manera de dar un versionamiento sería la siguiente:

Para el formato de versionamiento:

mayor.menor.micro

Ejemplo: 1.2.1



Se aumenta el número cuando:

- **mayor:** el software sufre grandes cambios y mejoras.
- **menor:** el software sufre pequeños cambios y/o correcciones de errores.
- **micro:** se aplica una corrección al software, y a su vez sufre pocos o ningún cambio.

2.3. Marco Legal

Ley de Propiedad Intelectual:

La autoría y derechos de autor por un producto original, nuevo da protección a un privilegio de propiedad intelectual los mismos que pueden ser de muchos tipos.

Artículo. 8. La protección del derecho de autor recae sobre todas las obras del ingenio, en el ámbito literario o artístico, cualquiera que sea su género, forma de expresión, mérito o finalidad. Los derechos reconocidos por el presente título son independientes de la propiedad del objeto material en el cual está incorporada la obra y su goce o ejercicio no están supeditados al requisito del registro o al cumplimiento de cualquier otra formalidad

Artículo. 28. los programas de ordenador se consideran obras literarias y se protegen como tales. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresados, ya sea en forma legible por el hombre (código fuente) o en forma legible por máquina (código objeto), ya sean programas operativos y programas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia y organización del programa.

Artículo. 30. la adquisición de un ejemplar de un programa de ordenador que haya circulado lícitamente, autoriza a su propietario a realizar exclusivamente: Una copia de la versión del programa legible por máquina (código objeto) con fines de seguridad o resguardo; fijar el programa en la memoria interna del aparato, ya sea que dicha fijación desaparezca o no al apagarlo, con el único fin y en la medida necesaria para utilizar el programa; y, salvo prohibición expresa, adaptar el programa para su exclusivo uso personal, siempre que se limite al uso normal previsto en la licencia. El adquirente no podrá transferir a ningún título el soporte que contenga el programa así adaptado, ni podrá utilizarlo de ninguna otra forma sin autorización expresa, según las reglas generales.



2.4. Glosario de Términos

Implementar: Poner en funcionamiento, aplicar los métodos y medidas necesarios para llevar algo a cabo.

Release: puede considerarse en ocasiones como un sinónimo de versionamiento de todas maneras es una versión de un software aunque se utiliza este término cuando la versión del software se hace pública.

Variable: Es un fenómeno, hecho proceso propiedad atributo que puede variar cualitativamente y cuantitativamente.

Variable dependiente: Es la que cambia por influencia de la variable independiente (es el consecuente o efecto).

Variable independiente: Es aquella que genera cambios en otra variable (es el antecedente o causa).

Versionamiento: Se refiere a la acción de realizar una nueva versión de algo, especialmente si es una obra artística, musical o software. También se lo puede definir como el proceso de asignación de un nombre o número único a un software para indicar su nivel de desarrollo

2.5. Caracterización de Variables

Variable Independiente: La Administración de Proyectos.

Variable Dependiente: Control de Versiones de Productos de Software.

2.6. Preguntas Directrices

¿Cómo mejorará el control de versiones de productos de software con la administración de proyectos?

¿Cómo dar seguimiento al control de versiones de productos de software?

¿Qué se busca con la administración de proyectos?

¿Quiénes van a ser los beneficiados con la administración de proyectos en el control de versiones de productos de software?



CAPÍTULO III

Marco Metodológico

3.1. Tipo de Investigación

En este proyecto, el problema planteado denota un conocimiento muy amplio de la materia, lo que nos ha permitido identificar elementos y características básicas de la investigación, además de facilitar el planteamiento de nuevas hipótesis de solución que ha permitido iniciar el conocimiento de tipo explicativo, en búsqueda de una solución al problema que presenta el área de desarrollo de software de la compañía de seguros de vida COLVIDA.

3.1.1. Investigación Aplicada

Se empleó este tipo de investigación ya que se busca dar solución al problema planteado por el área de desarrollo de software, durante la ejecución y desarrollo de esta investigación se fortalecieron nuestras capacidades para la resolución de los problemas mencionados, los que en forma práctica permitieron poner nuestras destrezas en funcionamiento a través de la utilización de los conceptos doctrinarios aprendidos durante la capacitación profesional y estos colocarlos en una plataforma práctica de ejecución a fin de obtener precisión y confiabilidad en los resultados.

La aplicación de este método de investigación nos ayudó con el desarrollo del proyecto, inicialmente en la realización de encuestas para transformar los datos en información, ya que mediante la utilización de estas herramientas propias de la investigación científica se puede llegar a determinar claramente los procesos que se requieren materializar en un esquema ordenado que permita conocer cuál es la necesidad real de la aplicación de éste proyecto.



3.2. Población y Muestra

La población de la compañía de seguros de vida COLVIDA esencia de esta investigación se eleva a un número de 15 personas en los niveles de área de operaciones, área de desarrollo y usuarios solicitantes.

Cuadro 1: Calculo de Población

Formula	Datos
$N = \text{Tamaño de la muestra}$ $E = \text{Error de muestreo}$	$\frac{15}{15 - 1 * 0.0025 + 1} = 1.268$
$n = \frac{N}{(N - 1)E^2 + 1}$	$N = 1.268$

Fuente: Ricardo Morales

Por contar con una muestra demasiado baja se encuestara a toda la población existente, la muestra será de 15 personas.

3.3. Técnicas de Recolección de Información

La aplicación de técnicas de recolección de información permitirá obtener datos que al ser procesados determinarán la factibilidad del proyecto, básicamente trabajaremos con encuestas que se aplicarán en los diferentes áreas de la compañía de seguros de vida COLVIDA.

Las técnicas de investigación se aplican al proyecto con el fin de obtener datos principales acerca de la compañía, los mismos que permitirán realizar una observación de la infraestructura, ambiente de trabajo, y en fin datos iniciales que determinen el tipo de organización tecnológica en la que se va a desarrollar el sistema, para la ejecución de la técnica de recolección de información es necesario estar presente en el lugar donde se va efectuar el levantamiento de información por tanto se debe hacer una análisis visual de las áreas en donde se va a obtener la información y las relaciones que existan con el resto de áreas.

3.3.1. La Encuesta

La encuesta es un instrumento importante de la investigación, misma que se realizará en la Compañía de Seguros de Vida COLVIDA, además de obtener la información por parte del encuestado para poder desarrollar el control y versionamiento de proyectos, se podrá observar la realidad de la compañía y conocer las políticas y procesos a seguir para el buen desenvolvimiento del proyecto.

El modelo de encuesta aplicado a la población de estudio se encuentra



detallado en al Anexo 03.

3.4. Operacionabilidad de las Variables

Cuadro 2: Operacionabilidad de las Variables

Variables	Dimensión	Indicadores
Variable Independiente: La Administración de Proyectos	Tecnológico	¿Cómo ayudará la administración de proyectos?
Variable Dependiente: Control de Versiones de Productos de Software	Tecnológico	¿De qué manera mejorará el control de versiones de productos de software con la implementación de un sistema de información?

Fuente: Ricardo Morales



CAPÍTULO IV

Análisis e Interpretación de Resultados

4.1. Análisis de Encuestas

La aplicación de este instrumento de investigación en la compañía de seguros de vida COLVIDA, nos ha permitido obtener los siguientes resultados los mismos que se utilizarán en el desarrollo del proyecto.

Desarrollo de Resultados:

Pregunta 1:

Tabla 1: Tabulación resultados pregunta 1

¿Al momento el área de desarrollo de software utiliza herramientas informáticas para realizar los procesos de administración y versionamiento de proyectos?		
Parámetros	Cantidad	Porcentaje
Si	8	54%
No	2	13%
No sabe/No opina	5	33%
Total	15	100%

Fuente: Ricardo Morales



1. ¿Al momento el área de desarrollo de software utiliza herramientas informáticas para realizar los procesos de administración y versionamiento de proyectos?

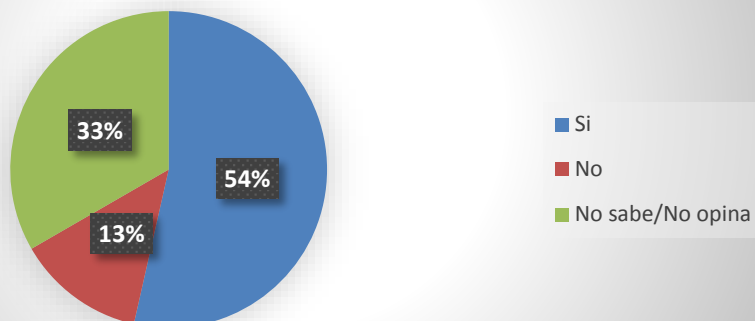


Ilustración 1: Grafico estadístico resultados pregunta 1
Fuente: Ricardo Morales

Interpretación de resultados: En esta pregunta se logra establecer a través de las respuestas de los encuestados que, el 54% de las personas utilizan herramientas informáticas para el desarrollo de sus actividades cotidianas, las respuestas negativas o en duda que representan el 46% de los resultados no reflejan en forma certera la falta de cultura informática sino su desconocimiento a los procesos del trabajo diario.

Pregunta 2:

Tabla 2: Tabulación resultados pregunta 2

¿Qué tipo de herramienta informática utiliza el área de desarrollo para los procesos de administración y versionamiento de proyectos?		
Parámetros	Cantidad	Porcentaje
Ofimática	10	67%
Sistema de Información	0	0%
No sabe/No opina	5	33%
Total	15	100%

Fuente: Ricardo Morales

2. ¿Qué tipo de herramienta informática utiliza el área de desarrollo para los procesos de administración y versionamiento de proyectos?

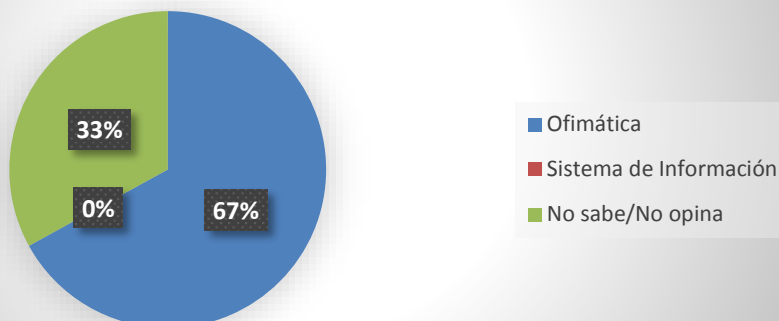


Ilustración 2: Grafico estadístico resultados pregunta 2
Fuente: Ricardo Morales

Interpretación de resultados: Luego de tabular las respuestas de esta pregunta se puede interpretar que el 67% de las personas encuestadas utiliza herramientas ofimáticas para los procesos que maneja el área, mientras que el 33% no saben qué tipo de herramienta se ocupa para dichos procesos.

Pregunta 3:

Tabla 3: Tabulación resultados pregunta 3

¿Cree que las herramientas informáticas a implementarse mejoraran los procesos de administración y versionamiento de proyectos?		
Parámetros	Cantidad	Porcentaje
Si	12	80%
No	0	0%
No sabe/No opina	3	20%
Total	15	100%

Fuente: Ricardo Morales

3. ¿Cree que las herramientas informáticas a implementarse mejoraran los procesos de administración y versionamiento de proyectos?

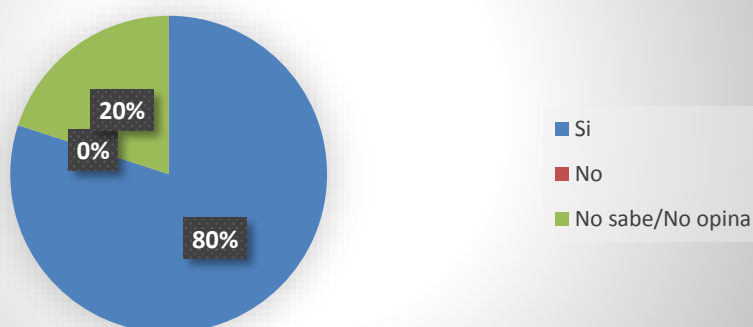




Ilustración 3: Grafico estadístico resultados pregunta 3
Fuente: Ricardo Morales

Interpretación de resultados: Como podemos ver se puede evidenciar que el 80% de las personas encuestadas creen que los procesos que maneja el área de desarrollo mejoraran con la implementación de herramientas informáticas, mientras un 20% no sabe o no opina sobre esta pregunta.

Pregunta 4:

Tabla 4: Tabulación resultados pregunta 4

¿Cómo evalúa el proceso que lleva el área de desarrollo de software para la administración y versionamiento de proyectos?		
Parámetros	Cantidad	Porcentaje
Excelente	3	20%
Bueno	7	47%
Malo	3	20%
No sabe/No opina	2	13%
Total	15	100%

Fuente: Ricardo Morales

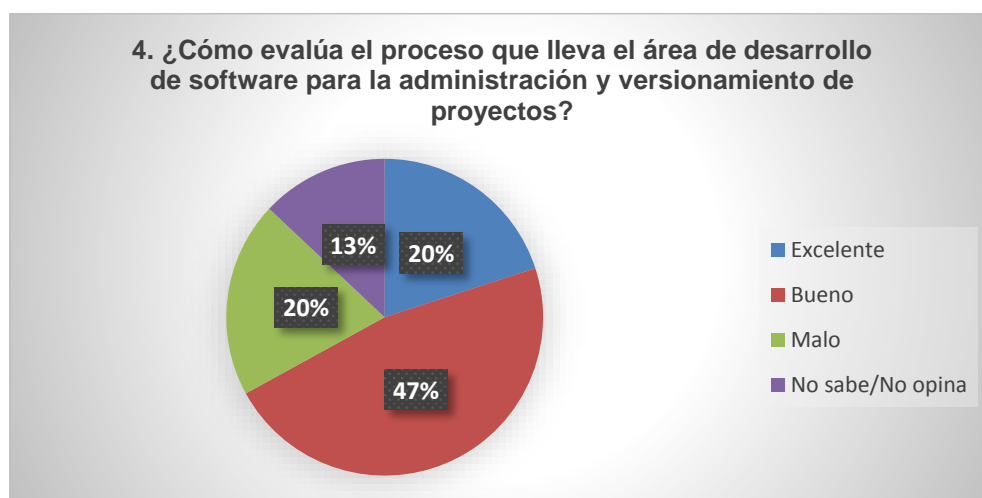


Ilustración 4: Grafico estadístico resultados pregunta 4
Fuente: Ricardo Morales

Interpretación de resultados: Las opiniones en esta pregunta son diversas y denotan un trabajo que a pesar de no contar con sistemas de información se lo considera como bueno e incluso es considerado de excelente por varios usuarios sin embargo hay un porcentaje que considera que el proceso es malo o prefiere no opinar.



Pregunta 5:

Tabla 5: Tabulación resultados pregunta 5

¿Considera que el área de desarrollo deba implementar un software que permita administrar de forma digital la documentación de cada proyecto?		
Parámetros	Cantidad	Porcentaje
Si	15	100%
No	0	0%
No sabe/No opina	0	0%
Total	15	100%

Fuente: Ricardo Morales

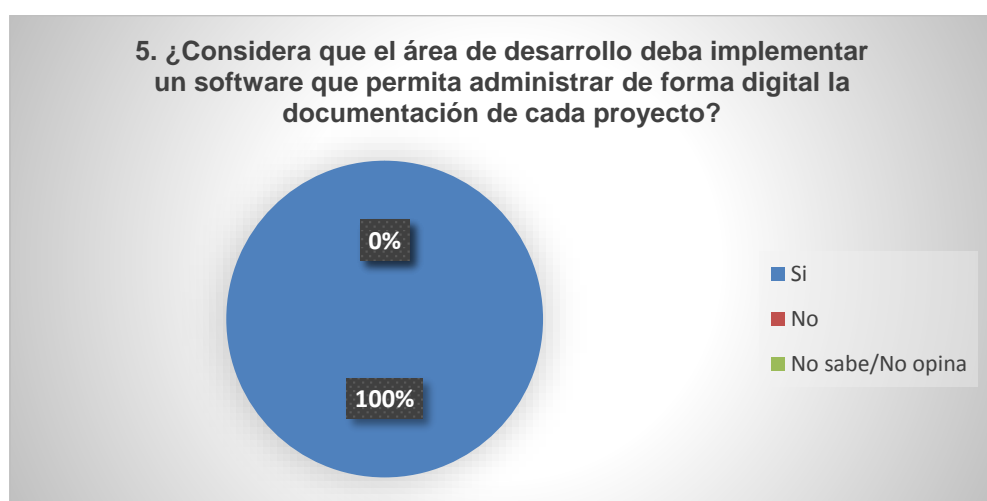


Ilustración 5: Gráfico estadístico resultados pregunta 5

Fuente: Ricardo Morales

Interpretación de resultados: Se puede evidenciar que todos los encuestados consideran que el área de desarrollo debe implementar un software para la administración de la documentación en forma digital de cada proyecto.

Pregunta 6:

Tabla 6: Tabulación resultados pregunta 6

¿Considera que el área de desarrollo deba implementar un software que permita versionar automáticamente cada proyecto al igual que permita llevar un histórico de versiones?		
Parámetros	Cantidad	Porcentaje
Si	10	67%
No	0	0%
No sabe/No opina	5	33%
Total	15	100%

Fuente: Ricardo Morales



6. ¿Considera que el área de desarrollo deba implementar un software que permita versionar automáticamente cada proyecto al igual que permita llevar un histórico de versiones?

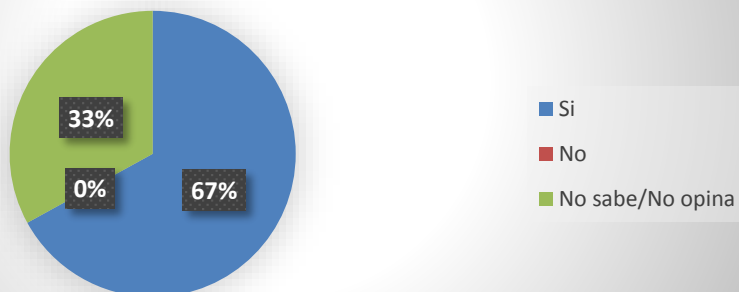


Ilustración 6: Gráfico estadístico resultados pregunta 6
Fuente: Ricardo Morales

Interpretación de resultados: El 67% de las personas encuestadas considera que el área de desarrollo debe implementar un software para el versionamiento automático al igual que permita llevar un histórico de versiones, mientras el 33% restante no opina en esta pregunta ya que no están enterados o no están involucrados con esta parte del proceso.

Pregunta 7:

Tabla 7: Tabulación resultados pregunta 7

¿El nivel de información que recibe sobre el estado de su proyecto es?		
Parámetros	Cantidad	Porcentaje
Excelente	1	7%
Bueno	10	67%
Malo	2	13%
No sabe/No opina	2	13%
Total	15	100%

Fuente: Ricardo Morales



Ilustración 7: Grafico estadístico resultados pregunta 7

Fuente: Ricardo Morales

Interpretación de resultados: En esta pregunta hay diversas opiniones en cuanto a la información que se da al usuario del estado de sus proyectos presentados al área de desarrollo por lo cual se puede evidenciar que un 67% considera bueno sin embargo hay porcentajes q consideran malo o simplemente no opinan.

Pregunta 8:

Tabla 8: Tabulación resultados pregunta 8

¿Cree usted que el área de desarrollo deba incorporar un sistema de información para conocer el estado de cada uno de los proyectos presentados?		
Parámetros	Cantidad	Porcentaje
Si	15	100%
No	0	0%
No sabe/No opina	0	0%
Total	15	100%

Fuente: Ricardo Morales

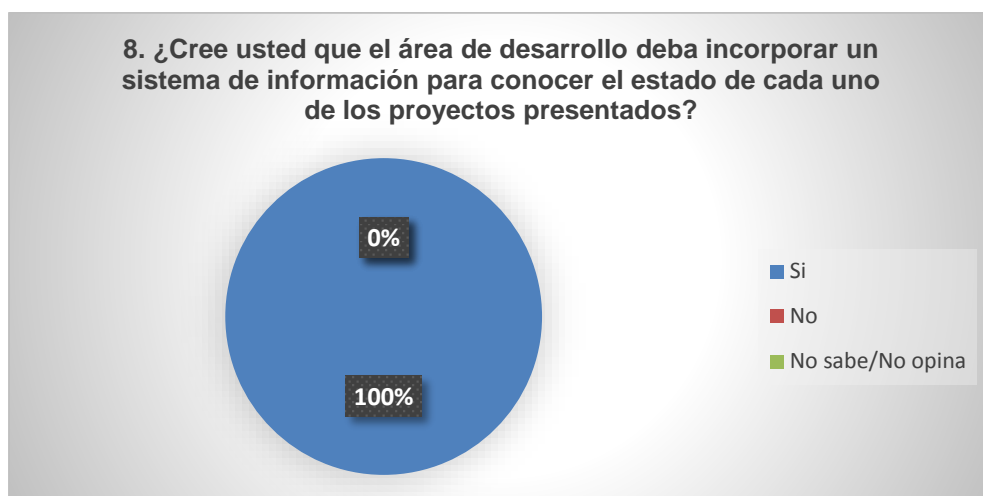


Ilustración 8: Grafico estadístico resultados pregunta 8
Fuente: Ricardo Morales

Interpretación de resultados: Como se puede observar en la tabulación de resultados de esta pregunta todos los encuestados consideran que el área de desarrollo debe implementar un sistema el mismo que permita conocer el estado de cada uno de los proyectos presentados al área.

Pregunta 9:

Tabla 9: Tabulación resultados pregunta 9

¿El área de desarrollo requiere un software que permita la administración y versionamiento de proyectos?		
Parámetros	Cantidad	Porcentaje
Si	15	100%
No	0	0%
No sabe/No opina	0	0%
Total	15	100%

Fuente: Ricardo Morales

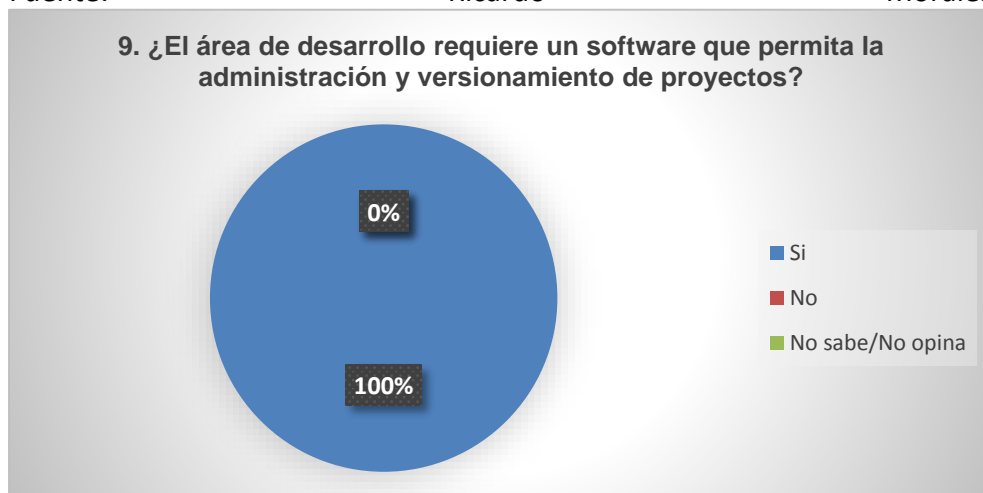


Ilustración 9: Grafico estadístico resultados pregunta 9



Fuente: Ricardo Morales

Interpretación de resultados: Todos los encuestados consideran que el área de desarrollo requiere de un software para los procesos de administración y versionamiento de software.

