



INSTITUTO TECNOLÓGICO
"CORDILLERA"

INSTITUTO TECNOLÓGICO SUPERIOR "CORDILLERA"

CARRERA DE ANÁLISIS DE SISTEMAS

PERSONALIZACIÓN Y EJECUCIÓN DE INTEGRACIONES ENTRE BASES DE
DATOS DE DISTINTAS PLATAFORMAS, POR MEDIO DE UNA
HERRAMIENTA WEB PARA LA EMPRESA PLUS PROJECTS S.A.

Proyecto de investigación previo a la obtención del Título de

Tecnólogo en Análisis de Sistemas

Autor: Carrera Estrada Rodrigo Isaac

Tutor: Jaime Basantes

Quito, Octubre 2014

DECLARATORIA

Declaro que la investigación es absolutamente original, autentica, personal, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes. Las ideas, doctrinas resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.

Rodrigo Isaac Carrera Estrada

CC 171595831-8

CONTRATO DE CESIÓN SOBRE DERECHOS PROPIEDAD INTELECTUAL

Comparecen a la celebración del presente contrato de cesión y transferencia de derechos de propiedad intelectual, por una parte, el estudiante ***Carrera Estrada Rodrigo Isaac***, por sus propios y personales derechos, a quien en lo posterior se le denominará el “CEDENTE”; y, por otra parte, el INSTITUTO SUPERIOR TECNOLÓGICO CORDILLERA, representado por su Rector el Ingeniero Ernesto Flores Córdova, a quien en lo posterior se lo denominará el “CESIONARIO”. Los comparecientes son mayores de edad, domiciliados en esta ciudad de Quito Distrito Metropolitano, hábiles y capaces para contraer derechos y obligaciones, quienes acuerdan al tenor de las siguientes cláusulas:

PRIMERA: ANTECEDENTE.- a) El Cedente dentro del pensum de estudio en la carrera de **Análisis de Sistemas** que imparte el Instituto Superior Tecnológico Cordillera, y con el objeto de obtener el título de Tecnólogo en Análisis de Sistemas, el estudiante participa en el proyecto de grado denominado “**Personalización y ejecución de integraciones entre bases de datos de distintas plataformas por medio de una herramienta web para la empresa Plus Projects S.A.**”, el cual incluye el estudio técnico, para lo cual ha implementado los conocimientos adquiridos en su calidad de alumno. b) Por iniciativa y responsabilidad del Instituto Superior Tecnológico Cordillera se desarrolla el estudio , motivo por el cual se regula de forma clara la cesión de los derechos de autor que genera la obra literaria y que es producto del proyecto de grado, el mismo que culminado es de plena aplicación técnica, administrativa y de reproducción.

SEGUNDA: CESIÓN Y TRANSFERENCIA.- Con el antecedente indicado, el Cedente libre y voluntariamente cede y transfiere de manera perpetua y gratuita

PERSONALIZACION Y EJECUCION DE INTEGRACIONES ENTRE BASES DE DATOS DE
DISTINTAS PLATAFORMAS, POR MEDIO DE UNA HERRAMIENTA WEB PARA LA EMPRESA
PLUS PROJECTS S.A.

todos los derechos patrimoniales del estudio de factibilidad descrito en la cláusula anterior a favor del Cesionario, sin reservarse para sí ningún privilegio especial (código fuente, código objeto, diagramas de flujo, planos, manuales de uso, etc.). El Cesionario podrá explotar el estudio de factibilidad por cualquier medio o procedimiento tal cual lo establece el Artículo 20 de la Ley de Propiedad Intelectual, esto es, realizar, autorizar o prohibir, entre otros: a) La reproducción del estudio de factibilidad por cualquier forma o procedimiento; b) La comunicación pública del estudio; c) La distribución pública de ejemplares o copias y la comercialización; d) Cualquier transformación o modificación en el estudio de factibilidad; e) La protección y registro en el IEPI del estudio de factibilidad a nombre del Cesionario; f) Ejercer la protección jurídica del estudio de factibilidad; g) Los demás derechos establecidos en la Ley de Propiedad Intelectual y otros cuerpos legales que normen sobre la cesión de derechos de autor y derechos patrimoniales.