Pregunta 10:

Tabla 10: Tabulación resultados pregunta 10

¿En qué nivel cree usted que mejorara los procesos de administración y versionamiento de proyectos del área de desarrollo con la implementación de un software?		
Parámetros	Cantidad	Porcentaje
Excelente	9	60%
Bueno	6	40%
Malo	0	0%
No sabe/No opina	0	0%
Total	15	100%

Fuente: Ricardo Morales

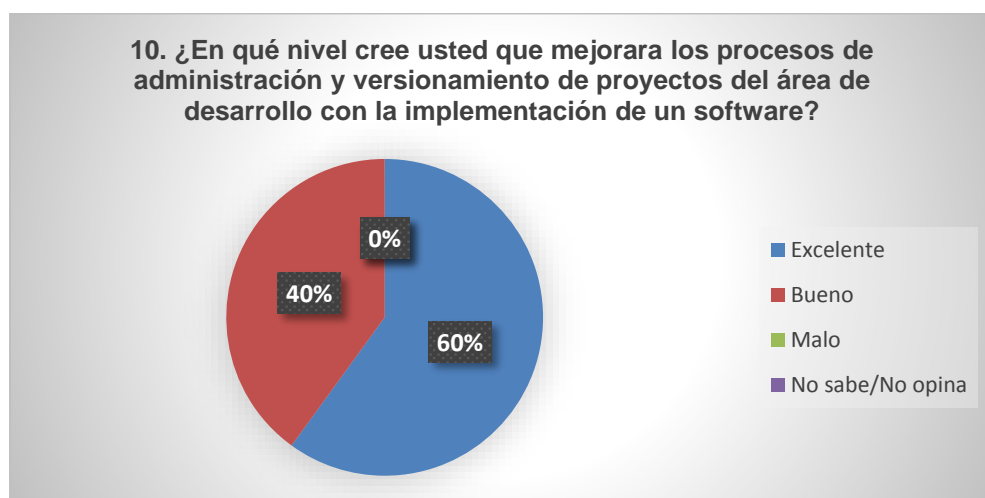


Ilustración 10: Gráfico estadístico resultados pregunta 10

Fuente: Ricardo Morales

Interpretación de resultados: Analizando estos resultados podemos ver que los encuestados consideran que con la implementación de un software para la administración y versionamiento de proyectos en el área de desarrollo de dichos procesos mejoraran sustancialmente.



4.2. Conclusión del Análisis de Datos

En conclusión al trabajo de investigación realizado se puede deducir que el área de desarrollo de la compañía de seguros de vida COLVIDA, al momento tiene una inminente necesidad de implementar un software que dé solución al problema que tiene en la administración y versionamiento de proyectos, dicho software permitiría mejorar los procesos antes mencionados.

En base a las encuestas realizadas se ha podido establecer que los usuarios no manejan los mismos procesos, por dicha razón varias preguntas de las encuestas recibieron respuestas de no saber o no opinar.



CAPÍTULO V

Conclusiones y Recomendaciones

5.1. Conclusiones

- En la investigación se pudo identificar que los procesos que maneja el área de desarrollo para la administración y versionamiento de proyectos son hechos manualmente o en herramientas ofimáticas lo cual genera pérdida de información y retraso en los procesos.
- Con la investigación realizada también podemos concluir que la evaluación que dan los usuarios al proceso que maneja el área es de buena ya que la información al ser archivada físicamente corre riesgos de extraviarse o dañarse.
- También se puede concluir que existe poca o nada información sobre el estado de los proyectos al usuario que solicita una modificación a los sistemas.
- Los procesos de administración y versionamiento de proyectos del área de desarrollo al momento no cuentan con un sistema de información apropiado para estas actividades.

5.2. Recomendaciones

- Se recomienda automatizar los procesos de administración y versionamiento de proyectos del área de desarrollo para así evitar la pérdida de información.
- Con la automatización de los procesos se podrá mejorar la calificación que los usuarios dan al proceso que maneja el área.
- Se recomienda mejorar el proceso de tal manera que el usuario pueda estar informado constantemente del estado de los proyectos presentados al área.



- Con el desarrollo e implementación de un sistema de información el cual gestione y versione los proyectos que son asignados al área de desarrollo se podrán solucionar los conflictos que tiene actualmente el área, además de poder llevar un histórico de las modificaciones que se realizan a los sistemas de información que maneja la compañía de seguros COLVIDA S.A.



CAPÍTULO VI

La Propuesta

6.1. Introducción

6.1.1. Propósito

El desarrollo de este proyecto tiene la finalidad de automatizar los procesos que maneja el área de desarrollo de la compañía de seguros de vida COLVIDA S.A. para la administración de proyectos y versionamiento de software. Con el desarrollo del sistema se mejorara la administración de proyectos asignados al área.

6.1.2. Objetivo

- Implementar un Sistema de Versionamiento y Control de Productos de Software.
- Establecer que el desarrollo de la aplicación cumpla con los tiempos previstos tanto por la compañía y la coordinación de proyectos del instituto.
- Verificar que la utilidad del sistema que se va a desarrollar para el área de desarrollo sirva de apoyo en los procesos que maneja el área.

6.1.3. Alcance

El software que se va a diseñar e implantar realizara el versionamiento, así como también dará seguimiento a cada proyecto asignado al área de desarrollo en las diferentes etapas de su construcción las mismas que se detallan en el procedimiento que maneja el área los módulos que se desarrollaran son los siguientes:

- **Módulo de Seguridad:** El acceso a la información que maneje el sistema es de mucha importancia, por tal motivo para acceder a ella necesariamente y obligadamente se requerirá de una autenticación mediante la cual se establecerán permisos y acceso de acuerdo a los roles del usuario.



- **Módulo de Mantenimiento de Tablas Maestras:** Este módulo consta de las acciones principales (insertar, modificar y eliminar registros) a las tablas maestras que maneja el sistema en la base de datos como son: usuarios, roles, accesos, tipos de componentes, etc.
- **Módulo Transaccional:** En este módulo se encontrarán todas las acciones para automatizar el proceso que lleva el área de desarrollo como son: ingreso, aprobación, designación, desarrollo, pruebas, implementación y evaluación de proyectos.
- **Módulo de Reportes:** Para la toma de decisiones, dentro de este módulo se podrán obtener informes estadísticos de proyectos en un determinado tiempo, así como también informes de un determinado proyecto.

6.1.4. Definiciones, Acrónimos, y Abreviaciones

Ambiente de Desarrollo.- Es el lugar en el cual se desarrollan cada uno de los requerimientos solicitados por los usuarios se utilizará la abreviatura AD.

Ambiente de Producción.- Es el lugar en el cual se encuentra el sistema que es usado para los procesos que se realizan dentro de la compañía diariamente se utilizará la abreviatura AP.

Ambiente de Pruebas.- Es el lugar en el cual se realizan pruebas de cada uno de los proyectos desarrollados se utilizará la abreviatura APR.

Asistente de Calidad de Software.- Es el encargado de realizar las pruebas a cada proyecto que es enviado al área de desarrollo de software se usará la abreviatura QA.

Caso de Uso.- Descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Se usará la abreviatura UC.

DBB.- Siglas de para identificar una Base de datos.

Diagrama de clases.- Es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas.

Diagrama de iteración.- El diagrama global de las interacciones es un diagrama de comportamiento, más precisamente, uno de los cuatro diagramas de interacción.

Diagrama de secuencias.- Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso.



Mientras que el diagrama de casos de uso permite el modelado de una vista negocio del escenario.

Diseño Técnico.- Es el documento el cual se detallan los cambios que se realizaron en el sistema el mismo es realizado por el desarrollador a cargo del proyecto se usara la abreviatura DT.

Especificación Funcional.- Es el documento que entrega el usuario detallando las mejoras o detalles para la realización de cada proyecto se usara la abreviatura EF.

Manual de Implementación.- Es el documento en el cual se describen los procesos para implementar los cambios realizados al sistema este documento lo realiza el desarrollador a cargo del proyecto se usara la abreviatura MI.

Oracle.- Es un sistema de gestión de base de datos objeto-relacional desarrollado por Oracle Corporation. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando: soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma.

PHP.- Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.

Plan de Pruebas.- Es el documento en el cual se describen las pruebas que deberá realizar QA al proyecto este documento es realizado por el desarrollador a cargo del proyecto se usara la abreviatura PP.

SQL.- Structured Query Language o Lenguaje de Consulta Estructurado, es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

UML.- Siglas Lenguaje de Moldeamiento Universal, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

6.2. Posicionamiento

6.2.1. Oportunidad de Negocio

Al implementar el sistema para el versionamiento y control de productos de software para el área de desarrollo de la compañía de seguros de vida COLVIDA se podrá obtener una información detallada de cada proyecto asignado a esta área, así como también el estado del proyecto en cualquier momento deseado.



Adicionalmente se podrá automatizar el proceso de versionamiento de software el cual es realizado manualmente en la actualidad para cada cambio que se realiza al sistema.

6.2.2. Declaración del Problema

En la siguiente tabla se detalla el problema así como también los principales afectados:

Cuadro 3: Declaración del Problema

El Problema	La deficiencia en los procesos de administración proyectos y versionamiento de software del área de desarrollo de la compañía de seguros de vida COLVIDA S.A.
Afecta	<ul style="list-style-type: none">• Área de Sistemas• Área Técnica• Área de Servicio al Cliente• Área Financiera• Área Administrativa
Impacto del Problema	Al momento de buscar información de un determinado proyecto se pierde demasiado tiempo buscando y en muchos de los casos no se obtienen los resultados deseados, así como también se cometen errores en el versionamiento de cada cambio que se realiza al sistema.
Solución Factible	Sistema para la administración de proyectos y versionamiento de software.

Fuente: Ricardo Morales

6.2.3. Declaración del Producto

Cuadro 4: Declaración del Producto

Para	El área de desarrollo de la compañía de seguros de vida COLVIDA S.A.
Quien	El sistema que se desarrollara para la administración de proyectos y versionamiento de software.
Que	Entre las principales ventajas que tendrá el área con la implementación de este sistema son: mejora en la obtención de información de un proyecto determinado, automatización del versionamiento a los cambios realizados en el sistema, además de la generación de reportes estadísticos.

Fuente: Ricardo Morales

6.3. Descripción de los Usuarios y Trabajadores del Negocio

A continuación se detalla brevemente las funciones y responsabilidades de tienen los trabajadores o usuarios del que intervienen en el negocio para tener una visión más profunda de su vinculación con nuestra investigación.



6.3.1. Descripción de Usuarios

Cuadro 5: Descripción de los Usuarios del Negocio

Cargo	Descripción	Responsabilidad
Usuario Solicitante	Es el encargado de generar las especificaciones funcionales de acuerdo a sus requerimientos.	Generar la especificación funcional de manera que sea clara y entendible.
Asistente Desarrollo	Es el encargado del desarrollo de las especificaciones funcionales, dar seguimiento a los proyectos y solicitar que se realicen pruebas de los mismos.	Programación Documentación de proyectos. Notificación a usuarios. Seguimiento a proyectos. Paso a pruebas de QA.
Asistente de QA	Es el encargado de realizar pruebas a los proyectos entregados por los asistentes de desarrollo además de versionar y solicitar el paso a producción.	Pruebas a proyectos. Versionamiento. Paso a producción.
Gerente de Sistemas	Es el encargado de aprobar todas las especificaciones funcionales y entregarlas al área de desarrollo estableciendo una prioridad.	Aprueba las especificaciones funcionales entregadas por los usuarios. Envío de especificaciones funcionales aprobadas al área de desarrollo.
Jefatura Desarrollo	Es el encargado de asignar los proyectos a los asistentes de desarrollo para su ejecución.	Asignación de proyectos a desarrolladores.
Asistente de Operaciones	Es el encargado de implementar los proyectos enviados por el asistente de QA en el ambiente de	Implementar proyectos en producción. Notificación al asistente de



	producción.	QA y al desarrollador.
--	-------------	------------------------

Fuente: Ricardo Morales

6.3.2. Descripción de Trabajadores del Negocio

Dentro de la investigación no se toma en cuenta a trabajadores del negocio, ya que son los mismos que se encuentran detallados en la descripción de usuarios del negocio.

6.3.3. Perfil de los Usuarios del Negocio

6.3.3.1. Usuario Solicitante

Cuadro 6: Usuario Solicitante

Descripción	Es el encargado de generar las especificaciones funcionales de acuerdo a sus requerimientos.
Tipo	Este usuario tendrá el perfil de Usuario y tendrá privilegios de ingresar y evaluar proyectos, y consultar estado de los proyectos ingresados.
Criterio de Éxito	Ingreso de proyectos al sistema.
Responsabilidades	Cargar especificación funcional en formato PDF. Llenar todos los campos correctamente de ingreso de proyectos.

Fuente: Ricardo Morales

6.3.3.2. Asistente Desarrollo

Cuadro 7: Asistente Desarrollo

Descripción	Es el encargado del desarrollo de las especificaciones funcionales, dar seguimiento a los proyectos y solicitar que se realicen pruebas de los mismos.
Tipo	Tendrá el perfil de Desarrollador y podrá modificar el nombre del proyecto, la clase de proyectos.
Criterio de Éxito	Para mejorar la administración de proyectos este usuario deberá cargar al sistema todos los documentos al sistema.
Responsabilidades	Cargar la documentación de proyectos en formatos PDF. Registrar componentes que fueron modificados.

Fuente: Ricardo Morales

6.3.3.3. Asistente de QA

Cuadro 8: Asistente de QA

Descripción	Es el encargado de realizar pruebas a los proyectos
--------------------	---



	entregados por los asistentes de desarrollo además de versionar y solicitar el paso a producción.
Tipo	Tendrá el perfil de QA, y podrá registrar las veces que fue probado un proyecto al igual que cargar un informe con las pruebas realizadas.
Criterio de Éxito	Ingresa al sistema las observaciones de las pruebas que realice a cada proyecto.
Responsabilidades	Cargar documentación de pruebas realizadas en formato PDF.

Fuente: Ricardo Morales

6.3.3.4. Gerente de Sistemas

Cuadro 9: Gerente de Sistemas

Descripción	Es el encargado de aprobar todas las especificaciones funcionales y entregarlas al área de desarrollo estableciendo una prioridad.
Tipo	Tendrá el perfil de Gerencia, y podrá aprobar o negar los requerimientos ingresados por los usuarios.
Criterio de Éxito	Aprobar en el sistema los proyectos ingresados por el usuario.
Responsabilidades	Establecer prioridades al proyecto. Aprobar o negar proyectos ingresados.

Fuente: Ricardo Morales

6.3.3.5. Jefatura Desarrollo

Cuadro 10: Jefatura Desarrollo

Descripción	Es el encargado de asignar los proyectos a los asistentes de desarrollo para su ejecución.
Tipo	Tendrá el perfil de Jefatura, y podrá asignar los proyectos aprobados por la gerencia.
Criterio de Éxito	Designar con el sistema al desarrollador de los proyectos aprobados.
Responsabilidades	Designar los proyectos aprobados a los desarrolladores.

Fuente: Ricardo Morales

6.3.3.6. Asistente de Operaciones

Cuadro 11: Asistente Operaciones

Descripción	Es el encargado de implementar los proyectos enviados por el asistente de QA en el ambiente de producción.
Tipo	Tendrá el perfil de Operaciones, y podrá indicar que el proyecto fue implementado.
Criterio de Éxito	Usar el sistema para indicar las observaciones o



	pasos a producción según sea el caso.
Responsabilidades	Pasar a producción el proyecto.

Fuente: Ricardo Morales

6.3.4. Perfiles de los Trabajadores del Negocio

No se describen perfiles de trabajadores del negocio ya que los usuarios son los mismos trabajadores y el perfil de estos ya se describió en los perfiles de usuario.

6.3.5. Necesidades Claves de los Trabajadores del Negocio y Usuarios

Cuadro 12: Necesidades Claves de los Trabajadores y Usuarios del Negocio

Necesidad	Prioridad	Incumbe	Solución actual	Solución propuesta
Versionamiento	Alta	Área de Desarrollo	Se lleva un registro en un archivo Excel.	Automatizar este proceso y eliminar el archivo y llevar dicho registro en una base de datos.
Seguimiento a proyectos	Alta	Área de Desarrollo	Se lleva una carpeta con toda la documentación física de cada proyecto.	Se llevara la documentación en la base de datos conjuntamente con documentos escaneados.

Fuente: Ricardo Morales

6.4. Perspectivas del Producto

6.4.1. Resumen de Capacidades

Cuadro 13: Resumen de Capacidades

Beneficios del cliente	Soporte
Versionamiento automático	El versionamiento de cada proyecto se lo realizara de manera automática al realizar el paso a



	producción.
Información de estados de proyectos.	Cada usuario solicitante de un proyecto podrá tener acceso al estado de su proyecto ingresando al sistema.
Notificaciones a email de cambios de estado o de cada proyecto.	El sistema enviara automáticamente emails a los usuarios o responsables de cada proceso que tiene el desarrollo de un proyecto.
Reportes estadísticos	El sistema permitirá generar reportes del estado de proyectos, reportes estadísticos de proyectos elaborados en un rango de fechas.

Fuente: Ricardo Morales

6.5. Definición de Reglas del Negocio

La definición de reglas del negocio será muy concreto y son los siguientes:

- La clave de cada usuario deberá caducar cada 30 días.
- El sistema deberá almacenar un histórico de claves.
- Si la clave es errada en tres intentos se bloqueara el usuario.
- Solo el usuario administrador podrá reactivar usuarios bloqueados.
- Para la modificación de los registros de las tablas maestras del sistema solo se lo podrá hacer con el usuario administrador.
- El administrador tendrá acceso a todas las funciones del sistema.
- Todos los usuarios deberán iniciar sesión para el uso del sistema.
- Para ingresar al sistema cada usuario deberá tener su propio nombre de usuario y contraseña.
- Solo el usuario administrador podrá ingresar nuevos usuarios al sistema.
- Para el registro y aprobación de un proyecto por parte de un usuario solicitante deberá ser anexada la Especificación Funcional.
- Para la aprobación de un proyecto se deberá indicar la prioridad y la fecha en la que deberá ser entregado el proyecto.
- Todos los documentos que se carguen al sistema deberán estar en formato PDF.



- El versionamiento se lo hará de la siguiente manera: si es un proyecto nuevo se aumentara el primer dígito; si es una mejora o adecuación o un reporte se aumentara el segundo dígito; y si es un reproceso se aumentara el tercer dígito. Quedando el formato de número de versión de la siguiente manera: **Eje.: 1.0.0.**
- Para el paso a producción de un proyecto, este no deberá tener observaciones de pruebas de QA.
- En caso de que un proyecto tenga observaciones ya sean de pruebas de QA o de producción deberá ser reenviado al desarrollador asignado para su corrección.
- Para el paso a producción de un proyecto este deberá tener la siguiente documentación: Especificación Funcional, Diseño Técnico, Plan de Pruebas, Cronograma, Manual de Implementación.
- Para dar como finalizado un proyecto este deberá ser evaluado por el usuario solicitante.
- Para el nombre del usuario de ingreso al sistema se ocupara la primera letra de su primer nombre seguido del primer apellido.

6.6. Arquitectura del Negocio

6.6.1. Representación Arquitectónica

6.6.1.1. Visión del Cliente

Para el cliente interno será de mucha utilidad el aplicativo ya que le permitirá estar constantemente informado del estado de su proyecto. Al igual el área de desarrollo se beneficiara al contar con un sistema que le permita versionar automáticamente, además de permitir gestionar los proyectos asignados.

6.6.2. Visión de la Organización

El sistema para el versionamiento y control de productos de software que será implementado en el área de desarrollo generara un impacto significativo para el área ya que esta se beneficiara al contar con información al instante de un proyecto o proyectos determinados.

6.6.2.1. Estructura de la Organización

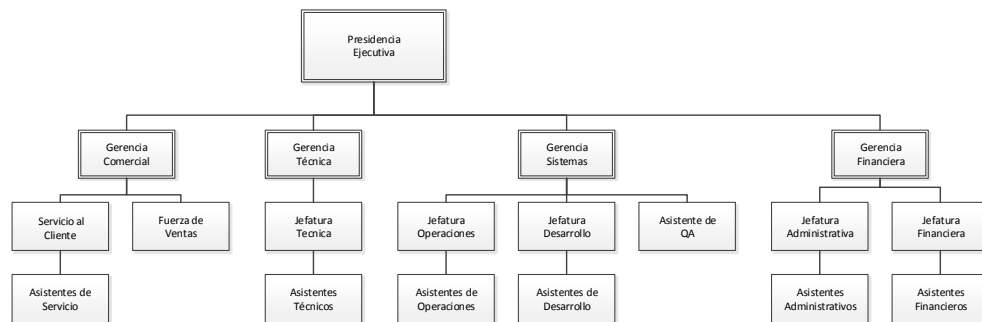


Ilustración 11: Organigrama COLVIDA S.A.

Fuente: Ricardo Morales

Dentro de la compañía de seguros COLVIDA S.A. se maneja el organigrama descrito en la ilustración 11, dentro del mismo se puede observar que el departamento de sistemas está dividido en dos áreas: la de operaciones y la de desarrollo dentro de esta última es el lugar donde se implementará el sistema que se desarrolle.

6.6.2.2. Realización de los Casos de Uso del Negocio

Ingreso de Proyectos.- Es la actividad que realiza el Usuario Solicitante en la cual indica el tipo de proyecto, y anexa la especificación funcional.

Aprobación de Proyectos.- Es la actividad que realiza el gerente de sistemas a las solicitudes enviadas por el usuario solicitante.

Asignación de Proyectos.- Es la actividad que realiza el jefe de desarrollo para la asignación de un desarrollador para la ejecución de la solicitud aprobada.

Desarrollo.- Es la actividad que realiza el asistente de desarrollo al requerimiento asignado lo que comprende la programación como tal y la generación de la documentación del proyecto.

Pruebas de QA.- Es la actividad que realiza el asistente de QA con la finalidad de garantizar la calidad de funcionamiento de los cambios realizados al sistema.

Versiónamiento.- Es el proceso que realiza el asistente de QA, para dar una nueva versión al sistema de la compañía de acuerdo al tipo de proyecto que sea.

Pasó a Producción.- Es la actividad que realiza el asistente de operaciones, la cual consiste en seguir el Manual de Implementación.

Evaluación.- Es la actividad que realiza el Usuario Solicitante donde evalúa si el proyecto está de acuerdo a lo solicitado.



6.6.2.3. Patrones Generales de Comportamiento

Para el correcto funcionamiento del software que se desarrollara tenemos los siguientes patrones generales de comportamiento:

- Es de responsabilidad cuidar la clave que se le proporcione para el ingreso al sistema.
- Para el uso del sistema todo usuario deberá iniciar sesión.
- Cuando un usuario haya terminado de utilizar el sistema deberá cerrar sesión correctamente.

6.7. Especificación de Requerimientos

6.7.1. Casos de Uso y Actores

6.7.1.1. Lista de Actores

A001. Usuario Solicitante.- Es el usuario que da vida al proceso de desarrollo mediante la solicitud de una mejora o adecuación con la presentación de la especificación funcional, además de evaluar el requerimiento cuando ya se encuentre en producción.

A002. Gerente Sistemas.- Es quien aprueba las especificaciones funcionales solicitadas por los usuarios solicitantes, además de establecer una prioridad y una fecha de entrega si es aprobada la solicitud.

A003. Jefe Desarrollo.- Es el actor encargado de designar a un asistente de desarrollo para la ejecución de la solicitud aprobada.