TERCERA: OBLIGACIÓN DEL CEDENTE.- El cedente no podrá transferir a ningún tercero los derechos que conforman la estructura, secuencia y organización del estudio de factibilidad que es objeto del presente contrato, como tampoco emplearlo o utilizarlo a título personal, ya que siempre se deberá guardar la exclusividad del estudio de factibilidad a favor del Cesionario.

CUARTA: CUANTIA.- La cesión objeto del presente contrato, se realiza a título gratuito y por ende el Cesionario ni sus administradores deben cancelar valor alguno o regalías por este contrato y por los derechos que se derivan del mismo.

QUINTA: PLAZO.- La vigencia del presente contrato es indefinida.

SEXTA: DOMICILIO, JURISDICCIÓN Y COMPETENCIA.- Las partes fijan como su domicilio la ciudad de Quito. Toda controversia o diferencia derivada de

éste, será resuelta directamente entre las partes y, si esto no fuere factible, se solicitará la asistencia de un Mediador del Centro de Arbitraje y Mediación de la Cámara de Comercio de Quito. En el evento que el conflicto no fuere resuelto mediante este procedimiento, en el plazo de diez días calendario desde su inicio, pudiendo prorrogarse por mutuo acuerdo este plazo, las partes someterán sus controversias a la resolución de un árbitro, que se sujetará a lo dispuesto en la Ley de Arbitraje y Mediación, al Reglamento del Centro de Arbitraje y Mediación de la Cámara de comercio de Quito, y a las siguientes normas: a) El árbitro será seleccionado conforme a lo establecido en la Ley de Arbitraje y Mediación; b) Las partes renuncian a la jurisdicción ordinaria, se obligan a acatar el laudo arbitral y se comprometen a no interponer ningún tipo de recurso en contra del laudo arbitral; c) Para la ejecución de medidas cautelares, el árbitro está facultado para solicitar el auxilio de los funcionarios públicos, judiciales, policiales y administrativos, sin que sea necesario recurrir a juez ordinario alguno; d) El procedimiento será confidencial y en derecho; e) El lugar de arbitraje serán las instalaciones del centro de arbitraje y mediación de la Cámara de Comercio de Quito; f) El idioma del arbitraje será el español; y, g) La reconvenición, caso de haberla, seguirá los mismos procedimientos antes indicados para el juicio principal.

SÉPTIMA: ACEPTACIÓN.- Las partes contratantes aceptan el contenido del presente contrato, por ser hecho en seguridad de sus respectivos intereses.

En aceptación firman a los 24 días del mes de Octubre del dos mil trece.

f) _____

f) _____

C.C. N° 1715958318

Instituto Superior Tecnológico Cordillera

CEDENTE

CESIONARIO

INDICE DE CONTENIDO	Pág
DECLARACIÓN DE APROBACIÓN DE TUTOR Y LECTOR.....	I
DECLARATORIA.....	II
CONTRATO DE CESIÓN SOBRE DERECHOS PROPIEDAD INTELECTUAL....	II
INDICE DE CONTENIDO	V
INDICE DE TABLAS.....	VIII
INDICE DE FIGURAS	X
RESUMEN EJECUTIVO	XIII
ABSTRACT.....	XV
CAPITULO I. ANTECEDENTES	1
1.01 CONTEXTO	1
1.02 JUSTIFICACIÓN	2
1.03 DEFINICIÓN DEL PROBLEMA CENTRAL.....	3
CAPITULO II. ANÁLISIS DE INVOLUCRADOS	6
2.01 REQUERIMIENTOS.....	6
2.01.01 DESCRIPCIÓN DEL SISTEMA ACTUAL.....	6
2.01.02 VISIÓN Y ALCANCE.....	7
2.01.03 ENTREVISTAS	7
2.01.04 MATRIZ DE REQUERIMIENTOS	7
2.02 MAPEO DE INVOLUCRADOS	8
CAPITULO III. PROBLEMAS Y OBJETIVOS	9
3.01 ÁRBOL DE PROBLEMAS.....	9
3.02 ÁRBOL DE OBJETIVOS.....	10
3.03 DIAGRAMA DE CASOS DE USO.....	11
3.04 DIAGRAMAS DE REALIZACIÓN	12