A004. Asistente Desarrollo.- Es el encargado de desarrollar los cambios o mejoras indicadas en la especificación funcional en el ambiente de desarrollo, también es el encargado de generar la documentación del proyecto como son: Cronograma, Diseño Técnico, Plan de Pruebas y Manual de Implementación.

A005. Asistente QA.- Este actor es el encargado de realizar pruebas de calidad en el ambiente de pruebas a los proyectos hechos por el asistente de desarrollo siguiendo el Plan de Pruebas, adicionalmente es quien da la nueva versión al software.

A006. Asistente Operaciones.- Es el encargado de implementar en el ambiente de producción el requerimiento enviado por el asistente de QA.

6.7.1.2. Lista de Casos de Uso

UC001. Ingreso de Proyectos.- Es la actividad que realiza el Usuario Solicitante en la cual indica el tipo de proyecto, y anexa la especificación



funcional.

UC002. Aprobación de Proyectos.- Es la actividad que realiza el gerente de sistemas a las solicitudes enviadas por el usuario solicitante.

UC003. Asignación de Proyectos.- Es la actividad que realiza el jefe de desarrollo para la asignación de un desarrollador para la ejecución de la solicitud aprobada.

UC004. Desarrollo.- Es la actividad que realiza el asistente de desarrollo al requerimiento asignado lo que comprende la programación como tal y la generación de la documentación del proyecto.

UC005. Pruebas de QA.- Es la actividad que realiza el asistente de QA con la finalidad de garantizar la calidad de funcionamiento de los cambios realizados al sistema.

UC006. Versionamiento.- Es el proceso que realiza el asistente de QA, para dar una nueva versión al sistema de la compañía de acuerdo al tipo de proyecto que sea.

UC007. Pasó a Producción.- Es la actividad que realiza el asistente de operaciones, la cual consiste en seguir el Manual de Implementación.

UC008. Evaluación.- Es la actividad que realiza el Usuario Solicitante donde evalúa si el proyecto está de acuerdo a lo solicitado.

6.7.2. Requerimientos

6.7.2.1. Funcionales

RF.01. Para el ingreso de proyectos es necesario se adjunte la especificación funcional en formato PDF.

RF.02. Para la correcta administración y seguimiento de proyectos deben ser registrados en el sistema.

RF.03. Todos los usuarios deben estar registrados y asignado un perfil de uso.

RF.04. Debe tener autenticación de usuarios.

RF.05. Para el paso a pruebas de cada proyecto el asistente de desarrollo deberá registrar en el sistema la documentación así como también los componentes que fueron modificados.

RF.06. Para que un proyecto pueda ser pasado a producción este no deberá



tener observaciones de pruebas.

RF.07. Para que un proyecto pueda ser finalizado la evaluación por parte del usuario no deberá tener observaciones.

RF.08. El sistema deberá permitir el versionamiento automático de acuerdo al procedimiento de desarrollo de software.

RF.09. Deberá enviar notificaciones al email de cada paso que realice el proyecto.

RF.10. Permitirá generar e imprimir reportes.

6.7.2.2. No Funcionales

NRF.01. No requiere de programas adicionales para su funcionamiento a menos los posteriores por mencionar en requerimientos mínimos de software.

6.7.3. Requerimientos de Software

Cuadro 14: Requerimientos de Software

Descripción	Mínimo	Óptimos
Sistema Operativo	Windows Sever 2003 Windows Professional XP	Windows Server 2008 Windows 7
Base de datos	Oracle XE	Oracle 10g
Explorador de Internet	Mozilla Firefox Internet Explorer	Google Chrome

Fuente: Ricardo Morales

6.7.4. Requerimientos de Hardware

Cuadro 15: Requerimientos Hardware

Descripción		Mínimos	Óptimos
Servidor	Procesador	1.6 GHz CPU, 384 MB RAM	2.2 GHz o Superior
	Disco Duro	80 Gigas	500 Gigas 7200 RPM
	Memoria RAM	4 Gigas	8 Gigas
Pantalla		1024x600	1280x1024

Fuente: Ricardo Morales

6.7.5. Interfaces de Usuario

Login.- Esta ventana servirá para el ingreso al sistema en donde el usuario deberá colocar el nombre de usuario y contraseña proporcionados.



Cambio de Clave.- Esta ventana será utilizada para el cambio de clave y será ocupada cada que la clave caduque o sea la primera vez que el usuario ingresa al sistema.

Página de Inicio.- A esta página tendrán acceso todos los usuarios del sistema una vez hayan ingresado al sistema y desde la cual podrán acceder a las diferentes opciones según el perfil de usuario.

Mantenimiento de Empresa.- Desde esta ventana se podrán modificar los datos de la empresa así como la duración de la clave, y el lugar donde se guardaran los archivos de los proyectos que se registren a esta ventana solo tendrá acceso el usuario con perfil de administrador.

Mantenimiento de Área.- Esta ventana servirá para insertar, modificar, eliminar y buscar datos de las áreas que tiene la empresa de igual manera a esta ventana solo tendrá acceso el usuario con perfil de administrador.

Mantenimiento de Rol.- Dentro de esta página se podrá insertar, modificar, eliminar y buscar datos de los roles que va a manejar el sistema, a esta ventana solo tendrá acceso el usuario con perfil de administrador.

Mantenimiento de Usuario.- En esta página se podrá insertar, modificar, eliminar y buscar los datos de los usuarios del sistema a esta página solo tendrá acceso el usuario con perfil de administrador.

Mantenimiento de Paginas.- Esta ventana servirá para insertar, modificar, eliminar y buscar datos de las páginas del sistema, al igual que desde esta ventana se podrá dar permisos a las diferentes páginas de acuerdo a los perfiles de usuario, esta ventana solo tendrá acceso el usuario con perfil de administrador.

Mantenimiento Tipo Proyectos.- Dentro de esta ventana se podrá insertar, modificar, eliminar y buscar los tipos de proyectos que se manejan en el área de desarrollo, a esta ventana solo se podrá tener acceso con el perfil de administrador.

Mantenimiento Tipo Documentación.- Dentro de esta ventana se podrá insertar, modificar, eliminar y buscar los tipos de documentos que maneja el área de desarrollo para documentar sus proyectos, a esta ventana solo se podrá tener acceso con el perfil de administrador.

Mantenimiento Tipo Observaciones.- Esta venta servirá para ingresar, modificar, eliminar y buscar los tipos de observaciones que se manejan en el proceso de desarrollo, a esta ventana solo se podrá acceder con el perfil de administrador.



Mantenimiento Tipo Componentes.- Dentro de esta ventana se podrá insertar, modificar, eliminar y buscar los tipos de componentes con los que cuentan los sistemas que maneja la compañía, a esta venta solo se podrá ingresar con el perfil de administrador.

Mantenimiento Componentes.- En esta ventana se podrá buscar, insertar, modificar y eliminar registros de los componentes que manejan los sistemas de información de la compañía, solo se podrá acceder con el perfil de administrador.

Mantenimiento Prioridades.- Es esta ventana se podrá buscar, insertar, modificar y eliminar las prioridades con las que deben ser tratados cada proyecto ingresado al sistema, solo se podrá acceder con el perfil de usuario de administrador.

Mantenimiento Sistemas.- Dentro de esta ventana se podrá insertar, modificar, eliminar y buscar los diferentes sistemas de información con los que cuenta la compañía, a esta ventana solo se podrá acceder con el perfil de administrador.

Ingreso de Proyectos.- Dentro de esta ventana se podrá ingresar un nuevo proyecto adjuntando la especificación funcional en formato PDF, a esta ventana tendrá acceso el perfil de Usuario.

Aprobación de Proyectos.- Dentro de esta ventana se podrá aprobar o negar los proyectos ingresados por los usuarios solicitantes, a esta ventana tendrá acceso con el perfil de Gerencia.

Asignación de Proyectos.- Dentro de esta ventana se podrá asignar a los desarrolladores los proyectos que se encuentre aprobados, a esta ventana se podrá acceder con el perfil de Jefatura.

Desarrollo de Proyectos.- Dentro de esta ventana se podrán registrar los componentes, y la documentación de concerniente a cada proyecto asignado, a esta ventana solo se podrá acceder con el perfil de Desarrollador.

Pruebas de Proyectos.- En esta ventana se podrá ingresar los registros de las pruebas que se realicen a los proyectos así como la documentación que respalda el paso de pruebas o las observaciones, para ingresar a esta ventana se deberá tener el perfil de QA.

Pasos a Producción.- Una vez que el proyecto ha pasado las pruebas estará habilitado dentro de esta ventana para su paso a producción, a esta ventana solo se podrá acceder con el perfil de Operaciones.

Evaluación de Proyectos.- Una vez que el proyecto ha sido pasado a



producción dentro de esta ventana estará disponible los proyectos para que sean evaluados por el usuario solicitante para acceder a esta ventana se lo hará con el perfil de Usuario.

Reportes.- Dentro de esta ventana estarán disponibles los reportes estadísticos, y de información de proyectos, a esta ventana tendrán acceso todos los perfiles y se mostraran los reportes de acuerdo al perfil con el que se haya iniciado sesión.

6.7.6. Estándares Aplicados

6.7.6.1. Estándares Base de Datos

Los estándares aplicados para la base de datos son:

- Para la definición del nombre de cada campo de las tablas en la base de datos se ocupó la siguiente caracterización:

Tipo de Campo	Estándar	Ejemplo
Código	C	CPROYECTO
Fecha	F	FMODIFICACION
Estados	I	IESTADO
Montos	M	MCOSTO
Nombres	O	XNOMPROYECTO
Descripciones	X	

- Para el nombre de las tablas se ocupó la siguiente nomenclatura: Si la tabla pertenece a seguridad empezara con SE más un corto descriptivo de la tabla ejemplo: SEROL; si la tabla contiene información de proyectos empezara con PRO seguido de un descriptivo de la tabla ejemplo: PROCOMPONENTE

6.7.6.2. Estándares de Programación

Los estándares de programación aplicados son:

- Los diferentes objetos que contengan las paginas se los nombrara de la siguiente manera:

Objeto	Estándar	Ejemplo
TextBox	txt	txtClave
Button	btn	btnIngresar
CheckBox	chk	chkDocumentacion
RadioButtom	rdb	rdbGenero
ComboBox	cmb	cmbEstado



Para dar nombre a las páginas del sistema se usara la siguiente nomenclatura:

- Para las páginas de mantenimiento de tablas:

Se iniciara con el nombre de la tabla a la cual se realiza mantenimiento.

Luego seguirá el nombre de la acción que realiza la página.

Ejemplo:

seusuarioNuevo

- Para las páginas que ejecutan procesos

Se iniciara con el nombre del proceso, seguido de la acción que realiza.

Ejemplo:

proyectoIngresar

6.7.7. Diseño de Casos de Uso

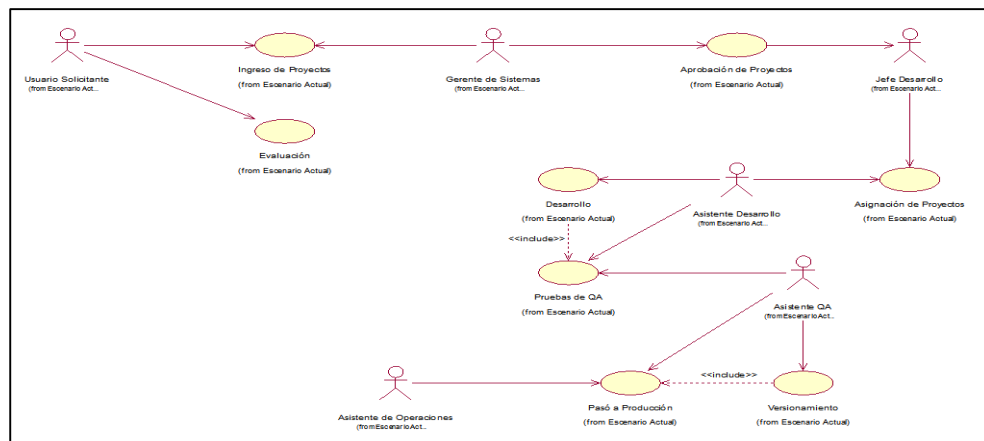


Ilustración 12: Diagrama de Casos de Uso

Fuente: Ricardo Morales

6.7.8. Especificación de Casos de Uso

Cuadro 16: Caso de Uso UC001

Caso de Uso: Ingreso de Proyectos
ID: UC001
Actores:



Usuario Solicitante
Precondición: El usuario debe tener una necesidad de mejora o adecuación a los sistemas. El usuario debe elaborar una especificación funcional.
Flujo de Eventos: El usuario deberá llenar un formato llamado Especificación Funcional para indicar cuales son los cambios se realicen en el sistema. Para el registro de proyectos la Especificación Funcional debe estar firmada por el usuario solicitante y su jefe inmediato.
Eventos Alternativos: Para casos extremos la Especificación Funcional puede ser receptada sin la firma del jefe inmediato.
Post-Condición: Una vez realizado el formato el usuario lo envía a la gerencia de sistemas para su aprobación.

Fuente: Ricardo Morales

Cuadro 17: Caso de Uso UC002

Caso de Uso: Aprobación de Proyectos
ID: UC002
Actores: Gerente de Sistemas
Precondición: Para la aprobación de un proyecto este debe haber sido enviado por el usuario solicitante a la gerencia de sistemas.
Flujo de Eventos: El gerente de sistemas deberá aprobar o negar los proyectos de acuerdo a las prioridades de la empresa. Si es aprobado deberá asignar una prioridad y un tiempo de entrega al proyecto.
Eventos Alternativos: En caso de no ser aprobado el proyecto se debe notificar al usuario por qué no se aprueba.
Post-Condición: Una vez aprobado el proyecto este es enviado al área de desarrollo para su ejecución.

Fuente: Ricardo Morales

Cuadro 18: Caso de Uso UC003

Caso de Uso: Asignación de Proyectos
ID: UC003
Actores: Jefe de Desarrollo
Precondición: Para la asignación de proyectos estos deben estar aprobados por la



gerencia de sistemas.
Flujo de Eventos: El jefe de desarrollo asignara los proyectos a los asistentes de desarrollo de acuerdo a la carga laboral de cada uno.
Eventos Alternativos: En caso de que no estuviera el jefe de desarrollo los asistentes se repartirán los proyectos de acuerdo a la carga laboral.
Post-Condición: Una vez asignado el proyecto se le entrega la Especificación Funcional al asistente de desarrollo para su ejecución.

Fuente: Ricardo Morales

Cuadro 19: Caso de Uso UC004

Caso de Uso: Desarrollo
ID: UC004
Actores: Asistente de Desarrollo, Usuario Solicitante, Asistente de QA
Precondición: Para esta está el asistente de desarrollo deberá tener asignado el proyecto, y la especificación funcional aprobada por la gerencia de sistemas.
Flujo de Eventos: Analiza el requerimiento. Realiza cronograma y notificar al usuario solicitante. Desarrolla y modifica los componentes o archivos fuentes. Realiza la documentación del proyecto. (Manual de Implementación, Plan de Pruebas, Diseño Técnico) Realiza pruebas con el usuario solicitante en el Ambiente de Desarrollo. Sociabiliza Plan de Pruebas con asistente de QA.
Eventos Alternativos: En casos en los que los procesos no se vean muy afectados las pruebas con el usuario en el ambiente de desarrollo se podrán omitir.
Post-Condición: Una vez concluido el desarrollo el asistente de desarrollo deberá solicitar que se realicen pruebas del requerimiento al asistente de QA.

Fuente: Ricardo Morales

Cuadro 20: Caso de Uso UC005

Caso de Uso: Pruebas de QA
ID: UC005
Actores: Asistente de QA Asistente de Desarrollo



Precondición: Para que el proyecto pueda ser probado este deberá tener toda la documentación y los componentes modificados.
Flujo de Eventos: Recibe los componentes y documentación de proyecto. Implementa los cambios en el ambiente de pruebas siguiendo el Manual de Implementación. Sigue las pruebas detalladas en el Plan de Pruebas Elabora el certificado de calidad avalando el funcionamiento correcto de las modificaciones al sistema.
Eventos Alternativos: En caso de que el proyecto no cumpla con las pruebas enviadas el asistente de desarrollo deberá emitir un archivo en el cual se detallen las observaciones y reenviarlo al asistente de desarrollo.
Post-Condición: Una vez que el proyecto haya aprobado las pruebas deberá ser versionado.

Fuente: Ricardo Morales

Cuadro 21: Caso de Uso UC006

Caso de Uso: Versionamiento
ID: UC006
Actores: Asistente de QA
Precondición: El proyecto a versionar debe haber aprobado las pruebas indicadas en el Plan de Pruebas.
Flujo de Eventos: El asistente QA versiona de acuerdo al tipo de proyecto. Registra los componentes en el archivo de versionamiento
Eventos Alternativos: No existen eventos alternativos.
Post-Condición: Una vez el proyecto se haya versionado el asistente de QA solicita se implemente en el ambiente de producción.

Fuente: Ricardo Morales

Cuadro 22: Caso de Uso UC007

Caso de Uso: Paso a Producción
ID: UC007
Actores: Asistente de Operaciones, Asistente de QA, Asistente de Desarrollo
Precondición: Para poder implementar en el ambiente de producción el proyecto debe



tener asignada una nueva versión.
Flujo de Eventos: El asistente de operaciones recibe el proyecto con la nueva versión. Implementa en el ambiente de producción siguiendo el Manual de Implementación.
Eventos Alternativos: Si por alguna razón el proyecto no se puede implementar en el ambiente de producción el asistente de operaciones debe notificar al asistente de QA y al de desarrollo.
Post-Condición: Una vez implementado debe notificar a los asistentes de QA y Desarrollo.

Fuente: Ricardo Morales

Cuadro 23: Caso de Uso UC008

Caso de Uso: Evaluación
ID: UC008
Actores: Usuario Solicitante Asistente Desarrollo
Precondición: El proyecto debe estar implementado en producción.
Flujo de Eventos: El asistente de desarrollo notifica al usuario de las nuevas modificaciones en el sistema y entrega el Registro de Evaluación. El usuario solicitante prueba las modificaciones. El usuario llena el formulario Registro de Evaluación indicando que todo funciona correctamente.
Eventos Alternativos: En caso de que existieran errores en las pruebas en producción el usuario deberá llenar el Registro de Evaluación y notificar al asistente de desarrollo.
Post-Condición: Si el proyecto no tiene observaciones es finalizado.

Fuente: Ricardo Morales

6.8. Análisis y Diseño de la Propuesta

6.8.1. Arquitectura Física

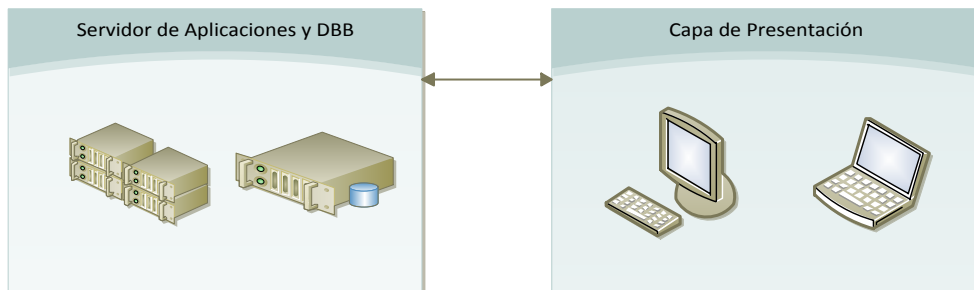


Ilustración 13: Diagrama Arquitectura Física
Fuente: Ricardo Morales

En el diagrama anterior se puede apreciar la arquitectura física de la manera cómo funcionara el sistema, y se describe cada una de sus partes a continuación.

6.8.1.1. Capa Presentación

En la capa de presentación se le mostrara al usuario la interface del sistema desde la cual podrá hacer uso del sistema, para acceder al sistema el usuario deberá tener una maquina con cualquier navegador el cual establecerá la conexión con el servidor de aplicaciones y DBB.

6.8.1.2. Capa del Servidor de Aplicación y DBB

En esta capa del servidor de aplicaciones y DBB responde a las peticiones de los clientes, maneja las transacciones según la lógica del sistema y se conecta con la DBB que se encuentra dentro del mismo servidor atreves de consultas SQL.

La DBB que se encuentra dentro de este servidor es la Oracle 10G XE, y es la encargada de almacenar toda la información del sistema.

6.8.2. Arquitectura Lógica

6.8.2.1. Capa de Datos

La constitución del acceso a datos está formada por la instancia a un motor de bases de datos, el cual realiza todo el almacenamiento de la información enviada, recibiendo solicitudes de almacenamiento o recuperación de información desde la capa de lógica de negocio.

Serán todas las consultas en lenguaje SQL para realizar la conexión con la base de datos y gestionar las mismas en las tablas, estas consultas estarán

codificadas dentro de clases en una biblioteca de clases solo para acceder a la información solicitada.

6.8.2.2. Capa de Lógica del Negocio

Esta biblioteca de clases es donde se encuentran los procesos que se ejecutan así como también todas las validaciones requeridas por el sistema, esta capa recibe las peticiones del usuario y envía las respuestas tras el proceso ejecutado.

Es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes que realiza el usuario y presentar los resultados con la capa de datos. En esta capa solo se trabaja con la ejecución de las reglas del negocio.

6.8.2.3. Capa de Presentación

Se puede llamar también la interface de usuario, a través de esta capa el usuario puede acceder a la información del sistema, esta capa es la encargada de capturar la información ingresada por el usuario para enviarla a la capa de lógica del negocio.

Esta capa se comunica únicamente con la capa de lógica de negocio y está organizada de acuerdo a los requerimientos del usuario.