3.05 DIAGRAMAS DE SECUENCIA DEL SISTEMA	22
3.06 ESPECIFICACIÓN DE CASOS DE USO	27
CAPITULO IV. ANÁLISIS DE ALTERNATIVAS	36
4.01 MATRIZ DE ANÁLISIS DE ALTERNATIVAS	36
4.02 MATRIZ DE ANÁLISIS DE IMPACTO DE LOS OBJETIVOS.	36
4.03 ESTÁNDARES PARA EL DISEÑO DE CLASES	36
4.04 DIAGRAMA DE CLASES	37
4.05 MODELO LÓGICO – FÍSICO	38
4.06 DIAGRAMA DE COMPONENTES	39
4.07 DIAGRAMA DE ESTRATEGIAS	41
4.08 MATRIZ DE MARCO LÓGICO	41
CAPITULO V. PROPUESTA	42
5.01 ESPECIFICACION DE ESTÁNDARES DE PROGRAMACIÓN	42
5.01.01 ESTÁNDARES PARA CLASES	42
5.01.02 ESTÁNDARES PARA FUNCIONES	43
5.01.03 ESTÁNDARES PARA VARIABLES	43
5.01.04 ESTÁNDARES PARA NOMBRES DE CONTROLES	44
5.01.05 ESTÁNDARES PARA NOMBRES DE ARCHIVOS	44
5.01.06 COMPONENTES HTML	45
5.01.07 VARIABLES GLOBALES	45
5.01.08 ESTÁNDARES DE BASE DE DATOS	46
5.01.09 TIPOS DE DATOS	46
5.01.10 TABLAS	47
5.01.11 CAMPOS	48
5.02 DISEÑO DE INTERFACES DE USUARIO	48

5.03 ESPECIFICACIÓN DE PRUEBAS DE UNIDAD.....	53
5.04 ESPECIFICACIÓN DE PRUEBAS DE ACEPTACIÓN	55
5.05 ESPECIFICACIÓN DE PRUEBAS DE CARGA	60
5.06 CONFIGURACIÓN DEL AMBIENTE MÍNIMA/IDEAL	62
CAPITULO VI. ASPECTOS ADMINISTRATIVOS.....	63
6.01 RECURSOS	63
6.02 PRESUPUESTO.....	63
6.03 CRONOGRAMA	63
CAPITULO VII. CONCLUSIONES Y RECOMENDACIONES	64
7.01 CONCLUSIONES	64
7.02 RECOMENDACIONES	65
ANEXOS	66
8.01 MANUAL TÉCNICO.....	79
8.01.01 INSTALACIÓN DE WAMP SERVER 2.4 (APACHE, PHP 5, MYSQL 5).....	79
8.02 SCRIPT BASE DE DATOS MYSQL	85
8.03 DICCIONARIO DE DATOS	95

INDICE DE TABLAS	Pág
TABLA N° 5: INGRESAR CREDENCIALES.	13
TABLA N° 6: ACTUALIZAR REGISTROS.	14
TABLA N° 7: CREAR REGISTRO.	15
TABLA N° 8: ELIMINAR REGISTRO.	16
TABLA N° 9: LISTAR REGISTROS.	17
TABLA N° 10: OBTENER DATOS FUENTE.	18
TABLA N° 11: OBTENER EJECUCIONES A INICIAR.	19
TABLA N° 12: GUARDAR DATOS AUDITORIA.	20
TABLA N° 13: OBTENER DATOS AUDITORIA.	21
TABLA N° 14: EJECUTAR PROCESOS.	22
TABLA N° 15: INGRESAR CREDENCIALES.	27
TABLA N° 16: ACTUALIZAR REGISTRÓ.	28
TABLA N° 17: CREAR REGISTRÓ.	29
TABLA N° 18: ELIMINAR REGISTRÓ.	30
TABLA N° 19: LISTAR REGISTROS.	30
TABLA N° 20: OBTENER EJECUCIONES A INICIAR.	31
TABLA N° 21: OBTENER DATOS FUENTE.	32
TABLA N° 22: GUARDAR DATOS AUDITORIA.	33
TABLA N° 23: OBTENER DATOS AUDITORIA.	34
TABLA N° 24: EJECUTAR PROCESOS.	35
TABLA N° 26: MATRIZ DE ANÁLISIS DE IMPACTO DE LOS OBJETIVOS.	36
TABLA N° 28: TIPOS DE DATOS.	47
TABLA N° 29: NOMBRE DE CAMPOS.	48
TABLA N° 30: PRUEBA DE UNIDAD 1.	53

TABLA N° 31: PRUEBA DE UNIDAD 2.....	53
TABLA N° 32: PRUEBA DE UNIDAD 3.....	54
TABLA N° 33: PRUEBA DE UNIDAD 4.....	54
TABLA N° 34: PRUEBA DE ACEPTACIÓN 1	55
TABLA N° 35: PRUEBA DE ACEPTACIÓN 2	56
TABLA N° 36: PRUEBA DE ACEPTACIÓN 3	57
TABLA N° 37: PRUEBA DE ACEPTACIÓN 4	58
TABLA N° 38: PRUEBA DE ACEPTACIÓN 5	59
TABLA N° 39: PRUEBA DE CARGA 1	60
TABLA N° 40: PRUEBA DE CARGA 2	61
TABLA N° 41: PRUEBA DE CARGA 2	62
TABLA N° 42: RECURSOS.....	63
TABLA N° 43: PRESUPUESTO	63
TABLA N° 1. MATRIZ DE FUERZAS T.	68
TABLA N° 2: ENTREVISTA WLADIMIR PAZMIÑO (DIRECTOR TECNOLÓGICO).....	69
TABLA N° 3: MATRIZ DE REQUERIMIENTOS.	71
TABLA N° 4: MATRIZ DE ANÁLISIS DE INVOLUCRADOS.....	72
TABLA N° 25: MATRIZ DE ANÁLISIS DE ALTERNATIVAS.....	74
TABLA N° 27: ANÁLISIS DEL MARCO LÓGICO.....	75
TABLA N° 27: COMPONENTES HTML.....	76
TABLA N° 33: PRESUPUESTO	77

INDICE DE FIGURAS	Pág
FIGURA 1. MAPA DE INVOLUCRADOS	8
FIGURA 2. ANÁLISIS DEL PROBLEMA	9
FIGURA 3. ANÁLISIS DE OBJETIVOS	10
FIGURA 4. DIAGRAMA GENERAL	11
FIGURA 5. DIAGRAMA DE MANTENIMIENTO	12
FIGURA 7. ACTUALIZAR REGISTRO	13
FIGURA 8. CREAR REGISTRO	14
FIGURA 9. ELIMINAR REGISTROS	15
FIGURA 10. LISTAR REGISTROS	16
FIGURA 11. OBTENER DATOS FUENTE	17
FIGURA 12. OBTENER EJECUCIONES A INICIAR	18
FIGURA 13. GUARDAR DATOS AUDITORIA	19
FIGURA 14. OBTENER DATOS AUDITORIA	20
FIGURA 15. EJECUTAR PROCESOS	21
FIGURA 16. CU001	22
FIGURA 17. CU001	23
FIGURA 18. CU002:	23
FIGURA 19. CU003:	24
FIGURA 20. CU004	24
FIGURA 21. CU005:	25
FIGURA 22. CU006:	25
FIGURA 23. CU007:	25
FIGURA 24. CU008	26
FIGURA 25. CU009	26

FIGURA 26. MODELO LÓGICO	38
FIGURA 27. MODELO FÍSICO	39
FIGURA 28. DIAGRAMA DE COMPONENTES.....	40
FIGURA 30. ESTÁNDAR FUNCIONES	43
FIGURA 31. ESTÁNDAR VARIABLES	44
FIGURA 32. ESTÁNDAR CONTROLES	44
FIGURA 33. ESTÁNDAR ARCHIVOS	45
FIGURA 34. VARIABLES GLOBALES	46
FIGURA 35. INICIO DE SESIÓN	48
FIGURA 36. LISTAR USUARIOS	49
FIGURA 37. CREAR USUARIO	49
FIGURA 38. LISTAR CONEXIONES	49
FIGURA 39. CREAR CONEXIÓN	50
FIGURA 40. LISTAR FUENTE DE DATOS	50
FIGURA 41. CREAR FUENTE DE DATOS	51
FIGURA 43. CREAR PROCESO.....	51
FIGURA45. CREAR EJECUCIÓN.....	52
FIGURA46. DIAGRAMA DE CLASES	67
FIGURA. 46 ADVERTENCIA DE SEGURIDAD.....	80
FIGURA. 47 INICIO DE INSTALACIÓN WAMP SERVER 2.4	80
FIGURA. 48 CONTRATO DE LICENCIA	81
FIGURA. 49 INSTALAR EN UBICACIÓN.....	81
FIGURA. 50 SELECCIONAR TAREAS ADICIONALES	82
FIGURA 51. LISTO PARA INSTALAR.....	82
FIGURA 52. INSTALANDO	83