6.8.3. Modelo de Análisis

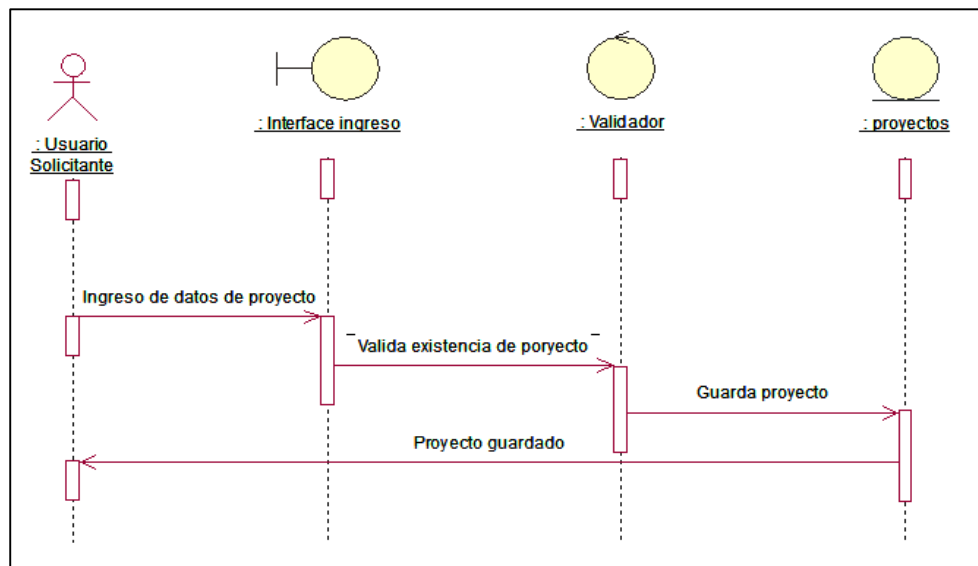


Ilustración 14: Diagrama de Secuencia UC001

Fuente: Ricardo Morales

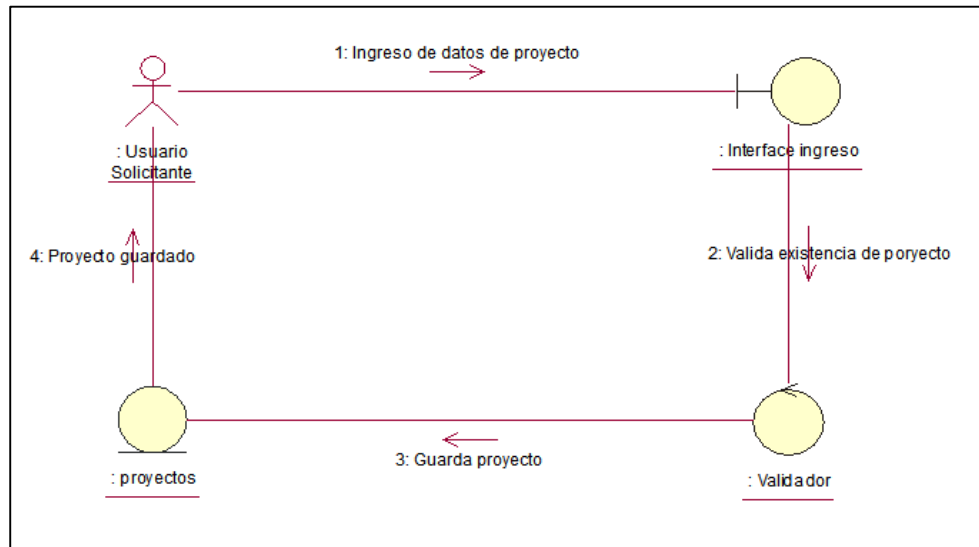


Ilustración 15: Diagrama Colaboración UC0010
Fuente: Ricardo Morales

6.8.4. Modelo de Diseño

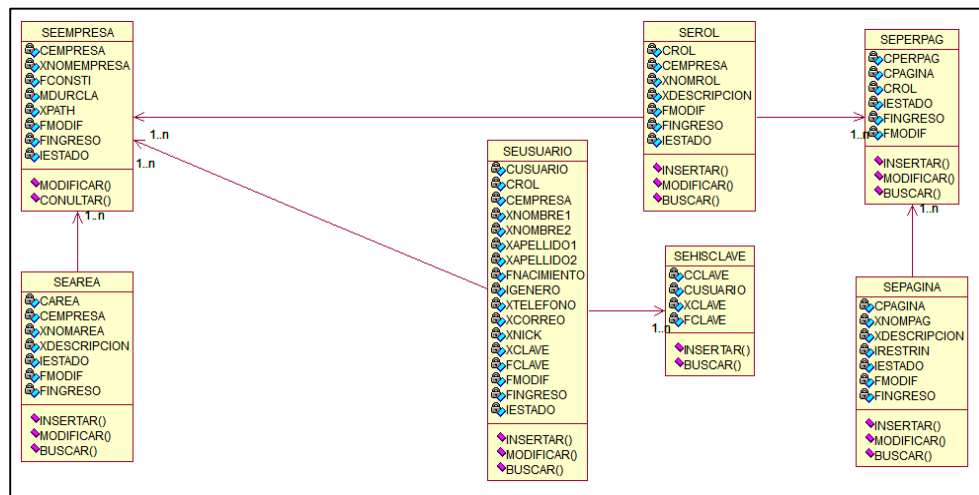


Ilustración 16: Diagrama de clases
Fuente: Ricardo Morales

6.8.5. Modelo de Implementación

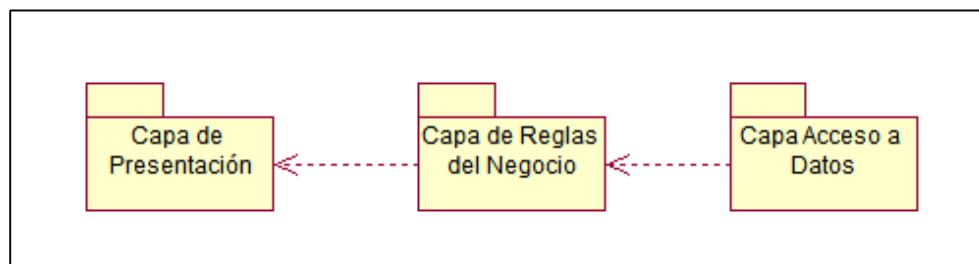


Ilustración 17: Diagrama de Implementación
Fuente: Ricardo Morales

6.8.6. Modelo Lógico

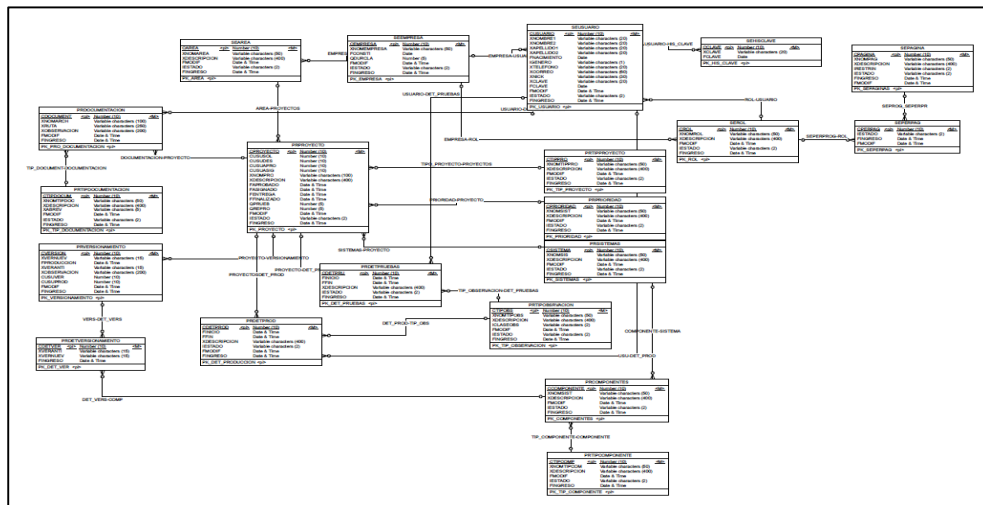


Ilustración 18: Diagrama Modelo Lógico

Fuente: Ricardo Morales

6.8.7. Modelo Físico

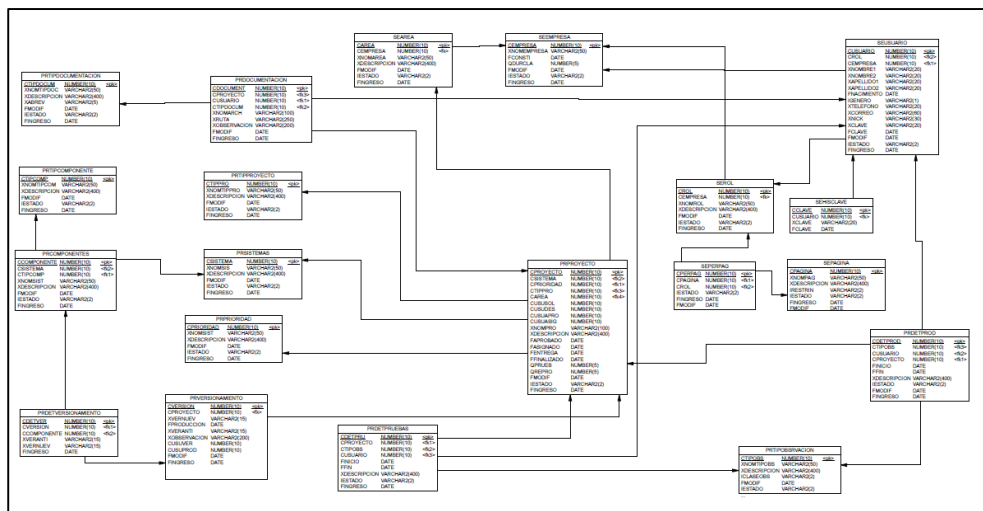


Ilustración 19: Diagrama Modelo Físico

Fuente: Ricardo Morales



6.8.8. Diseño de Interfaces

6.8.8.1. Login

Ilustración 20: Login - Ricardo Morales

Fuente: Ricardo Morales

Usuario: Caja de texto para el ingreso del nombre de usuario.

Password: Caja de texto para el ingreso de la contraseña.

Ingresar: Botón para la validación de datos e ingreso al sistema.

6.8.8.2. Cambio de Clave

Ilustración 21: Cambio de Clave

Fuente: Ricardo Morales

Clave Anterior: Caja de texto para ingresar la contraseña anterior.

Clave Nueva: Caja de texto para el ingreso de la nueva contraseña.

Confirmar Clave: Caja de texto para confirmación de la nueva contraseña.



Cambiar: Botón para la validación de contraseñas y guardar cambio.

Cancelar: Botón para anular la acción y volver a la página del Login.

6.8.8.3. Página de Inicio o Menú Principal

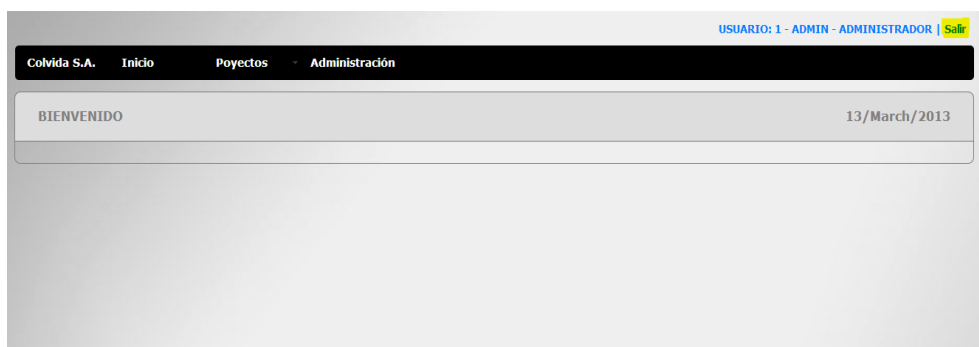


Ilustración 22: Página de Inicio

Fuente: Ricardo Morales

Menú: Dentro de este menú se encontraran las opciones para acceder a las diferentes páginas del menú.

Salir: Se usara este enlace para salir del sistema.

6.8.8.4. Mantenimiento Empresa

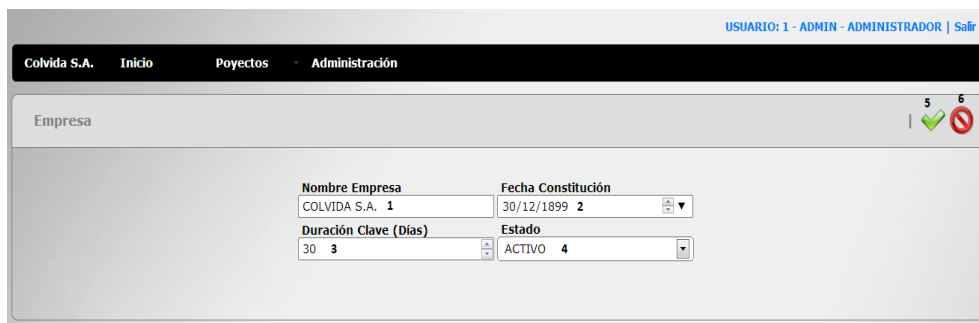


Ilustración 23: Mantenimiento de Empresa

Fuente: Ricardo Morales

1. **Nombre Empresa:** Caja de texto para el nombre de la empresa.
2. **Fecha Constitución:** Calendario para establecer fecha de constitución de la empresa.
3. **Duración Clave:** Caja de texto para establecer la duración de la contraseña.
4. **Estado:** Combo box para establecer el estado de la empresa.
5. **Guardar:** Botón para guardar los cambios que se realicen.
6. **Cancelar:** Botón para cancelar los cambios y volver al menú principal.



6.8.8.5. Ingreso de Proyectos

Ilustración 24: Ingreso de Clientes

Fuente: Ricardo Morales

1. **Nombre Proyecto:** Caja de texto para el ingreso del nombre del proyecto.
2. **Descripción de Proyecto:** Caja de texto para el ingreso de la descripción del proyecto.
3. **Área Solicitante:** Combo box para la selección del área que solicita el proyecto.
4. **Tipo de Proyecto:** Combo box para la selección del tipo proyecto que es.
5. **Especificación Funcional:** Botón para agregar la especificación funcional en formato PDF.
6. **Guardar:** Botón para guardar el proyecto y enviar para su aprobación.
7. **Cancelar:** Botón para cancelar el ingreso de proyecto y volver a la página de inicio.

6.9. Plan Maestro de Pruebas

Es muy importante llegar hasta este punto en el cual se definirá la calidad del producto que estamos entregando en el ambiente de verificación y acciones que ejecutara cuando realiza alguna solicitud el cliente.

Las pruebas que se realizaron al sistema son las siguientes:

- **Seguridad:**
 - Dentro de estas pruebas se realizó la validación de contraseñas en la ventana de Login, se ingresó tres veces erróneo el password lo cual hizo que se el usuario sea bloqueado.
 - Se realizó pruebas para accesos de acuerdo a perfiles de usuario.
 - Se probó que la contraseña caduque cada 30 días.
 - Se realizó pruebas de cambio de contraseña para comprobar que el sistema no admita contraseñas antes registradas.
- **Navegación:**
 - Se realizaron pruebas de navegación con el fin de que todas las opciones del menú tengan un direccionamiento correcto.



- **Interface de Usuarios**
 - Se probó que la interface sea amigable y de fácil entendimiento para el usuario de la aplicación.
 - Se verifico que la interface sea atractiva al usuario.
- **Manejo de bases de datos**
 - Para efectos de estas pruebas se realizó el mantenimiento de tablas ocupando la interfaz gráfica del sistema.
- **Envío de mails**
 - Para el efecto de esta prueba se ingresaron proyectos lo cual arrojó buenos resultados ya que en los buzones de correo si se recibieron los mails enviados por el sistema.

6.10. Diseños de Casos de Prueba

Al referirnos al caso de pruebas hacemos referencia a la información que necesita el software para interactuar con el usuario y además de controlar y validar dicha información, todos los datos ingresados correctamente son importantes, y debemos tomar en cuenta también que el usuario no ingrese información errónea para no almacenar en la base de datos información basura, ver Anexo 04.

6.11. Manuales y Entregables

Al culminar el desarrollo del proyecto se entregara:

- Script para la creación de la base de datos el cual se describe en el Anexo 05.
- Script para carga de datos iniciales en base de datos
- Códigos fuente del sistema desarrollado
- Manual de Implementación el cual se detalla en el Anexo 06.
- Manual de Usuario el mismo que se detalla en el Anexo 07.
- Manual Técnico el mismo que se detalla en el Anexo 08.



6.12. Capacitación a Usuarios

Para la capacitación a los usuarios se seguirá el cronograma que se encuentra detallado en el Anexo 09, y se capacitara sobre los siguientes temas:

- Registro en Sistema
- Administración del Sistema
- Ingreso de Proyectos
- Aprobación de Proyectos
- Asignación de Proyectos
- Desarrollo de Proyectos
- Pruebas de Proyectos
- Paso a Producción
- Evaluación de Proyectos
- Generación de Reportes

Dentro de los principales recursos que se ocuparan serán: infocus, maquina con acceso al sistema, maquina con diapositivas de procesos.



Instituto Tecnológico Superior "CORDILLERA"



ANEXOS

Anexo 01: Bibliografía Básica

- HEREDIA Hugo Ing. – Investigación Científica, Apuntes y conocimientos obtenidos en clase.
- Prof.: Francisco Leiva Zea – Investigación Científica – Nociones y Metodología
- ALEGRIA, Rodrigo Altamirano, Edwin – Investigación Científica

Anexo 02: Net Grafía

- Ing. USBMed, REVISIÓN DE LOS SISTEMAS DE CONTROL DE VERSIONES UTILIZADOS EN EL DESARROLLO DE SOFTWARE de <http://web.usbmed.edu.co/usbmed/fing/v3n1/v3n1a9.pdf>
- Version de software, http://es.wikipedia.org/wiki/Versi%C3%B3n_de_software
- Control de versiones de software, <http://www.cristalab.com/blog/control-de-versiones-de-software-c82353/>

Anexo 03: Encuesta

Encuesta

La siguiente encuesta está diseñada con el fin de mejorar la administración de proyectos y el proceso de versionamiento de software de los requerimientos presentados al área de desarrollo de la compañía de seguros de vida COLVIDA, por lo cual solicito a usted conteste las siguientes preguntas:



1. ¿Al momento el área de desarrollo de software utiliza herramientas informáticas para realizar los procesos de administración y versionamiento de proyectos?

Si	
No	
No sabe/No opina	

2. ¿Qué tipo de herramienta informática utiliza el área de desarrollo para los procesos de administración y versionamiento de proyectos?

Ofimática	
Sistema de Información	
No sabe/No opina	

3. ¿Cree que las herramientas informáticas a implementarse mejoraran los procesos de administración y versionamiento de proyectos?

Si	
No	
No sabe/No opina	

4. ¿Cómo evalúa el proceso que lleva el área de desarrollo de software para la administración y versionamiento de proyectos?

Excelente	
Bueno	
Malo	
No sabe/No opina	

5. ¿Considera que el área de desarrollo deba implementar un software que permita administrar de forma digital la documentación de cada proyecto?

Si	
No	
No sabe/No opina	



6. ¿Considera que el área de desarrollo deba implementar un software que permita versionar automáticamente cada proyecto al igual que permita llevar un histórico de versiones?

Si	
No	
No sabe/No opina	

7. ¿El nivel de información que recibe sobre el estado de su proyecto es?

Excelente	
Bueno	
Malo	
No sabe/No opina	

8. ¿Cree usted que el área de desarrollo deba incorporar un sistema de información para conocer el estado de cada uno de los proyectos presentados?

Si	
No	
No sabe/No opina	

9. ¿El área de desarrollo requiere un software que permita la administración y versionamiento de proyectos?

Si	
No	
No sabe/No opina	

10. ¿En qué nivel cree usted que mejorara los procesos de administración y versionamiento de proyectos del área de desarrollo con la implementación de un software?

Excelente	
Bueno	
Malo	
No sabe/No opina	



Anexo 04: Diseño de Casos de Prueba

Nombre:	Iniciar Sesión	
Conclusión:	El Usuario de estar registrado y tener un nombre de usuario y contraseña para iniciar su sesión en el sistema.	
Entrada	<ul style="list-style-type: none"> • Nombre de Usuario • Contraseña de Seguridad 	
Salida	Validación Sesión Caducidad de contraseña 30 días	Mensaje Ingreso de Datos

Nombre:	Página Principal	
Condición:	Cuando el usuario ya ingresa correctamente, se habilitaran los permisos del programa de acuerdo al perfil del usuario.	
Entrada	<ul style="list-style-type: none"> • Nombre de Usuario • Fecha • Hora 	
Salida	Guarda Nombre de usuario	

Nombre:	Mantenimientos de Tablas	
Condición:	Para ingresar nuevos registros los datos con asteriscos son obligatorios.	
Entrada	Los datos según el formulario varían	
Salida	Guarda datos del formulario.	Error de validación.

Nombre:	Ingreso Proyectos	
Condición:	Ingresar los datos de proyecto y adjuntar la especificación funcional.	
Entrada	<ul style="list-style-type: none"> • Datos de proyecto • Especificación Funcional 	
Salida	Guarda datos de proyecto.	

Nombre:	Aprobación de Proyectos	
Condición:	Ingresar los datos para la aprobación de proyectos validación de prioridad y fecha de finalización.	
Entrada	<ul style="list-style-type: none"> • Datos de aprobación de proyecto 	
Salida	Si la fecha es mayor a la fecha actual aprueba el proyecto.	



Nombre:	Asignación de Proyectos
Condición:	Los asistentes de desarrollo deben estar activos para su asignación.
Entrada	<ul style="list-style-type: none"> Asistente de Desarrollo
Salida	Guarda datos de proyecto.

Nombre:	Desarrollo de Proyectos
Condición:	El asistente de desarrollo debe ingresar toda la documentación de los proyectos (Diseño Técnico, Manual de Implementación, Plan de Pruebas, Cronograma)
Entrada	<ul style="list-style-type: none"> Documentación de proyecto
Salida	Si cumple con toda la documentación guarda el proyecto y pasa a pruebas.

Nombre:	Pruebas de Proyectos
Condición:	El asistente de calidad debe realizar las pruebas al proyecto y subir el certificado de calidad u observaciones.
Entrada	<ul style="list-style-type: none"> Certificado de calidad u Observaciones
Salida	Si el archivo subido es el certificado de calidad versiona el proyecto y lo pasa a producción. Si tiene observaciones lo reenvía al desarrollador.

Nombre:	Paso a Producción
Condición:	El asistente de operaciones implementara el proyecto en producción.
Entrada	<ul style="list-style-type: none"> Datos de implementación.
Salida	Si el proyecto no tuvo problemas en la implementación se notifica al usuario y se pasa a evaluación por el usuario. Si el proyecto tuvo problemas se notifica al desarrollador.

Nombre:	Evaluación de Proyecto
Condición:	El proyecto debe estar implementado en producción y el usuario solicitante debe estar notificado del cambio en el sistema.
Entrada	<ul style="list-style-type: none"> Calificación del proyecto.
Salida	Si el usuario lo califica sin observaciones se finaliza el proyecto, caso contrario se notifica al desarrollador.