FIGURA 53. PARÁMETROS DE EMAIL	83
FIGURA 54. INSTALACIÓN COMPLETA	84
FIGURA 55. ICONO PARA EJECUTAR WAMP SERVER	84
FIGURA 56. VENTANA DE ADMINISTRACIÓN DE SERVICIOS	85
FIGURA 57. ABRIR ARCHIVO DE CONFIGURACIÓN DE PHP	85
FIGURA 58. EXTENSIONES ACTIVAS	85

Resumen ejecutivo

El sistema informático Herramienta de integraciones fue creado para la personalización de integraciones de datos, el mismo se basa en diferentes módulos que permiten la configuración correcta de integraciones. Por medio de estos módulos se manejan las conexiones existentes que pueden realizarse a un servidor de base de datos que puede ser utilizado como fuente o como destino de la información.

La herramienta de integraciones consta de un módulo que permite la creación de sentencias SQL a partir de una interfaz de usuario gráfica, la cual permite seleccionar conexión, base de datos, tabla, filtros y campos a obtener, posteriormente esta fuente de datos será la encargada de entregar información a ser utilizada por procesos del tipo:

- Consultar
- Condicionar
- Actualizar
- Insertar
- Eliminar

Los procesos que utilizarán la información, son creados a partir de la fuente de datos, estos procesos son ejecutados por cada registro proveniente de la fuente de datos. La creación de procesos cuenta con interfaz de usuario que permite al desarrollador establecer conexión, base de datos, tabla, filtros y campos que van a ser insertados, actualizados, consultados, eliminados o que cumplan alguna condición, heredando la información proveniente de cada registro de la fuente de datos.

Los procesos pueden ser del tipo padre en cuyo caso heredan información de la fuente de datos y pueden ser del tipo hijo en cuyo caso heredan información de la fuente de datos y de los procesos predecesores.

La herramienta de integraciones también consta de un módulo de ejecuciones el cual permite la configuración de tareas programadas en un rango de fechas establecido.

Estas tareas programadas son las encargadas de iniciar cada ejecución, al obtener información de una fuente de datos y ejecutar todos sus procesos, al finalizar la ejecución se procede a revisar si se debe notificar resultados a algún actor de negocio, si es así se genera un archivo con los registros que hayan cumplido con las condiciones de notificación y el archivo será enviado por correo electrónico, finalizando de esta manera la ejecución y permitiendo al sistema continuar con su trabajo.

Abstract

The computer system integration tool was created for customization of data integration; it is based on different modules that allow the correct configuration of integrations. Through these modules the connections that can be made to a database server that can be used as a source or destination information is handled.

The tool consists of an integration module that allows creating SQL statements from a graphical user interface, which allows you to select connection, database, table, fields and filters to get, and then this data source will be responsible to deliver information to be used by processes such as:

- Check
- Adjust
- Update
- Insert
- Delete

The processes that use the information are created from the data source, these processes are executed for each record from the data source. The creation process has user interface that allows developers to connect, database, table, fields and filters that are to be inserted, updated, viewed, deleted or fulfilling any conditions, inheriting information from each record the data source .

Processes can be of the kind in which case parent inherit information from the data source and can be of the kind in which case child inherits information from the source data and processes predecessors.

The integration tool also includes module executions which allows configuration of scheduled tasks on a set date range.

These scheduled tasks are responsible for initiating each run, to get information from a data source and execute all processes at the end of the implementation are reviewed whether to notify an actor results business, if so it generates a file with records that have complied with the notification requirements and the file will be emailed, thus ending the execution and allowing the system to continue its work.

CAPITULO I. Antecedentes

1.01 Contexto

La empresa Plus Projects S.A. ubicada en la parroquia de Iñaquito en la calle Mariana de Jesús E7-8 y Pradera, edificio: Business Plus "La Pradera", oficina: 801, brinda el servicio de desarrollo de software, para diferentes empresas que requieran aplicaciones web de tipo digital, CRM y BPM.

La estructura funcional de la empresa es jerárquica conformada por Presidencia, Gerencia General, Dirección tecnológica, Dirección de diseño, Jefe web, Jefe CRM & BPM, diseñadores y desarrolladores.