Anexo 05: Script Base de Datos

```
/*=====
===*/

/* DBMS name:   ORACLE Version 10g           */

/* Created on:   10/04/2013 23:58:29           */

/*=====
===*/

/*=====
===*/

/* Table: PRCOMPONENTE                        */

/*=====
===*/

create table PRCOMPONENTE (

    CCOMPONENTE      NUMBER(10)              not null,

    CSISTEMA          NUMBER(10),

    CTIPCOMP          NUMBER(10),

    XNOMBRE           VARCHAR2(50),

    XDESCRIPCION      VARCHAR2(400),

    FMODIF            DATE,

    IESTADO           VARCHAR2(2),

    FINGRESO          DATE,

    XVERANT           VARCHAR2(15),

    XVERNUE           VARCHAR2(15),

    CPROYECT          NUMBER(10),

    constraint PK_PRCOMPONENTE primary key (CCOMPONENTE)

);
```



```
/*=====
===*/
```

```
/* Index: COMPONENTE_SISTEMA_FK */
```

```
/*=====
===*/
```

```
create index COMPONENTE_SISTEMA_FK on PRCOMPONENTE (
```

```
    CSISTEMA ASC
```

```
);
```

```
/*=====
===*/
```

```
/* Index: TIP_COMPONENTE_COMPONENTE_FK */
```

```
/*=====
===*/
```

```
create index TIP_COMPONENTE_COMPONENTE_FK on PRCOMPONENTE (
```

```
    CTIPCOMP ASC
```

```
);
```

```
/*=====
===*/
```

```
/* Table: PRDETCOMP */
```

```
/*=====
===*/
```

```
create table PRDETCOMP (
```

```
    CDETCOM      NUMBER(10)      not null,
```

```
    CPROYECTO    NUMBER(10),
```

```
    CCOMPONENTE  NUMBER(10),
```

```
    XVERANT       VARCHAR2(15),
```

```
    XVERNUE       VARCHAR2(15),
```



```
FINGRESO      DATE,

constraint PK_PRDETCOMP primary key (CDETCOM)

);

/*=====
===*/

/* Index: PROYECTO_DETCOMP_FK                      */

/*=====
===*/

create index PROYECTO_DETCOMP_FK on PRDETCOMP (

    CPROYECTO ASC

);

/*=====
===*/

/* Index: COMP_DETCOMP_FK                          */

/*=====
===*/

create index COMP_DETCOMP_FK on PRDETCOMP (

    CCOMPONENTE ASC

);

/*=====
===*/

/* Table: PRDETDOCUM                                */

/*=====
===*/

create table PRDETDOCUM (

    CDOCUMENT      NUMBER(10)          not null,

    CPROYECTO      NUMBER(10),
```



```
CUSUARIO      NUMBER(10),
CTIPDOCUM     NUMBER(10),
XNOMARCH      VARCHAR2(100),
XRUTA         VARCHAR2(250),
XOBSERVACION  VARCHAR2(200),
FMODIF        DATE,
FINGRESO      DATE,

constraint PK_PRDETDOCUM primary key (CDOCUMENT)

);

/*=====
===*/

/* Index: DOCUMENTACION_PROYECTO_FK          */

/*=====
===*/

create index DOCUMENTACION_PROYECTO_FK on PRDETDOCUM (

    CPROYECTO ASC

);

/*=====
===*/

/* Index: USUARIO_DOCUMENTACION_FK          */

/*=====
===*/

create index USUARIO_DOCUMENTACION_FK on PRDETDOCUM (

    CUSUARIO ASC

);

/*=====
===*/
```



```
/* Index: TIP_DOCUMENT_DOCUMENTACION_FK */

/*=====
===*/

create index TIP_DOCUMENT_DOCUMENTACION_FK on PRDETDOCUM (
    CTIPDOCUM ASC
);

/*=====
===*/

/* Table: PRDETPROD */

/*=====
===*/

create table PRDETPROD (
    CDETPROD      NUMBER(10)      not null,
    CTIPOBS       NUMBER(10),
    CUSUARIO      NUMBER(10),
    CPROYECTO    NUMBER(10),
    FINICIO       DATE,
    FFIN          DATE,
    CUSUPROD      NUMBER(10),
    XDESCRIPCION  VARCHAR2(400),
    IESTADO       VARCHAR2(2),
    FMODIF        DATE,
    FINGRESO      DATE,
    constraint PK_PRDETPROD primary key (CDETPROD)
);

/*=====
```



```
===*/

/* Index: PROYECTOSDET_PROD_FK                                */

/*=====
===*/

create index PROYECTOSDET_PROD_FK on PRDETPROD (

    CPROYECTO ASC

);

/*=====
===*/

/* Index: USU_DET_PROD_FK                                      */

/*=====
===*/

create index USU_DET_PROD_FK on PRDETPROD (

    CUSUARIO ASC

);

/*=====
===*/

/* Index: DET_PROD_TIP_OBS_FK                                  */

/*=====
===*/

create index DET_PROD_TIP_OBS_FK on PRDETPROD (

    CTIPOBS ASC

);

/*=====
===*/

/* Table: PRDETPRUEBAS                                         */

/*=====
===*/
```




```
create table PRDETPRUEBAS (  
    CDETPRU      NUMBER(10)          not null,  
    CPROYECTO    NUMBER(10),  
    CTIPOBS       NUMBER(10),  
    CUSUARIO      NUMBER(10),  
    FINICIO       DATE,  
    FFIN          DATE,  
    CUSUPRU       NUMBER(10),  
    XDESCRIPCION  VARCHAR2(400),  
    IESTADO       VARCHAR2(2),  
    FINGRESO      DATE,  
  
    constraint PK_PRDETPRUEBAS primary key (CDETPRU)  
  
);  
  
/*=====   
===*/  
  
/* Index: PROYECTO_DET_PRUE_FK */  
  
/*=====   
===*/  
  
create index PROYECTO_DET_PRUE_FK on PRDETPRUEBAS (  
    CPROYECTO ASC  
  
);  
  
/*=====   
===*/  
  
/* Index: TIP_OBSERVACION_DET_PRUEBAS_FK */  
  
/*=====   
===*/
```



```
create index TIP_OBSERVACION_DET_PRUEBAS_FK on PRDETPRUEBAS (
    CTIPOBS ASC
);

/*=====
===*/

/* Index: USUARIO_DET_PRUEBAS_FK */

/*=====
===*/

create index USUARIO_DET_PRUEBAS_FK on PRDETPRUEBAS (
    CUSUARIO ASC
);

/*=====
===*/

/* Table: PRPRIORIDAD */

/*=====
===*/

create table PRPRIORIDAD (
    CPRIORIDAD    NUMBER(10)        not null,
    XNOMBRE       VARCHAR2(50),
    XDESCRIPCION  VARCHAR2(400),
    FMODIF        DATE,
    IESTADO       VARCHAR2(2),
    FINGRESO      DATE,
    constraint PK_PRPRIORIDAD primary key (CPRIORIDAD)
);

/*=====
===*/
```



/* Table: PRPROYECTO

*/

/*=====

===*/

create table PRPROYECTO (

CPROYECTO NUMBER(10) not null,

CSISTEMA NUMBER(10),

CPRIORIDAD NUMBER(10),

CTIPPRO NUMBER(10),

CAREA NUMBER(10),

CUSUSOL NUMBER(10),

CUSUDES NUMBER(10),

CUSUAPRO NUMBER(10),

CUSUASIG NUMBER(10),

XNOMPRO VARCHAR2(100),

XDESCRIPCION VARCHAR2(400),

FAPROBADO DATE,

FASIGNADO DATE,

FENTREGA DATE,

FFINALIZADO DATE,

QPRUEB NUMBER(5),

QREPRO NUMBER(5),

FMODIF DATE,

IESTADO VARCHAR2(2),

FINGRESO DATE,

constraint PK_PRPROYECTO primary key (CPROYECTO)



```
);

/*=====
===*/

/* Index: PRIORIDAD_PROYECTO_FK */

/*=====
===*/

create index PRIORIDAD_PROYECTO_FK on PRPROYECTO (

    CPRIORIDAD ASC

);

/*=====
===*/

/* Index: SISTEMAS_PROYECTO_FK */

/*=====
===*/

create index SISTEMAS_PROYECTO_FK on PRPROYECTO (

    CSISTEMA ASC

);

/*=====
===*/

/* Index: TIPO_PROYECTO_PROYECTOS_FK */

/*=====
===*/

create index TIPO_PROYECTO_PROYECTOS_FK on PRPROYECTO (

    CTIPPRO ASC

);

/*=====
===*/

/* Index: AREA_PROYECTOS_FK */
```



```
/*=====
===*/
```

```
create index AREA_PROYECTOS_FK on PRPROYECTO (
```

```
    CAREA ASC
```

```
);
```

```
/*=====
===*/
```

```
/* Table: PRSISTEMA                                */
```

```
/*=====
===*/
```

```
create table PRSISTEMA (
```

```
    CSISTEMA      NUMBER(10)          not null,
```

```
    XNOMSIS       VARCHAR2(50),
```

```
    XDESCRIPCION  VARCHAR2(400),
```

```
    FMODIF        DATE,
```

```
    IESTADO       VARCHAR2(2),
```

```
    FINGRESO      DATE,
```

```
    constraint PK_PRSISTEMA primary key (CSISTEMA)
```

```
);
```

```
/*=====
===*/
```

```
/* Table: PRTIPCOMPONENTE                          */
```

```
/*=====
===*/
```

```
create table PRTIPCOMPONENTE (
```

```
    CTIPCOMP      NUMBER(10)          not null,
```

```
    XNOMTIPCOM    VARCHAR2(50),
```



```
XDESCRIPCION    VARCHAR2(400),

FMODIF         DATE,

IESTADO        VARCHAR2(2),

FINGRESO       DATE,

constraint PK_PRTIPCOMPONENTE primary key (CTIPCOMP)

);

/*=====
===*/

/* Table: PRTIPDOCUMENTACION */

/*=====
===*/

create table PRTIPDOCUMENTACION (

    CTIPDOCUM    NUMBER(10)          not null,

    XNOMTIPDOC   VARCHAR2(50),

    XDESCRIPCION VARCHAR2(400),

    XABREV       VARCHAR2(5),

    CUSUDOC     NUMBER(10),

    FMODIF      DATE,

    IESTADO     VARCHAR2(2),

    FINGRESO    DATE,

    constraint PK_PRTIPDOCUMENTACION primary key (CTIPDOCUM)

);

/*=====
===*/

/* Table: PRTIPOBSERVACION */

/*=====
```



===*/

```
create table PRTIPOBSERVACION (  
    CTIPOBS      NUMBER(10)          not null,  
    XNOMTIPOBS   VARCHAR2(50),  
    XDESCRIPCION VARCHAR2(400),  
    ICLASEOBS    VARCHAR2(2),  
    FMODIF       DATE,  
    IESTADO      VARCHAR2(2),  
    FINGRESO     DATE,  
    constraint PK_PRTIPOBSERVACION primary key (CTIPOBS)  
);
```

```
/*=====
```

```
/* Table: PRTIPPROYECTO */
```

```
/*=====
```

```
create table PRTIPPROYECTO (  
    CTIPPRO      NUMBER(10)          not null,  
    XNOMTIPPRO   VARCHAR2(50),  
    XDESCRIPCION VARCHAR2(400),  
    FMODIF       DATE,  
    IESTADO      VARCHAR2(2),  
    FINGRESO     DATE,  
    constraint PK_PRTIPPROYECTO primary key (CTIPPRO)  
);
```



```
/*=====
===*/
```

```
/* Table: PRVERSIONAMIENTO */
```

```
/*=====
===*/
```

```
create table PRVERSIONAMIENTO (
    CVERSION      NUMBER(10)          not null,
    CPROYECTO     NUMBER(10),
    XVERNUEV      VARCHAR2(15),
    FPRODUCCION   DATE,
    XVERANTI      VARCHAR2(15),
    XOBSERVACION  VARCHAR2(200),
    CUSUVER       NUMBER(10),
    CUSUPROD      NUMBER(10),
    FMODIF        DATE,
    FINGRESO      DATE,
    QVX           NUMBER(5),
    QVY           NUMBER(5),
    QVZ           NUMBER(5),
    constraint PK_PRVERSIONAMIENTO primary key (CVERSION)
);
```

```
/*=====
===*/
```

```
/* Index: PROYECTO_VERSIONAMIENTO_FK */
```

```
/*=====
===*/
```




```
create index PROYECTO_VERSIONAMIENTO_FK on PRVERSIONAMIENTO (
    CPROYECTO ASC
```

```
);
```

```
/*=====
===*/
```

```
/* Table: SEAREA */
```

```
/*=====
===*/
```

```
create table SEAREA (
```

```
    CAREA          NUMBER(10)          not null,
```

```
    CEMPRESA       NUMBER(10),
```

```
    XNOMAREA       VARCHAR2(50),
```

```
    XDESCRIPCION   VARCHAR2(400),
```

```
    FMODIF         DATE,
```

```
    IESTADO        VARCHAR2(2),
```

```
    FINGRESO       DATE,
```

```
    constraint PK_SEAREA primary key (CAREA)
```

```
);
```

```
/*=====
===*/
```

```
/* Index: EMPRESA_AREA_FK */
```

```
/*=====
===*/
```

```
create index EMPRESA_AREA_FK on SEAREA (
```

```
    CEMPRESA ASC
```

```
);
```



```
/*=====
===*/
```

```
/* Table: SEEMPRESA */
```

```
/*=====
===*/
```

```
create table SEEMPRESA (
    CEMPRESA      NUMBER(10)      not null,
    XNOMEMPRESA   VARCHAR2(50),
    FCONSTI       DATE,
    QDURCLA       NUMBER(5),
    XUBIARCH      VARCHAR2(300),
    FMODIF        DATE,
    IESTADO       VARCHAR2(2),
    FINGRESO      DATE,
    constraint PK_SEEMPRESA primary key (CEMPRESA)
);
```

```
/*=====
===*/
```

```
/* Table: SEHISCLAVE */
```

```
/*=====
===*/
```

```
create table SEHISCLAVE (
    CCLAVE        NUMBER(10)      not null,
    CUSUARIO      NUMBER(10),
    XCLAVE        VARCHAR2(200),
    FCLAVE        DATE,
```



constraint PK_SEHISCLAVE primary key (CCLAVE)

);

```
/*=====
===*/
```

```
/* Index: USUARIO_HIS_CLAVE_FK */
```

```
/*=====
===*/
```

create index USUARIO_HIS_CLAVE_FK on SEHISCLAVE (

CUSUARIO ASC

);

```
/*=====
===*/
```

```
/* Table: SEPAGINA */
```

```
/*=====
===*/
```

create table SEPAGINA (

CPAGINA NUMBER(10) not null,

XNOMPAG VARCHAR2(50),

XDESCRIPCION VARCHAR2(400),

IRESTRIN VARCHAR2(2),

IESTADO VARCHAR2(2),

FINGRESO DATE,

FMODIF DATE,

constraint PK_SEPAGINA primary key (CPAGINA)

);

```
/*=====
===*/
```



```
/* Table: SEPERPAG */

/*=====
===*/

create table SEPERPAG (

    CPERPAG      NUMBER(10)      not null,

    CPAGINA      NUMBER(10),

    CROL         NUMBER(10),

    IESTADO      VARCHAR2(2),

    FINGRESO     DATE,

    FMODIF       DATE,

    constraint PK_SEPERPAG primary key (CPERPAG)

);

/*=====
===*/

/* Index: SEPROG_SEPERPR_FK */

/*=====
===*/

create index SEPROG_SEPERPR_FK on SEPERPAG (

    CPAGINA ASC

);

/*=====
===*/

/* Index: SEPERPROG_ROL_FK */

/*=====
===*/

create index SEPERPROG_ROL_FK on SEPERPAG (

    CROL ASC
```



```
);

/*=====
===*/

/* Table: SEROL */

/*=====
===*/

create table SEROL (

    CROL          NUMBER(10)          not null,

    CEMPRESA      NUMBER(10),

    XNOMROL       VARCHAR2(50),

    XDESCRIPCION  VARCHAR2(400),

    FMODIF        DATE,

    IESTADO       VARCHAR2(2),

    FINGRESO      DATE,

    constraint PK_SEROL primary key (CROL)

);

/*=====
===*/

/* Index: EMPRESA_ROL_FK */

/*=====
===*/

create index EMPRESA_ROL_FK on SEROL (

    CEMPRESA ASC

);

/*=====
===*/

/* Table: SEUSUARIO */
```



```
/*=====
===*/
```

```
create table SEUSUARIO (
    CUSUARIO      NUMBER(10)      not null,
    CROL          NUMBER(10),
    CEMPRESA      NUMBER(10),
    XNOMBRE1      VARCHAR2(20),
    XNOMBRE2      VARCHAR2(20),
    XAPELLIDO1    VARCHAR2(20),
    XAPELLIDO2    VARCHAR2(20),
    FNACIMIENTO   DATE,
    IGENERO       VARCHAR2(1),
    XTELEFONO     VARCHAR2(20),
    XCORREO       VARCHAR2(60),
    XNICK         VARCHAR2(30),
    XCLAVE        VARCHAR2(200),
    FCLAVE        DATE,
    FMODIF        DATE,
    IESTADO       VARCHAR2(2),
    FINGRESO      DATE,
    constraint PK_SEUSUARIO primary key (CUSUARIO)
);
```

```
/*=====
===*/
```

```
/* Index: EMPRESA_USUARIO_FK */
```



```
/*=====
===*/
```

```
create index EMPRESA_USUARIO_FK on SEUSUARIO (
    CEMPRESA ASC
);
```

```
/*=====
===*/
```

```
/* Index: ROL_USUARIO_FK */
```

```
/*=====
===*/
```

```
create index ROL_USUARIO_FK on SEUSUARIO (
    CROL ASC
);
```

```
alter table PRCOMPONENTE
```

```
    add constraint FK_PRCOMPON_COMPONENT_PRSISTEM foreign key
(CSISTEMA)
```

```
    references PRSISTEMA (CSISTEMA);
```

```
alter table PRCOMPONENTE
```

```
    add constraint FK_PRCOMPON_TIP_COMPO_PRTIPCOM foreign key
(CTIPCOMP)
```

```
    references PRTIPCOMPONENTE (CTIPCOMP);
```

```
alter table PRDETCOMP
```

```
    add constraint FK_PRDETCOM_COMP_DETC_PRCOMPON foreign key
(CCOMPONENTE)
```

```
    references PRCOMPONENTE (CCOMPONENTE);
```

```
alter table PRDETCOMP
```

```
    add constraint FK_PRDETCOM_PROYECTO__PRPROYEC foreign key
(CPROYECTO)
```



references PRPROYECTO (CPROYECTO);

alter table PRDETDOCUM

add constraint FK_PRDETDOC_DOCUMENTA_PRPROYEC foreign key
(CPROYECTO)

references PRPROYECTO (CPROYECTO);

alter table PRDETDOCUM

add constraint FK_PRDETDOC_TIP_DOCUM_PRTIPDOC foreign key
(CTIPDOCUM)

references PRTIPDOCUMENTACION (CTIPDOCUM);

alter table PRDETDOCUM

add constraint FK_PRDETDOC_USUARIO_D_SEUSUARI foreign key
(CUSUARIO)

references SEUSUARIO (CUSUARIO);

alter table PRDETPROD

add constraint FK_PRDETPRO_DET_PROD__PRTIPOBS foreign key
(CTIPOBS)

references PRTIPOBSERVACION (CTIPOBS);

alter table PRDETPROD

add constraint FK_PRDETPRO_PROYECTOS_PRPROYEC foreign key
(CPROYECTO)

references PRPROYECTO (CPROYECTO);

alter table PRDETPROD

add constraint FK_PRDETPRO_USU_DET_P_SEUSUARI foreign key
(CUSUARIO)

references SEUSUARIO (CUSUARIO);

alter table PRDETPRUEBAS

add constraint FK_PRDETPRU_PROYECTO__PRPROYEC foreign key
(CPROYECTO)



references PRPROYECTO (CPROYECTO);

alter table PRDETPRUEBAS

add constraint FK_PRDETPRU_TIP_OBSER_PRTIPOBS foreign key
(CTIPOBS)

references PRTIPOBSERVACION (CTIPOBS);

alter table PRDETPRUEBAS

add constraint FK_PRDETPRU_USUARIO_D_SEUSUARI foreign key
(CUSUARIO)

references SEUSUARIO (CUSUARIO);

alter table PRPROYECTO

add constraint FK_PRPROYEC_AREA_PROY_SEAREA foreign key (CAREA)

references SEAREA (CAREA);

alter table PRPROYECTO

add constraint FK_PRPROYEC_PRIORIDAD_PRPRIORI foreign key
(CPRIORIDAD)

references PRPRIORIDAD (CPRIORIDAD);

alter table PRPROYECTO

add constraint FK_PRPROYEC_SISTEMAS__PRSISTEM foreign key
(CSISTEMA)

references PRSISTEMA (CSISTEMA);

alter table PRPROYECTO

add constraint FK_PRPROYEC_TIPO_PROY_PRTIPPRO foreign key
(CTIPPRO)

references PRTIPPROYECTO (CTIPPRO);

alter table PRVERSIONAMIENTO

add constraint FK_PRVERSIO_PROYECTO__PRPROYEC foreign key
(CPROYECTO)



references PRPROYECTO (CPROYECTO);

alter table SEAREA

add constraint FK_SEAREA_EMPRESA_A_SEEMPRES foreign key
(CEMPRESA)

references SEEMPRESA (CEMPRESA);

alter table SEHISCLAVE

add constraint FK_SEHISCLA_USUARIO_H_SEUSUARI foreign key
(CUSUARIO)

references SEUSUARIO (CUSUARIO);

alter table SEPERPAG

add constraint FK_SEPERPAG_SEPERPROG_SEROL foreign key (CROL)

references SEROL (CROL);

alter table SEPERPAG

add constraint FK_SEPERPAG_SEPROG_SE_SEPAGINA foreign key
(CPAGINA)

references SEPAGINA (CPAGINA);

alter table SEROL

add constraint FK_SEROL_EMPRESA_R_SEEMPRES foreign key
(CEMPRESA)

references SEEMPRESA (CEMPRESA);

alter table SEUSUARIO

add constraint FK_SEUSUARI_EMPRESA_U_SEEMPRES foreign key
(CEMPRESA)

references SEEMPRESA (CEMPRESA);

alter table SEUSUARIO

add constraint FK_SEUSUARI_ROL_USUAR_SEROL foreign key (CROL)

references SEROL (CROL);

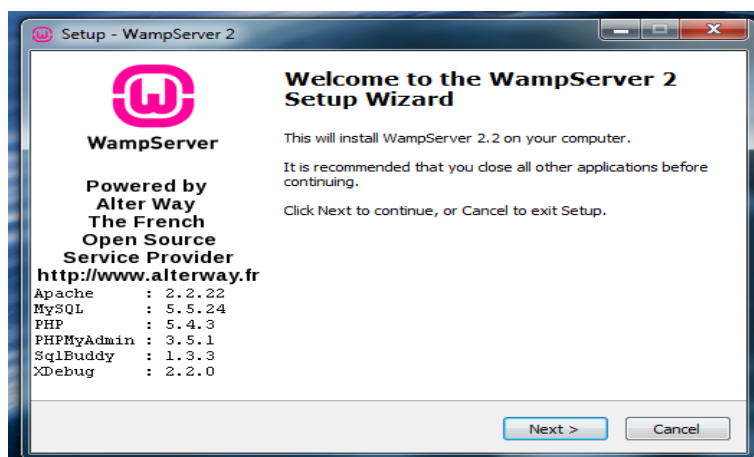


Anexo 06: Manual de Implementación

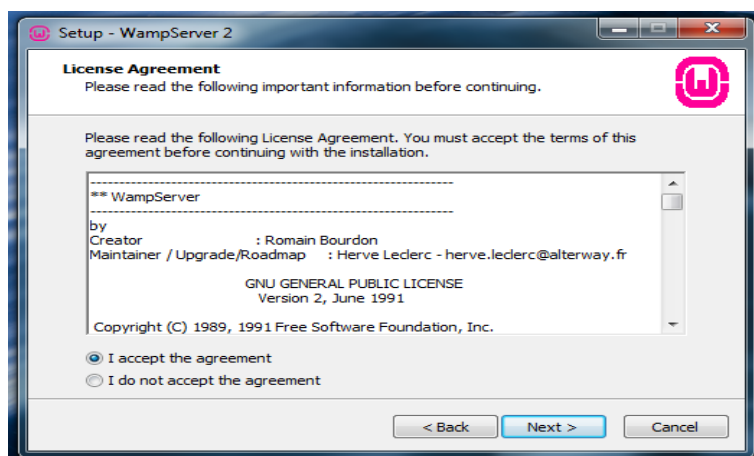
Para la implementación del sistema se deberán seguir los siguientes pasos:

Instalación de WampServer:

1. Ejecutamos el instalador y damos clic en siguiente.

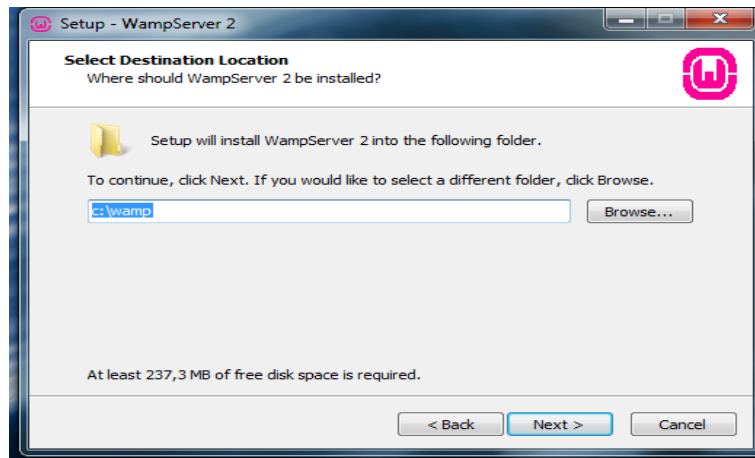


2. Aceptamos los términos de la licencia y damos clic en siguiente.

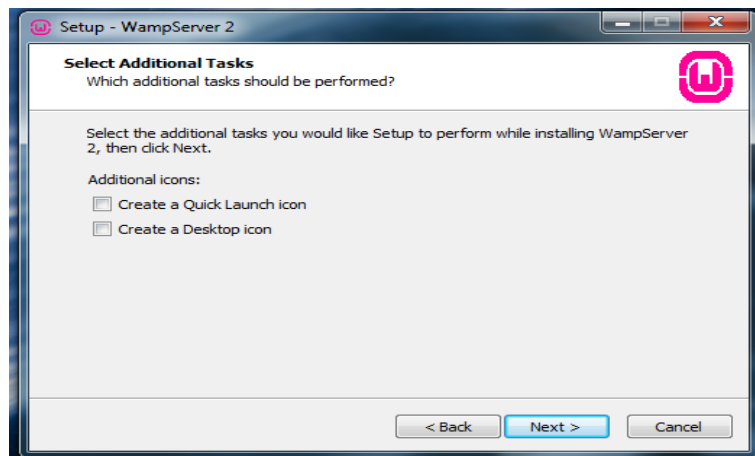




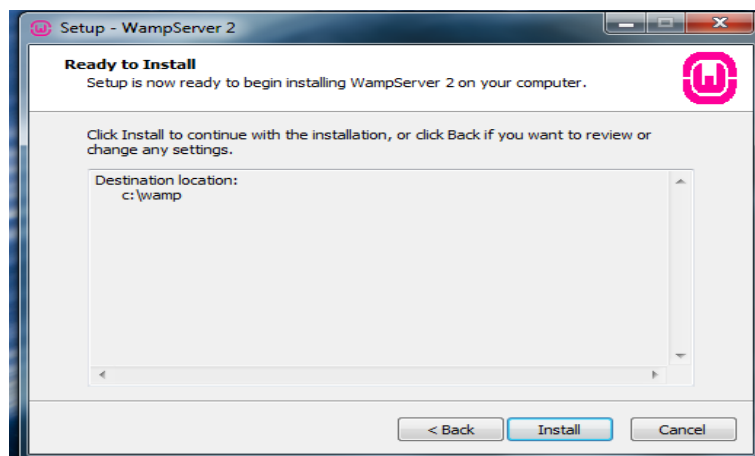
3. Escogemos el directorio de instalación, dejamos el que viene por defecto y damos clic en siguiente.