Para el desarrollo de software especialmente en el área de CRM, es necesario realizar una comunicación constante de datos entre las empresas contratantes y el sistema CRM, esta comunicación se la realiza a través de integraciones de datos las cuales aseguran el flujo correcto de información y ubican la misma en la base de datos estructurada del sistema CRM.

El sistema CRM al ser requerido por varias empresas y como cada una de estas tiene un diferente campo de acción y por lo tanto un diferente flujo de trabajo, necesita que su información se maneje en un flujo de datos que permita visualizar la información correctamente. Este flujo de datos es implementado como un desarrollo complementario del sistema CRM, el cual permite que los registros pasen por varios procesos, antes de ser ingresados en la base de datos final.

Estos desarrollos para integraciones de datos, permiten a las empresas solicitantes tener un reporte constante de los registros que cumplen o no cumplen algún requerimiento de negocio importante para mantener la integridad de los datos.

El conflicto que presenta la empresa es que los desarrollos de integraciones de datos, antes expuestos, utilizan recursos humanos y de factor tiempo, para que puedan ser realizados de forma correcta y conveniente para las empresas contratantes.

Al no disponer de una herramienta que permita prescindir del proceso de desarrollo de integraciones de datos, la empresa debe asignar recursos humanos y tiempo en el cronograma de trabajo, este tiempo contemplado retarda la fecha de entrega del software, lo cual es una desventaja en las licitaciones en las que se compite con otras empresas que realizan la misma actividad económica.

Si no se realiza este proyecto la empresa continuara utilizando innecesariamente mucho tiempo y recursos en el desarrollo de integraciones de datos, lo que a corto o largo plazo provocara inconformidad en los nuevos o actuales clientes.

1.02 Justificación

Actualmente la empresa no cuenta con un sistema que permita genera integraciones de datos de una forma práctica, estandarizada y rápida, esto tiene como consecuencia un abundante tiempo de entrega de proyectos, además de que no se maneja una estructura establecida que facilite la modificación y creación de nuevos requerimientos para todo el personal del área de desarrollo.

Por lo general las integraciones de datos utilizan 3 bases de datos para su funcionamiento, siendo estas:

Base de datos fuente.

Base de datos para auditoria

Base de datos destino.

Los procesos que deben ser desarrollados para utilizar estas bases de datos y realizar un flujo de datos correcto según los requerimientos de negocio de las empresas contratantes son largos y repetitivos, además de que no garantizan optimización de recursos.

Para la empresa sería beneficioso contar con un sistema el cual permita reducir el tiempo, recursos humanos y por lo tanto disminuir el gasto empleado en cada desarrollo. Al tener clientes diferentes y con requerimientos diferentes no es posible prescindir de todas las tareas, sin embargo es posible prescindir de las tareas más comunes al automatizarlas, permitiendo de esta manera finalizar proyectos en menor tiempo, sin acumulación de trabajo al cumplir fechas establecidas en el cronograma y aumentando el tiempo de vida del software final y con valor agregado.

Los beneficiarios principales de este proyecto son los programadores, ya que al automatizar el proceso de integración de datos, se podrá utilizar el tiempo en cumplir con los requerimientos principales y sin tener sobrecarga de trabajo.

Adicional a esto la empresa en su parte económica resultaría favorecida, ya que sería una ventaja estratégica en función de tiempo y dinero, que permitiría obtener más clientes y ofrecerles un mejor producto.

1.03 Definición del problema central.

Ver Anexo A.02

5 = ALTO

4 = MEDIO ALTO

3 = MEDIO

2 = MEDIO BAJO

1 = BAJO

I = Intensidad: Cual es el nivel de impacto de la fuerza sobre la problemática actual.

PC = Potencial de cambio: ¿Cuánto se puede modificar o aprovechar la fuerza para llegar a la situación deseada?

Para la realización del proyecto es necesario hacer un análisis que involucre la situación actual, la situación mejorada y empeorada, de esta manera se puede determinar las fuerzas impulsadoras y bloqueadoras, para posteriormente analizar cuanto pueden ser modificadas para alcanzar el objetivo.

Manejo de estándares que permitirán el desarrollo confiable de integraciones y facilidad en el soporte técnico, tiene un nivel medio bajo en la situación actual y será modificado hasta un nivel alto al implementar una herramienta estandarizada y fácil de usar.

Prescindir del desarrollo de integraciones y disminuir el tiempo de entrega, tiene un nivel