4. Seleccionamos las opciones si deseamos que se creen accesos directos en el escritorio, damos clic en siguiente.

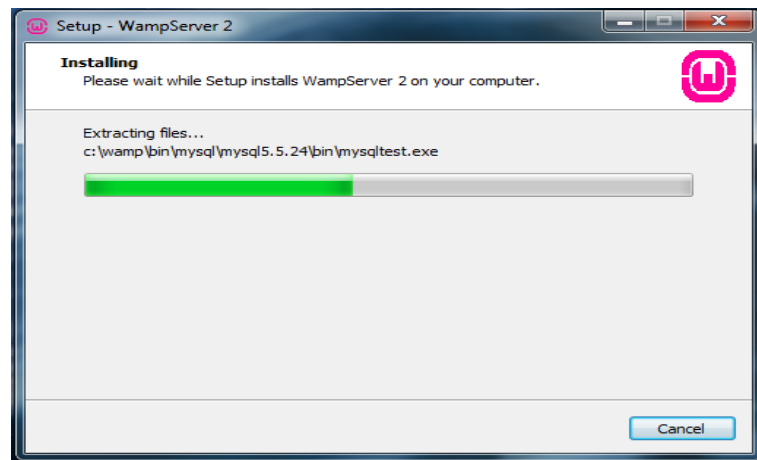


5. Confirmamos la instalación haciendo clic en el botón install.

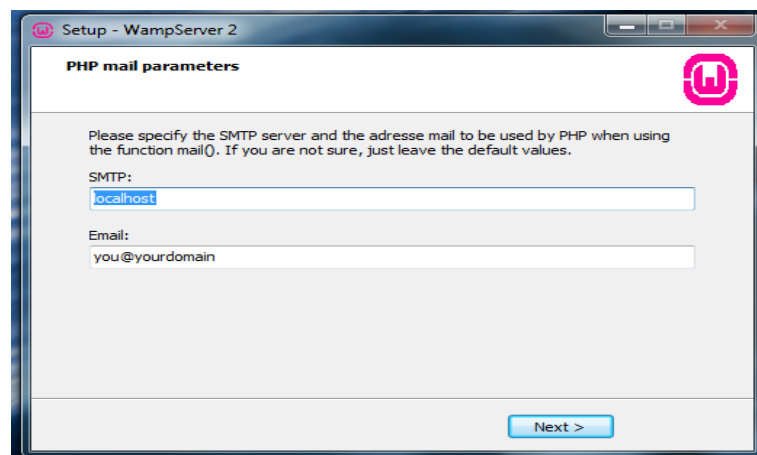




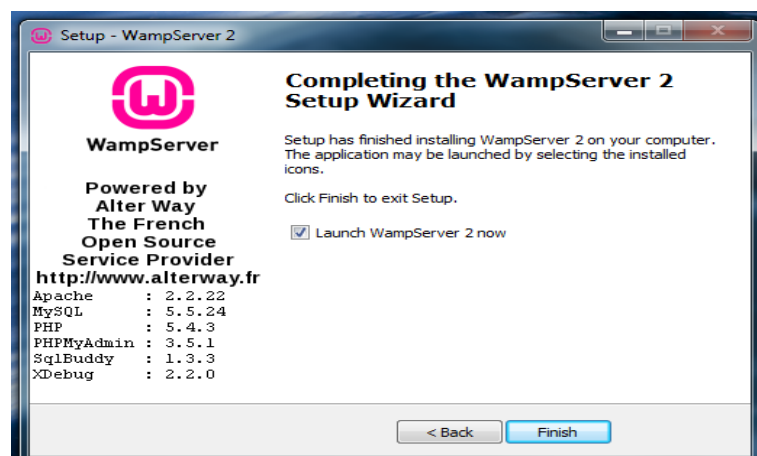
6. Esperamos a que se copien los archivos de instalación.



7. Configuramos las opciones de envío de mail, las dejamos por defecto y damos clic en siguiente.

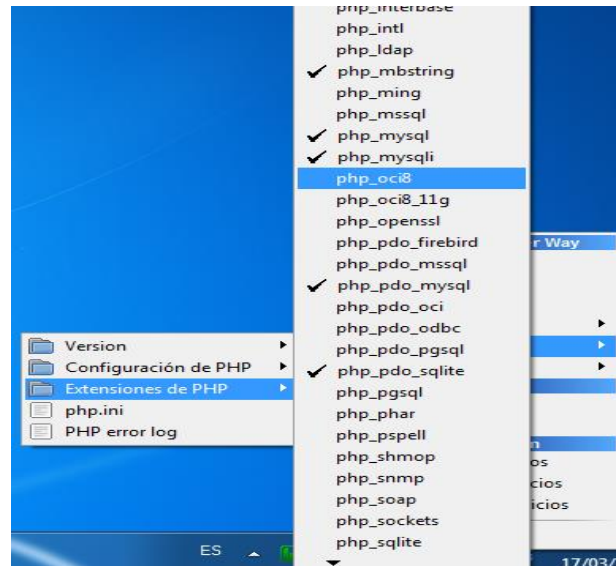


8. Damos clic en finalizar.



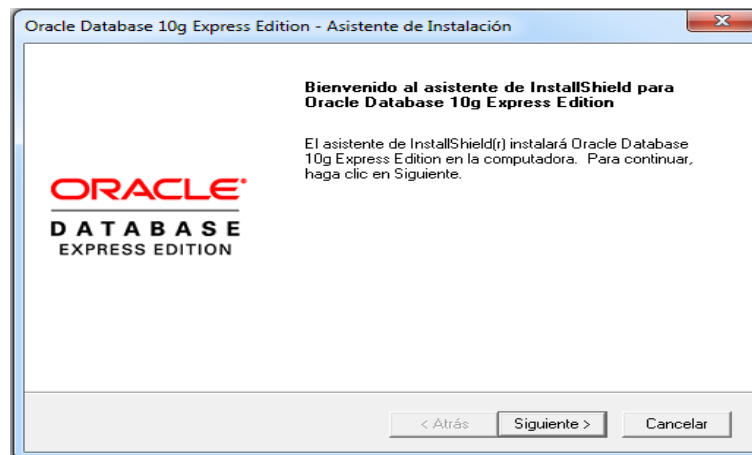


9. Habilitamos la librería OCI8 de PHP, en el icono verde de WampServer ubicado en la barra de tareas damos clic, seleccionamos la opción **PHP**, luego **Extensiones de PHP**, y damos clic en la opción **php_oci8**.

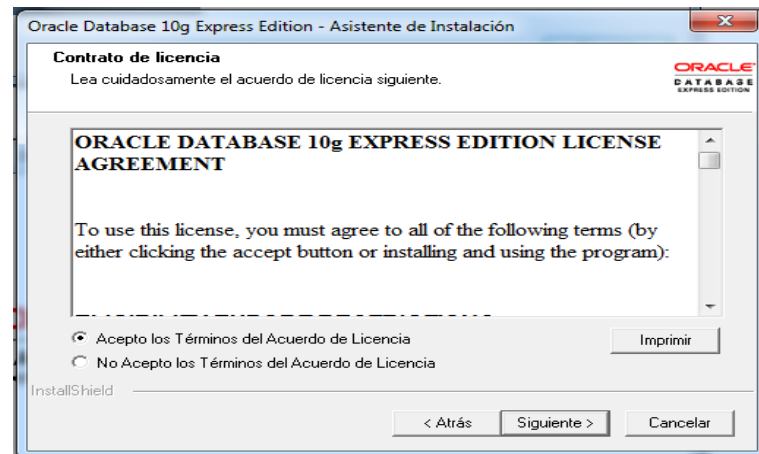


Instalación de Oracle:

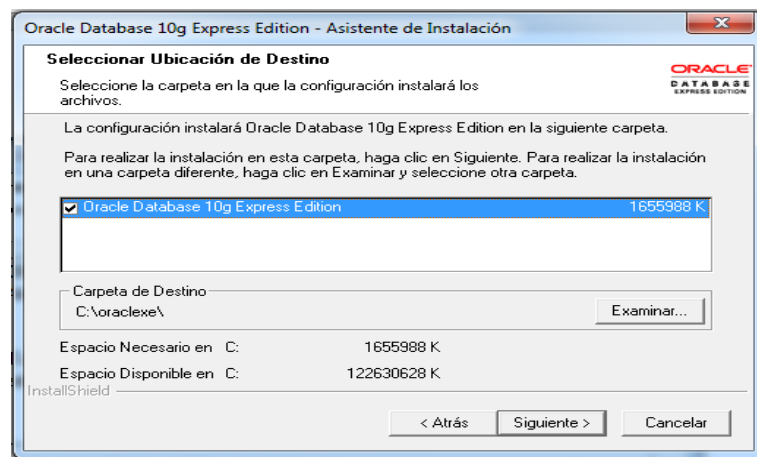
1. Ejecutamos el instalador y damos clic en siguiente.



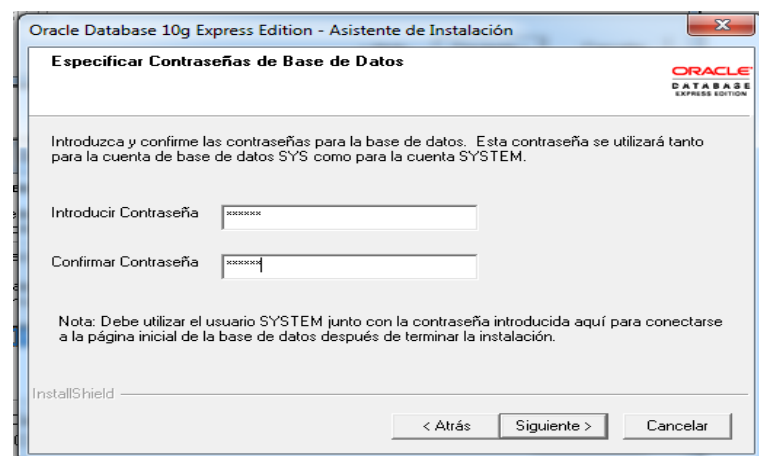
2. Aceptamos el acuerdo de licencia.



3. Seleccionamos el directorio de instalación.

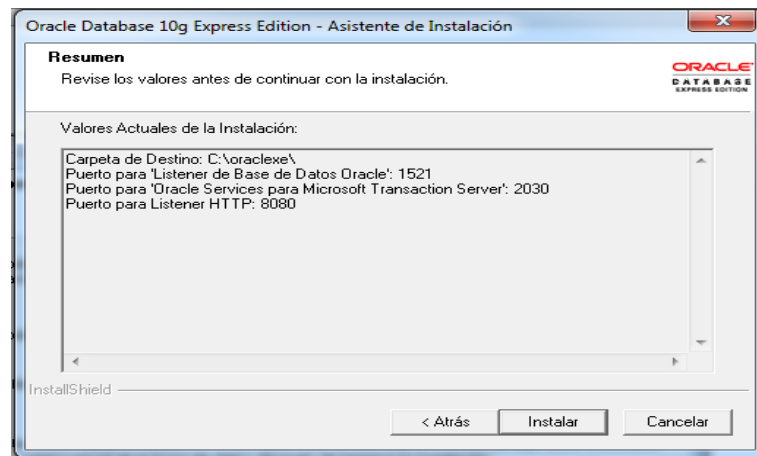


4. Establecemos la contraseña para el usuario SYSTEM de la base de datos.

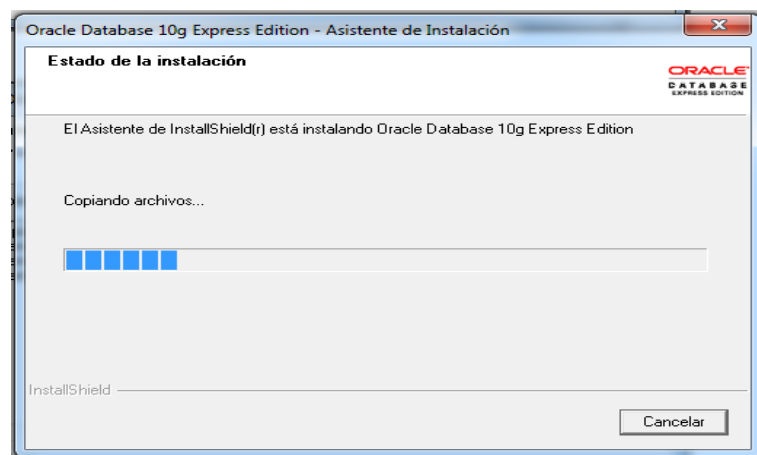




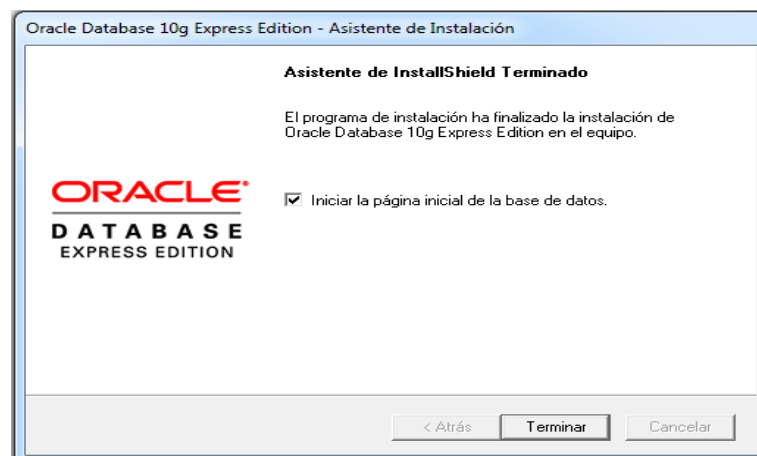
5. Verificamos las opciones de instalación y damos clic en Instalar.



6. Esperamos hasta que se copien los archivos de instalación.



7. Damos clic en finalizar para terminar con la instalación.



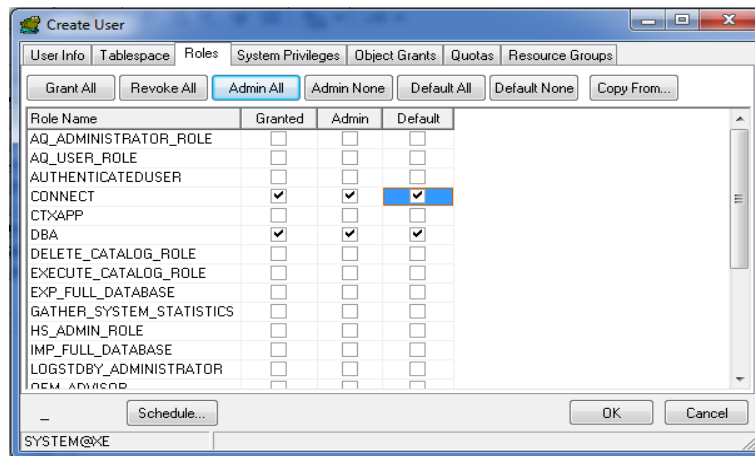


Creación de la Base de Datos:

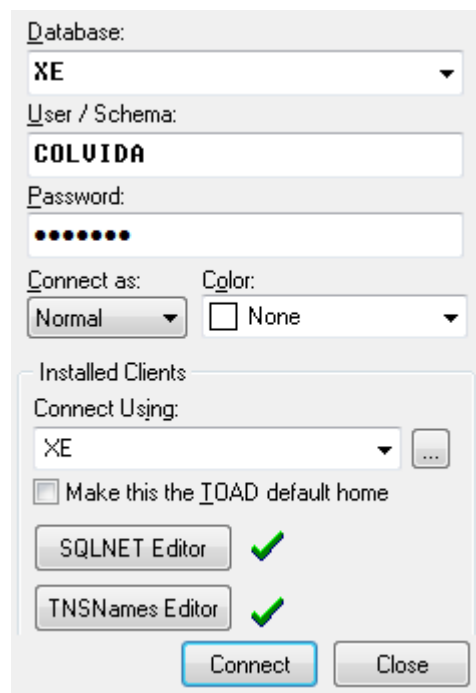
1. Para la administración de la base de datos utilizaremos TOAD.
2. Iniciamos sesión con el usuario **SYSTEM** y la contraseña que definimos en la instalación de Oracle.

3. Creamos un nuevo usuario para la base de datos, para esto en **Users, Create User**. Para la creación del nuevo usuario debemos tener las siguientes consideraciones:

- **User Name:** COLVIDA
- **New Password:** COLVIDA
- **En la pestaña Roles:** Seleccionamos las opciones CONNECT y DBA



4. Damos clic en OK para que se cree el nuevo usuario. Luego nos desconectamos e iniciamos una nueva sesión con el usuario que acabamos de crear.



5. Abrimos un **SQL EDITOR** y copiamos el script de la base de datos que se encuentra en la carpeta base de datos.

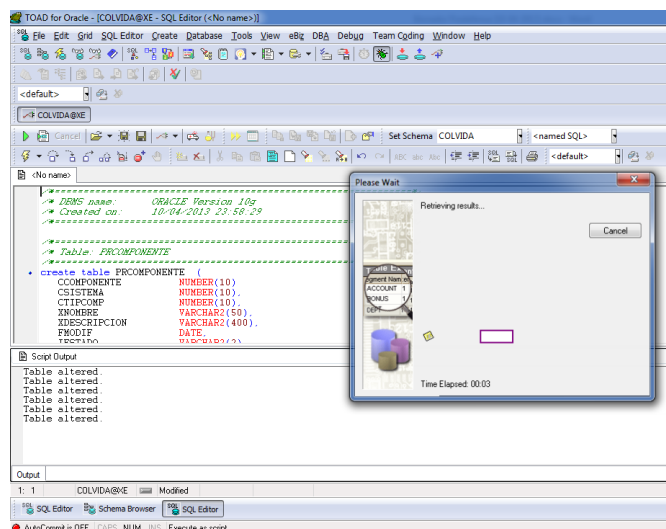


```
/*=====*/
/* DEMS name: ORACLE Version 10g */
/* Created on: 10/04/2013 23:58:29 */
/*=====*/

/* Table: PRCOMPONENTE */
/*=====*/
create table PRCOMPONENTE (
  CCOMPONENTE          NUMBER(10)                not null,
  CSISTEMA              NUMBER(10),
  CTIPCOMP              NUMBER(10),
  XNOMBRE               VARCHAR2(50),
  XDESCRIPCION          VARCHAR2(400),
  FMODIF               DATE,
  IESTADO               VARCHAR2(2),
  FINGRESO              DATE,
  XVERANT               VARCHAR2(15),
  XVERNUE               VARCHAR2(15),
  CPROYECT              NUMBER(10),
  constraint PK_PRCOMPONENTE primary key (CCOMPONENTE)
);

/*=====*/
/* Index: COMPONENTE_SISTEMA_FK */
/*=====*/
create index COMPONENTE_SISTEMA_FK on PRCOMPONENTE (
  CSISTEMA ASC
);
```

6. Luego corremos el script dando clic en el botón **Execute as script**, esperamos a que concluya la ejecución del script.



7. En otro **SQL EDITOR**, copiamos el script de las secuencias que también se encuentra en la carpeta base de datos.



```
--SECUENCIA PRCOMPONENTE
CREATE SEQUENCE COLVIDA.SQ_PRCOMPONENTE
START WITH 1
INCREMENT BY 1
MINVALUE 1
NOCACHE
NOCYCLE
NOORDER;

--SECUENCIA PRDETPROD
CREATE SEQUENCE COLVIDA.SQ_PRDETPROD
START WITH 1
INCREMENT BY 1
MINVALUE 1
NOCACHE
NOCYCLE
NOORDER;

--SECUENCIA PRDETPRUEBAS
CREATE SEQUENCE COLVIDA.SQ_PRDETPRUEBAS
START WITH 1
INCREMENT BY 1
MINVALUE 1
NOCACHE
NOCYCLE
NOORDER;

--SECUENCIA PRDETCOMI
```

8. Luego corremos el script dando clic en el botón **Execute as script**, esperamos a que concluya la ejecución del script.

```
--SECUENCIA PRCOMPONENTE
• CREATE SEQUENCE COLVIDA.SQ_PRCOMPONENTE
  START WITH 1
  INCREMENT BY 1
  MINVALUE 1
  NOCACHE
  NOCYCLE
  NOORDER;

--SECUENCIA PRDETPROD
• CREATE SEQUENCE COLVIDA.SQ_PRDETPROD
  START WITH 1
  INCREMENT BY 1
  MINVALUE 1
  NOCACHE
  NOCYCLE
```

Script Output

```
Sequence created.
Sequence created.
Sequence created.
Sequence created.
Sequence created.
Sequence created.
```

9. Abrimos otro **SQL EDITOR**, y copiamos el script de triggers.

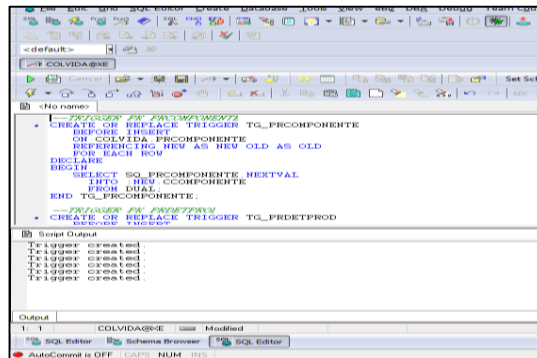


```
--TRIGGER FK PRCOMPONENTE
CREATE OR REPLACE TRIGGER TG_PRCOMPONENTE
BEFORE INSERT
ON COLVIDA.PRCOMPONENTE
REFERENCING NEW AS NEW OLD AS OLD
FOR EACH ROW
DECLARE
BEGIN
    SELECT SQ_PRCOMPONENTE.NEXTVAL
    INTO :NEW.CCOMPONENTE
    FROM DUAL;
END TG_PRCOMPONENTE;

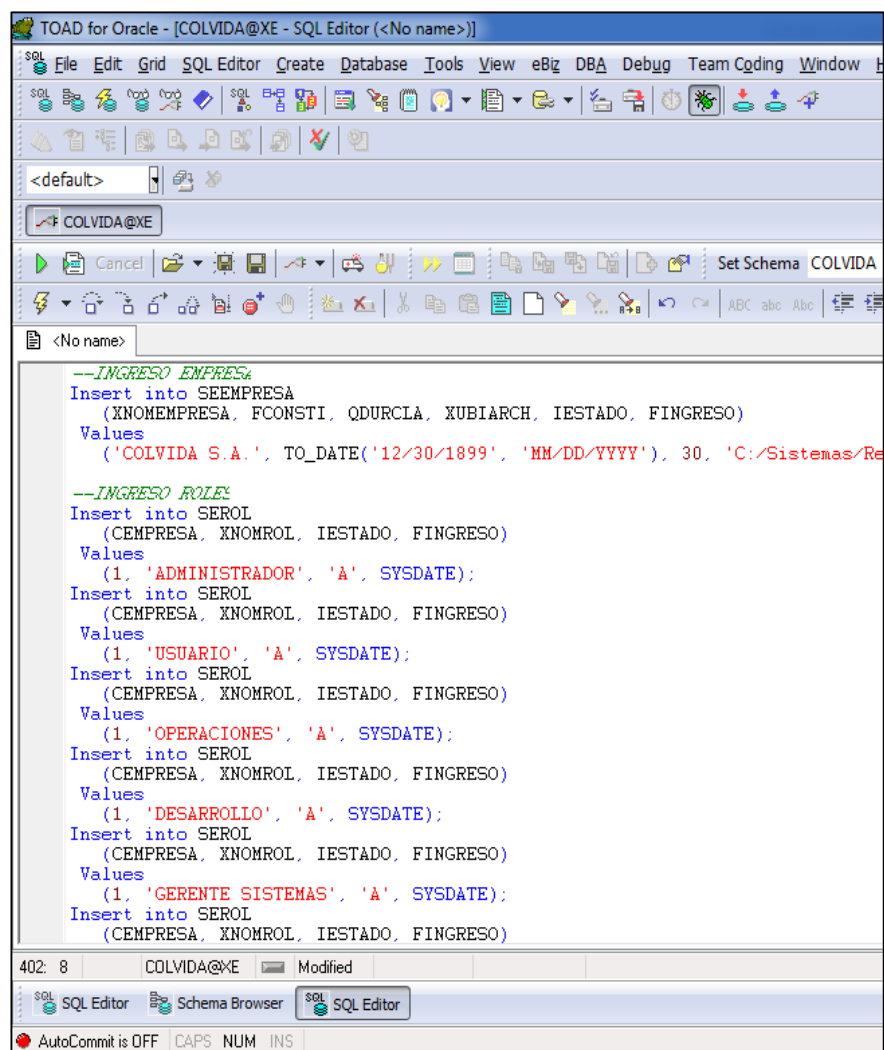
--TRIGGER FK PRDETPROD
CREATE OR REPLACE TRIGGER TG_PRDETPROD
BEFORE INSERT
ON COLVIDA.PRDETPROD
REFERENCING NEW AS NEW OLD AS OLD
FOR EACH ROW
DECLARE
BEGIN
    SELECT SQ_PRDETPROD.NEXTVAL
    INTO :NEW.CDETPROD
    FROM DUAL;
END TG_PRDETPROD;

--TRIGGER FK PRDETPRUEBAS
CREATE OR REPLACE TRIGGER TG_PRDETPRUEBAS
BEFORE INSERT
```

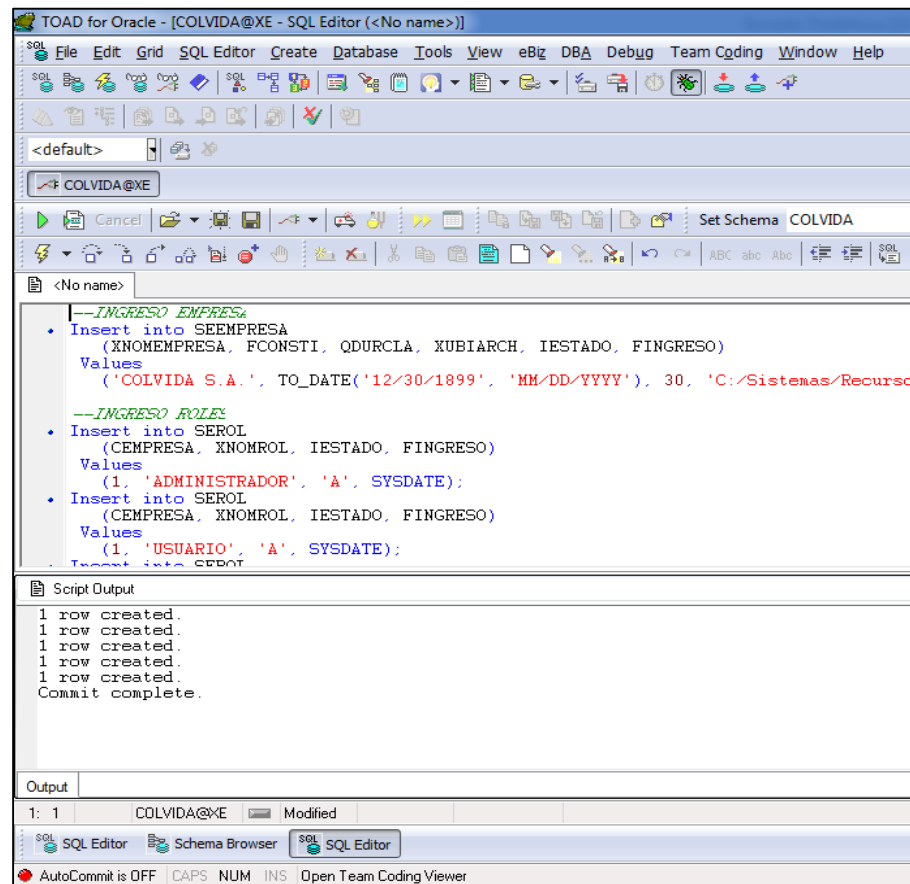
10. Ejecutamos el script dando clic en el botón **Execute as script**, esperamos a que concluya la ejecución del script.



11. Por ultimo copiamos el script de datos iniciales en otro **SQL EDITOR**.

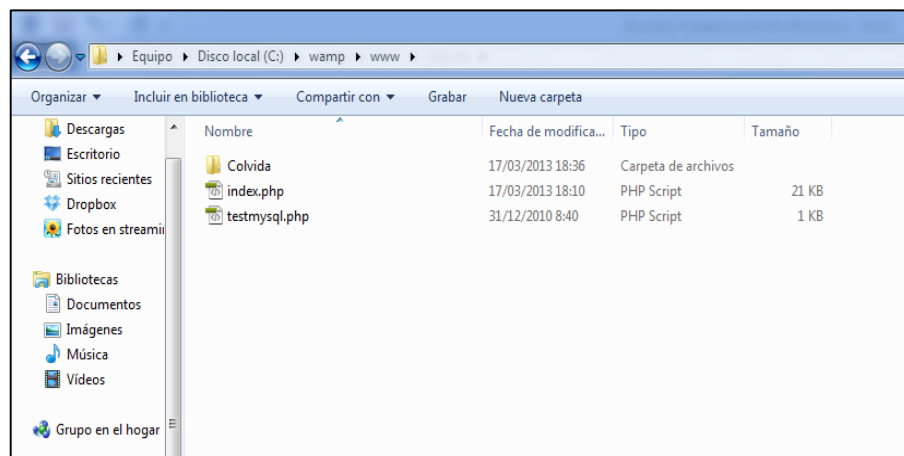


12. Ejecutamos el script dando clic en el botón **Execute as script**, esperamos a que concluya la ejecución del script.



Copia de Códigos Fuente

Este proceso consiste en pegar los códigos fuente en la carpeta de publicación de WampServer, para esto copiamos la carpeta **COLVIDA** a cual contiene todos los códigos fuentes del sistema en la siguiente dirección: **C:\wamp\www**.





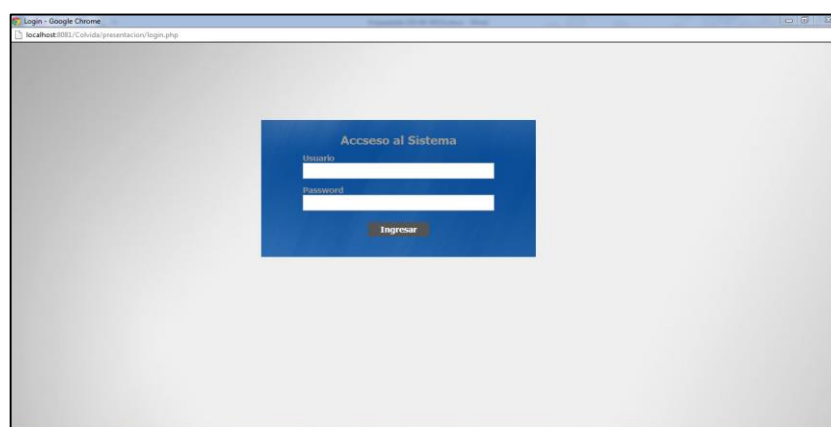
Anexo 07: Manual de Usuario

El siguiente manual describe la forma en la que debe operar el usuario el sistema.

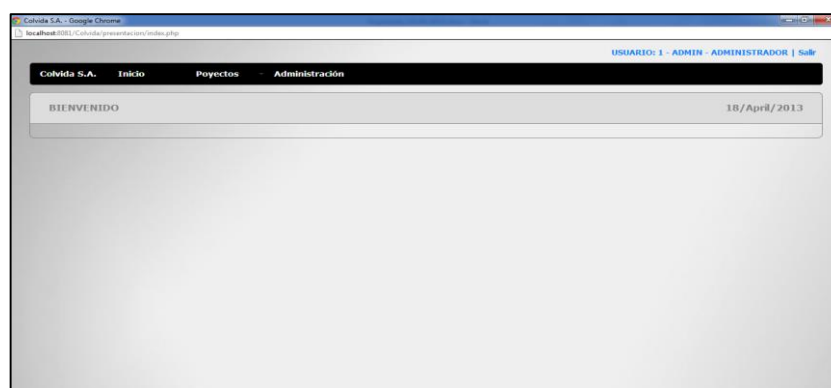
1. Inicio de Sesión

Como requisito para iniciar sesión dentro del sistema es tener un usuario y contraseña, el cual en caso de no poseerlo deberá solicitarlo al administrador del sistema.

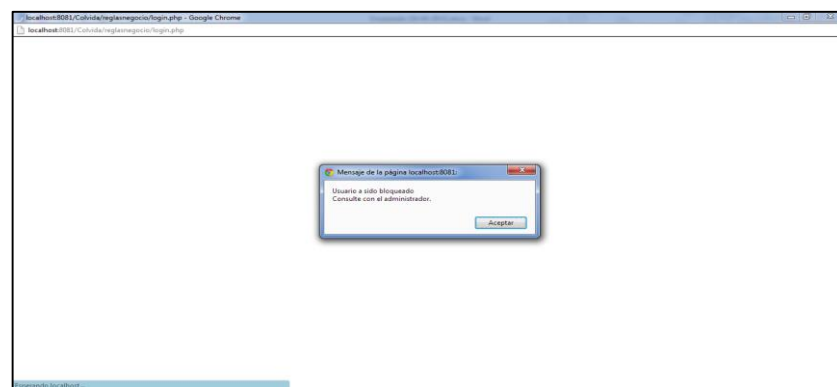
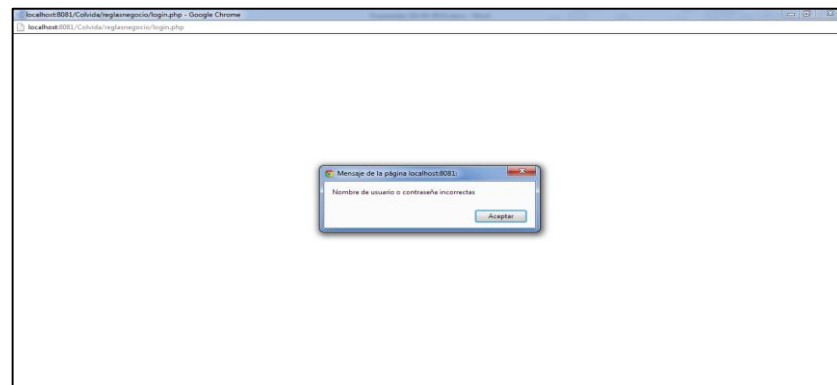
Para iniciar al sistema ingresamos a la ventana principal del sistema.



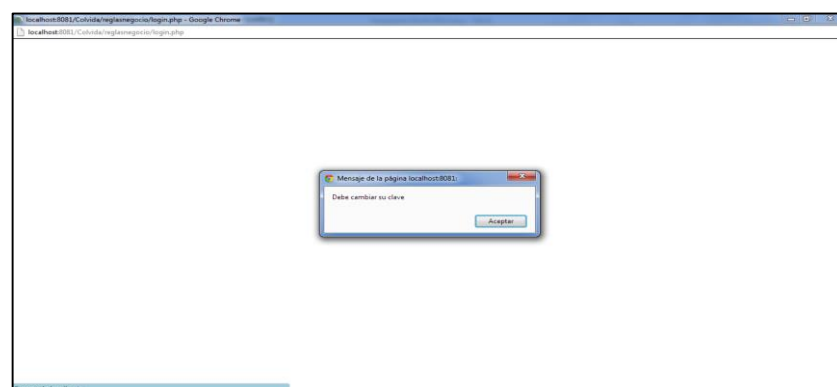
Luego en el campo de Usuario ingresamos el nombre de usuario que nos hayan asignado, y en el campo Password la clave asignada, damos clic en ingresar y nos validara nuestras credenciales si son correctas el sistema nos mostrara la pantalla de inicio del sistema.



Si son incorrectas nos mostrara una alerta, en caso de que el ingreso sea erróneo en tres ocasiones nuestro usuario será bloqueado automáticamente por el sistema y para su desbloqueo se solicitara al administrador del sistema.



Si es la primera vez que el usuario ingresa al sistema automáticamente le obligara a cambiar su contraseña, para lo cual el sistema tiene algunos estándares de seguridad, para cambiar su contraseña deberá ingresar su contraseña actual, la nueva contraseña y confirmar la nueva contraseña luego deberá presionar el botón cambiar para que se guarde la nueva contraseña.





Cambio de Clave

Usuario: RHORALES

Clave Anterior

Clave Nueva

Confirmar Clave

Cambiar

Cancelar

2. Mantenimiento de Tablas Maestras

Para el mantenimiento de tablas maestras se debe haber ingresado al sistema con el perfil de administrador.

Para acceder a cualquiera de los mantenimientos de las tablas maestras del sistema vamos a: Administración, y escogemos la tabla que deseamos administrar.

Colvida S.A. Inicio Proyectos Administración Seguridad Proyectos Empresa Arca Rol Usuario Páginas

BIENVENIDO

USUARIO: 1 - ADMIN - ADMINISTRADOR | Salir

19/April/2013















Luego al dar clic en la opción seleccionada nos mostrara una ventana la cual nos indicara los registros existentes para la opción seleccionada.

Colvida S.A. Inicio Proyectos Administración

USUARIO: 1 - ADMIN - ADMINISTRADOR | Salir

Usuarios

Listado Usuarios +

USUARIO	NOMBRE	CORREO	ESTADO	OPCIONES
ADMIN	SUPER USUARIO ADMINISTRADOR SISTEMA	mmorales@seguroscolvida.com	ACTIVO	 
FRUNEZ	FRANCISCO NUÑEZ LL	mmorales@seguroscolvida.com	ACTIVO	 
SCADENA	IVORIO CADENA L	mmorales@seguroscolvida.com	ACTIVO	 
CHALA	JORGE CHALA LL	mmorales@seguroscolvida.com	ACTIVO	 
PORTIZ	PAUL ORTIZ LL	mmorales@seguroscolvida.com	ACTIVO	 
RHORALES	RICARDO XAVIER MORALES CAICEDO	mmorales@seguroscolvida.com	BLOQUEADO	 
VSANCHEZ	VANESSA SANCHEZ LL	mmorales@seguroscolvida.com	ACTIVO	 

Para insertar un nuevo registro damos clic en el botón nuevo el cual nos

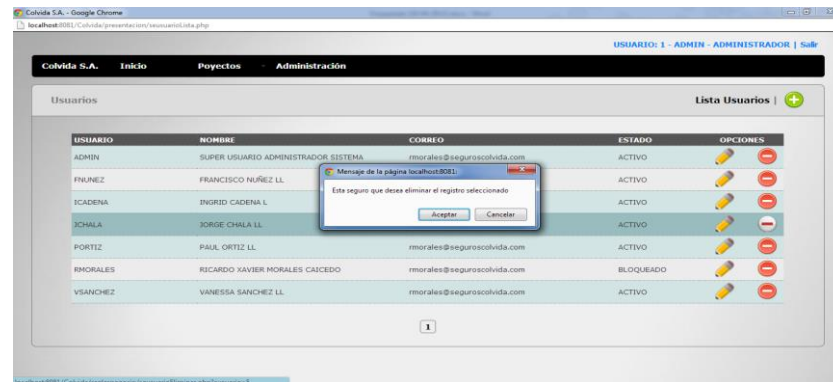


Llevara a un formulario en el cual ingresaremos los datos solicitados por el sistema, para guardar damos clic en el visto verde.

Para modificar un registro en el listado de registros damos clic en el icono del lápiz el cual nos mostrara un mensaje de confirmación de modificación el registro, una vez que confirmemos la alerta nos mostrara un formulario en el cual se encuentran los datos del registro seleccionado, para guardar los cambios damos clic en el icono de visto verde.

USUARIO	NOMBRE	CORREO	ESTADO	OPCIONES
ADMIN	SUPER USUARIO ADMINISTRADOR SISTEMA	mmorales@seguroscolvida.com	ACTIVO	[Pencil] [X]
FINUNEZ	FRANCISCO NUÑEZ LL		ACTIVO	[Pencil] [X]
SCADENA	INGRID CADENA L		ACTIVO	[Pencil] [X]
SCHALA	JORGE CHALA LL		ACTIVO	[Pencil] [X]
PORTIZ	PAUL ORTIZ LL	mmorales@seguroscolvida.com	ACTIVO	[Pencil] [X]
RMORALES	RICARDO XAVIER MORALES CAICEDO	mmorales@seguroscolvida.com	BLOQUEADO	[Pencil] [X]
VISANCHEZ	VANESSA SANCHEZ LL	mmorales@seguroscolvida.com	ACTIVO	[Pencil] [X]

Para eliminar un registro damos clic en el icono de menos de la ventana del listado de registros una vez demos clic el sistema nos pedirá una confirmación de eliminación el cual si es aceptado realiza la eliminación del registro seleccionado.

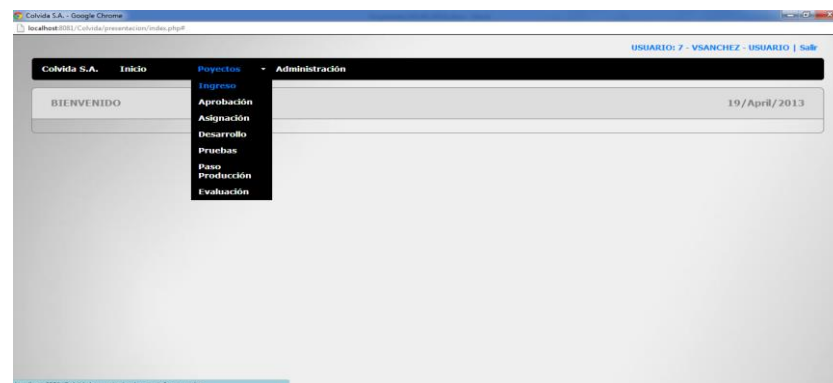


3. Ingreso de Proyectos

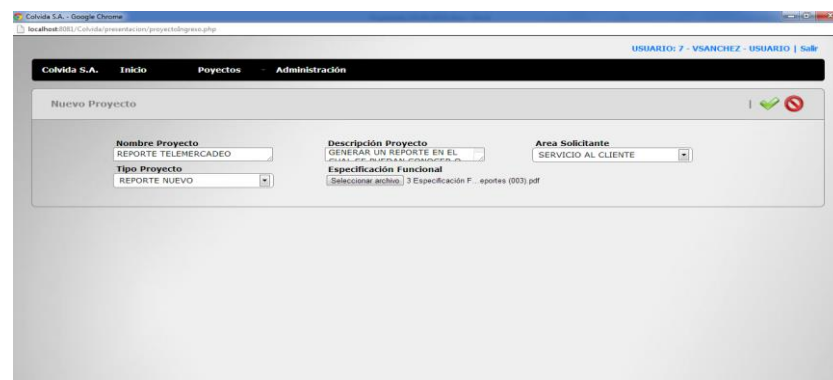
Es el proceso que realiza un usuario para registrar el proyecto en el sistema y este sea procesado, para esto el usuario deberá hacer lo siguiente:

Iniciar sesión con su usuario y contraseña.

Una vez dentro del sistema en el menú principal en la pestaña proyectos deberá seleccionar la opción **Ingreso**.



Una vez seleccionada la opción **Ingreso** en el menú principal se mostrara un formulario en el cual el usuario deberá especificar: nombre, objetivo, área solicitante, tipo de proyecto y adjuntar la especificación funcional del proyecto.





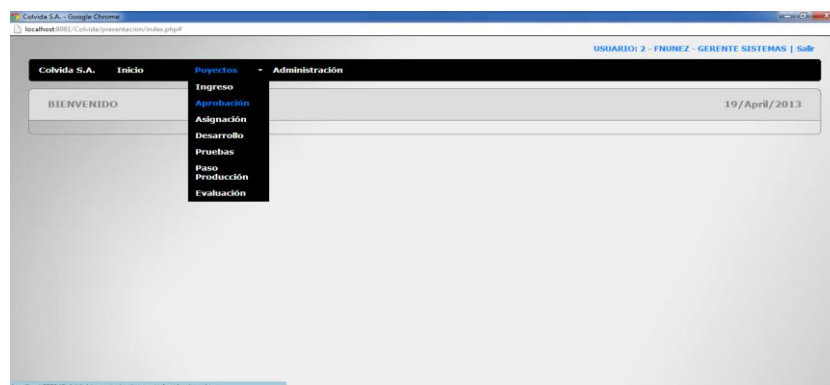
Para registrar el proyecto y enviarlo para su aprobación se debe dar clic en el icono del visto verde.

4. Aprobación de Proyectos

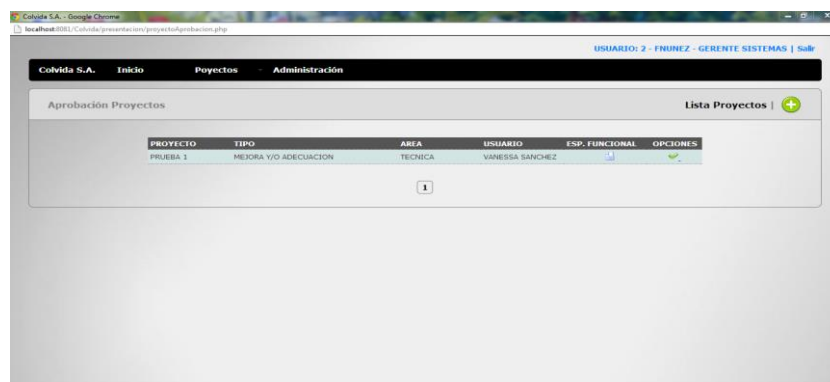
Este proceso es el que realiza el gerente de sistemas para la aprobación o negación de los proyectos solicitados por los usuarios. Para la aprobación de proyectos deberá realizar lo siguiente:

Iniciar sesión con perfil de gerente.

Una vez dentro del sistema en el menú principal en la pestaña Proyectos se deberá escoger la opción **Aprobación**.



Al seleccionar esta opción se nos mostrara una ventana con el listado de proyectos pendientes de aprobación.



Dentro de este listado se puede acceder a la especificación funcional subida al sistema por el usuario solicitante para esto hay que dar clic en el icono de la hoja en la columna de especificación funcional.

Para procesar la aprobación o negación del proyecto hay que dar clic en el icono del visto verde en la columna de opciones, lo cual nos mostrara un formulario en el cual se debe ingresar la prioridad, la fecha de entrega y el estado de la aprobación, en caso de que sea negado se debe llenar el campo



razón de negación.

Colvida S.A. Inicio Proyectos Administración

Aprobación Proyectos Proyecto Pendientes Aprobación |

Nombre Proyecto Descripción Prioridad
PRUEBA 1 DISCOURS MEDIA

Fecha Entrega Estado Aprobación Razón Negación
30/04/2013 APROBADO

USUARIO: 2 - FERNANDEZ - GERENTE SISTEMAS | [Salir](#)

Para guardar los cambios al proyecto se dará clic en el icono del visto verde.

5. Asignación de Proyectos

Es el proceso que realiza el jefe del área de desarrollo de software para la asignación de proyectos a cada asistente de desarrollo. Para este proceso se deberá:

Iniciar sesión con el perfil de Jefe de Desarrollo.

Dentro del sistema en el menú principal en la pestaña Proyectos seleccionamos la opción **Asignación**.

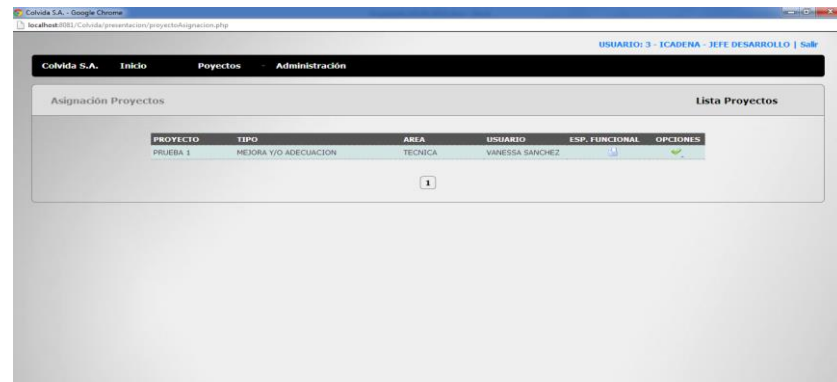
Colvida S.A. Inicio **Proyectos** Administración

BIENVENIDO 19/Abril/2013

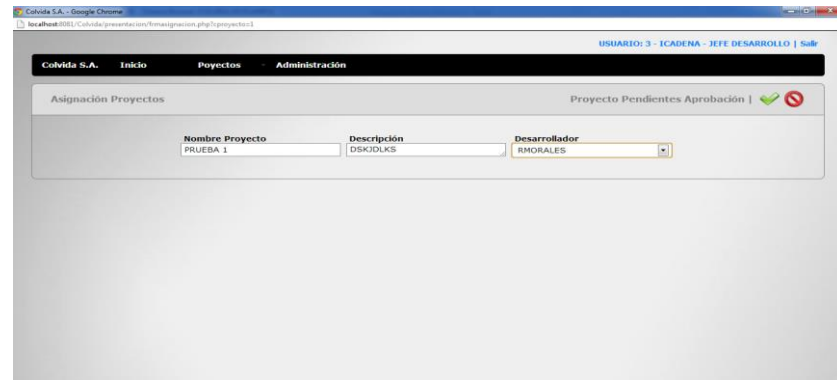
Ingreso
Aprobación
Asignación
Desarrollo
Pruebas
Paso
Producción
Evaluación

USUARIO: 3 - ICADENA - JEFE DESARROLLO | [Salir](#)

Al seleccionar la opción **Asignación** se abrirá una ventana en la cual se encuentran todos los proyectos aprobados pendientes de asignación.



Dentro de este listado se podrá acceder a visualizar la especificación funcional subida por el usuario, para asignar el proyecto hay que dar clic en el visto verde lo cual abrirá un formulario en el cual se deberá seleccionar el asistente de desarrollo.



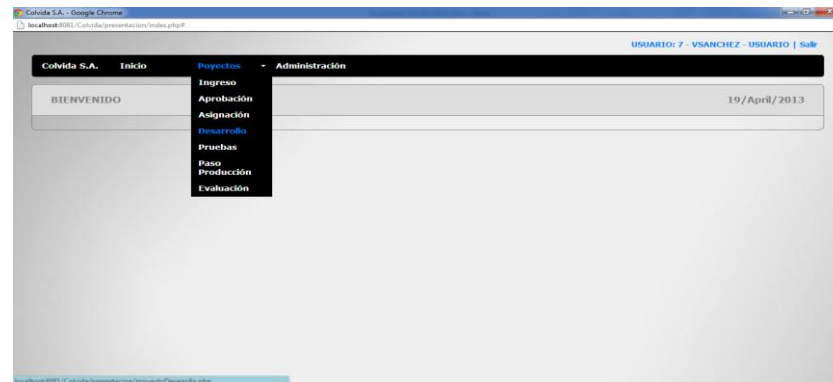
Para guardar los cambios damos clic en el visto verde.

6. Desarrollo de Proyectos

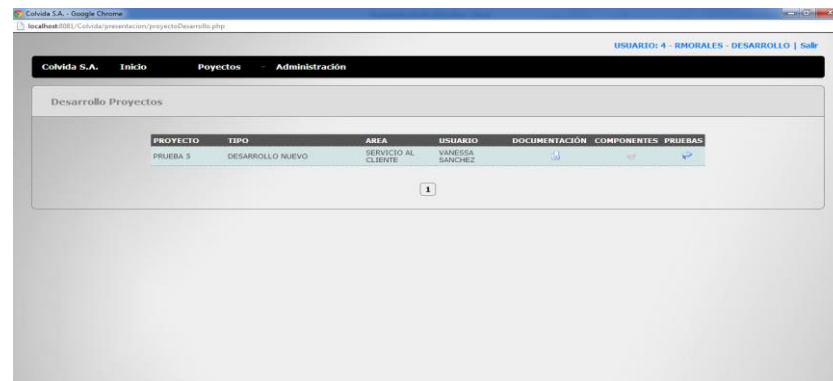
Este es el proceso que realiza el asistente de desarrollo en el cual debe registrar la documentación así como también los componentes que fueron modificados. Para este proceso se debe realizar lo siguiente:

Iniciar sesión como asistente de desarrollo.

Dentro del sistema en el menú principal en la pestaña Proyectos seleccionamos la opción **Desarrollo**.



Al seleccionar esta opción nos mostrara una ventana mostrando el listado de proyectos pendientes de desarrollo asignados al asistente de desarrollo.



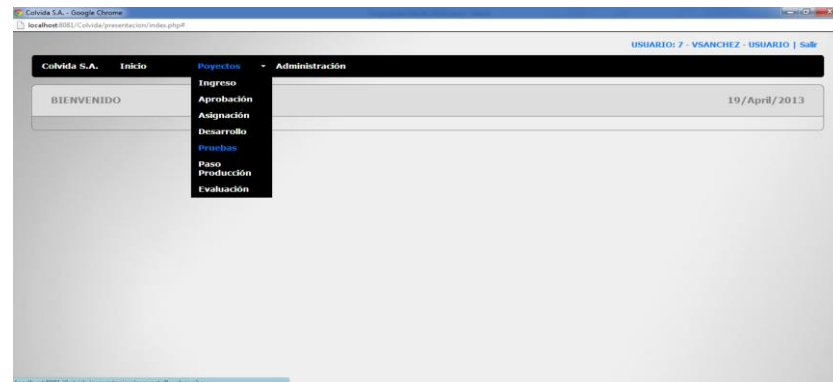
Dentro de esta ventana se puede cargar la documentación, componentes y realizar el paso a pruebas.

7. Pruebas de Proyectos

Es el proceso que realiza el asistente de QA en el cual evalúa los cambios realizados en los sistemas. Para esto se realizara lo siguiente:

Iniciar sesión con perfil de QA.

Dentro del sistema en el menú principal dentro de la pestaña Proyectos seleccionamos la opción **Pruebas**.



Una vez seleccionado esta opción se mostrara una ventana la cual tendrá el listado de los proyectos pendientes de pruebas.

El asistente luego de terminar de realizar las pruebas deberá cargar al sistema el certificado de calidad del proyecto, en caso de existir errores en la implementación el asistente de QA deberá subir al sistema el archivo de observaciones y remitirle al asistente de desarrollo.

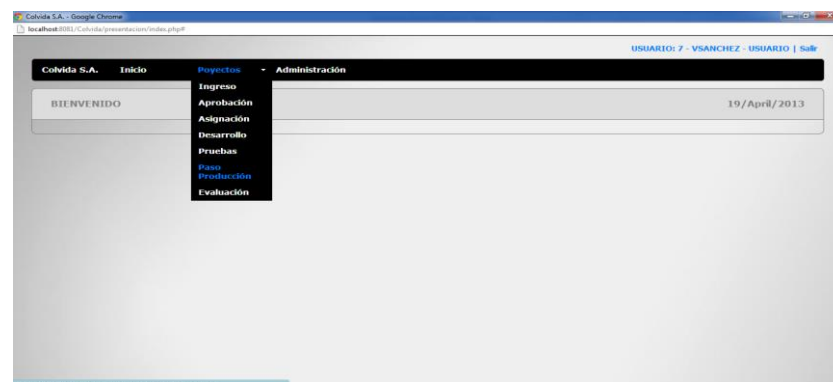
Para guardar los cambios en el sistema damos clic en el visto verde.

8. Paso Producción de Proyectos

Este proceso corresponde a los asistentes de operaciones los cuales implementan los componentes de cada proyecto aprobado por QA y los implementan en producción. Para estos procesos se deberá realizar lo siguiente:

Iniciar sesión con perfil de **Operaciones**.

Una vez dentro del sistema en el menú principal dentro de la pestaña Proyectos se deberá seleccionar la opción **Paso a Producción**.



Dentro de esta venta se mostrara un listado con todos los proyectos que han pasado QA y están listos para su paso a QA.



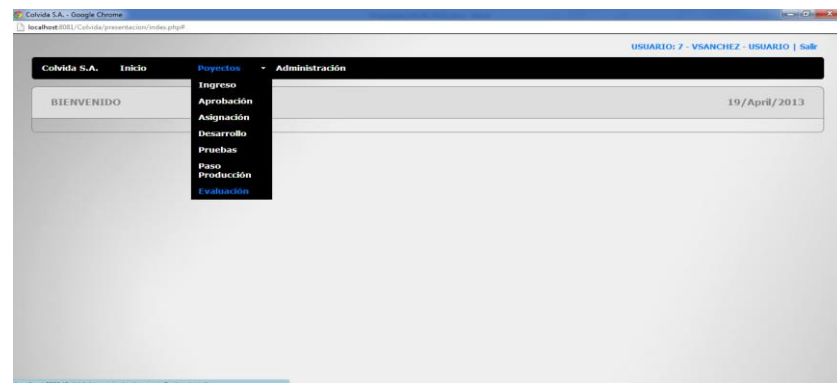
El asistente de operaciones una vez concluida la implementación en producción deberá indicar si hay observaciones en la implementación y guardar los cambios haciendo clic en el visto verde.

9. Evaluación de Proyectos

Es el proceso que realiza el usuario solicitante para evaluar que le proyecto solicitado este de acuerdo a lo estipulado en la especificación funcional. Para esto el usuario deberá realizar lo siguiente:

Iniciar sesión con el perfil de Usuario.

Dentro del sistema en el menú principal en la pestaña Proyectos seleccionamos la opción Evaluación.



Dentro de esta ventana el usuario escogerá el tipo de error si existiera si no deberá ponerlo como que no existen observaciones.

Para guardar los cambios se dará clic en el visto verde.

Anexo 08: Manual Técnico

A continuación se encuentra detallado el código fuente de la clase usuario de la capa de datos del sistema.

```
<?php
include_once "conexion.php";
class usuario
{
    protected $dbcon;

    public function verificarLogin($usuario, $clave)
    {
        $con=new Conexion();
        $dbcon=$con->get_conexion();
```



```
$res=$dbcon->Execute("select * from SEUSUARIO
                        where XNICK='$usuario'
                        and XCLAVE='$clave'
                        and IESTADO='A'");

if($res)
{
    if(!$res->EOF)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
else
{
    return 0;
}
}

public function datosLogin($nick)
{
    $con=new conexion();
    $dbcon=$con->get_conexion();
    $datos=array();
    $res=$dbcon->Execute ("select usu.CUSUARIO, rol.CROL,
                            rol.XNOMROL, usu.XNOMBRE1,
                            usu.XNOMBRE2, usu.XAPELLIDO1,
                            usu.XAPELLIDO2, usu.FCLAVE
                            from seusuario usu, serol rol
                            where usu.crol=rol.crol and
                            usu.XNICK='$nick'");

    while(!$res->EOF)
    {
        $datos[]=array("CUSUARIO"=>$res->fields[0],
                        "CROL"=>$res->fields[1],
                        "XNOMROL"=>$res->fields[2],
                        "XNOMBRE1"=>$res->fields[3],
                        "XNOMBRE2"=>$res->fields[4],
                        "XAPELLIDO1"=>$res->fields[5],
                        "XAPELLIDO2"=>$res->fields[6],
                        "FCLAVE"=>$res->fields[7]);
        $res->MoveNext();
    }
}
```



```
}
$res->close();
return $datos;
}

public function usuarioBloqueado($usuario)
{
    $con=new Conexion();
    $dbcon=$con->get_conexion();
    $res=$dbcon->Execute("select * from SEUSUARIO
                        where XNICK='$usuario' and
                        IESTADO='B'");

    if($res)
    {
        if(!$res->EOF)
        {
            return 1;
        }
        else
        {
            return 0;
        }
    }
    else
    {
        return 0;
    }
}

public function insertar($cusuario, $crol, $cempresa, $xnombre1,
                        $xnombre2, $xapellidio1, $xapellidio2,
                        $fnacimiento, $igenero, $xtelefono, $xcorreo,
                        $xnick, $xclave, $iestado, $fingreso)
{
    $con=new Conexion();
    $dbcon=$con->get_conexion();
    $res=$dbcon->Execute("insert into seusuario (cusuario, crol,
                        cempresa, xnombre1, xnombre2,
                        xapellidio1, xapellidio2, fnacimiento,
                        igenero, xtelefono, xcorreo, xnick, xclave,
                        iestado, fingreso)
                        values
                        ($cusuario, $crol, $cempresa,
                        '$xnombre1', '$xnombre2',
```



```
'$xapellidio1', '$xapellido2',
to_date('$fnacimiento', 'dd/mm/yyyy'),
'$igenero', '$xtelefono', '$xcorreo',
'$xnick', '$xclave', '$iestado',
to_date('$fingreso', 'dd/mm/yyyy/
hh24:mi:ss'))");

if(!$res)
{
    return 1;
}
else
{
    return 0;
}
}

public function modificar($cusuario, $crol, $xnombre1, $xnombre2,
    $xapellidio1, $xapellido2, $fnacimiento,
    $igenero, $xtelefono, $xcorreo, $xclave,
    $iestado, $fmodif)
{
    $con=new Conexion();
    $dbcon=$con->get_conexion();
    $res=$dbcon->Execute("update seusuario set
        crol=$crol, xnombre1='$xnombre1',
        xnombre2='$xnombre2',
        xapellidio1='$xapellidio1',
        xapellido2='$xapellido2',
        fnacimiento=to_date('$fnacimiento',
        'dd/mm/yyyy'), igenero='$igenero',
        xtelefono='$xtelefono',
        xcorreo='$xcorreo', xclave='$xclave',
        iestado='$iestado', fmodif=to_date
        ('$fmodif', 'dd/mm/yyyy hh24:mi:ss')
        where cusuario=$cusuario");

    if(!$res)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
```



```
}
```

```
public function eliminar($cusuario, $fmodif)
{
    $con=new Conexion();
    $dbcon=$con->get_conexion();
    $res=$dbcon->Execute("update seusuario set iestado='E',
                        fmodif=to_date('$fmodif', 'dd/mm/yyyy
                        hh24:mi:ss') where cusuario=$cusuario");

    if(!$res)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
```

```
public function activar($cusuario, $fmodif)
{
    $con=new Conexion();
    $dbcon=$con->get_conexion();
    $res=$dbcon->Execute("update seusuario set iestado='A',
                        fmodif=to_date('$fmodif', 'dd/mm/yyyy
                        hh24:mi:ss') where cusuario=$cusuario");

    if(!$res)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
```

```
public function bloquear($usuario)
{
    $con=new Conexion();
    $dbcon=$con->get_conexion();
    $res=$dbcon->Execute("update seusuario set iestado='B' where
                        xnick='$usuario'");

    if($res)
```



```
{
    if(!$res->EOF)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
else
{
    return 0;
}
}

public function cambiarClave($clave, $usuario, $fclave)
{
    $con=new Conexion();
    $dbcon=$con->get_conexion();
    $res=$dbcon->Execute("update seusuario set xclave='$clave',
                        fclave='$fclave' where
                        cusuario='$usuario'");

    if($res)
    {
        if(!$res->EOF)
        {
            return 1;
        }
        else
        {
            return 0;
        }
    }
    else
    {
        return 0;
    }
}

public function datosCodigo($usuario)
{
    $con=new conexion();
    $dbcon=$con->get_conexion();
```



```
$datos=array();
$res=$dbcon->Execute ("select cusuario, xnombre1, xnombre2,
                        xapellido1, xapellido2, fnacimiento,
                        igenero, xtelefono, xcorreo, xnick,
                        xclave, crol, iestado from seusuario
                        where cusuario=$cusuario");

while(!$res->EOF)
{
    $datos[]=array("cusuario"=>$res->fields[0],
                  "xnombre1"=>$res->fields[1],
                  "xnombre2"=>$res->fields[2],
                  "xapellido1"=>$res->fields[3],
                  "xapellido2"=>$res->fields[4],
                  "fnacimiento"=>$res->fields[5],
                  "igenero"=>$res->fields[6],
                  "xtelefono"=>$res->fields[7],
                  "xcorreo"=>$res->fields[8],
                  "xnick"=>$res->fields[9],
                  "xclave"=>$res->fields[10],
                  "crol"=>$res->fields[11],
                  "iestado"=>$res->fields[12]);
    $res->MoveNext();
}
$res->close();
return $datos;
}

public function listado($sql)
{
    $con=new conexion();
    $dbcon=$con->get_conexion();
    $datos=array();
    $res=$dbcon->Execute ($sql);
    while(!$res->EOF)
    {
        $datos[]=array("CUSUARIO"=>$res->fields[0],
                      "XNOMBRE1"=>$res->fields[1],
                      "XNOMBRE2"=>$res->fields[2],
                      "XAPELLIDO1"=>$res->fields[3],
                      "XAPELLIDO2"=>$res->fields[4],
                      "XNICK"=>$res->fields[5],
                      "XCORREO"=>$res->fields[6],
                      "FMODIF"=>$res->fields[7],
                      "IESTADO"=>$res->fields[8],
```




```
        "RNUM"=>$res->fields[9]);
        $res->MoveNext();
    }
    $res->close();
    return $datos;
}

public function listadoCount($sql)
{
    $con=new conexion();
    $dbcon=$con->get_conexion();
    $datos=array();
    $res=$dbcon->Execute ($sql);
    while(!$res->EOF)
    {
        $datos[]=array("contador"=>$res->fields[0]);
        $res->MoveNext();
    }
    $res->close();
    return $datos;
}

public function contadorPK()
{
    $con=new conexion();
    $dbcon=$con->get_conexion();
    $datos=array();
    $res=$dbcon->Execute ("select max(cusuario) maxusuario from
                           seusuario");
    while(!$res->EOF)
    {
        $datos[]=array("maxusuario"=>$res->fields[0]);
        $res->MoveNext();
    }
    $res->close();
    return $datos;
}

public function consultarNick($xnick)
{
    $con=new conexion();
    $dbcon=$con->get_conexion();
    $datos=array();
    $res=$dbcon->Execute ("select count(*) countnick from
```



```
Seusuario where xnick='$xnick');

while(!$res->EOF)
{
    $datos[]=array("countnick"=>$res->fields[0]);
    $res->MoveNext();
}
$res->close();
return $datos;
}

public function correo($nomrol)
{
    $con=new conexion();
    $dbcon=$con->get_conexion();
    $datos=array();
    $res=$dbcon->Execute ("select xcorreo from seusuario where
                           crol=(select crol from serol where
                           xnomrol='$nomrol')");

    while(!$res->EOF)
    {
        $datos[]=array("correo"=>$res->fields[0]);
        $res->MoveNext();
    }
    $res->close();
    return $datos;
}

public function correo2($cusuario)
{
    $con=new conexion();
    $dbcon=$con->get_conexion();
    $datos=array();
    $res=$dbcon->Execute ("select xcorreo from seusuario where
                           cusuario=$cusuario");

    while(!$res->EOF)
    {
        $datos[]=array("correo"=>$res->fields[0]);
        $res->MoveNext();
    }
    $res->close();
    return $datos;
}

public function consultarnicks()
```



```

{
    $con=new conexion();
    $dbcon=$con->get_conexion();
    $datos=array();
    $res=$dbcon->Execute ("select cusuario, xnick from seusuario
                           where iestado='A'");
    while(!$res->EOF)
    {
        $datos[]=array("cusuario"=>$res->fields[0],
                       "xnick"=>$res->fields[1]);
        $res->MoveNext();
    }
    $res->close();
    return $datos;
}
?>

```

Anexo 09: Cronograma Capacitación

El cronograma que se seguirá para la capacitación a los usuarios será el siguiente:

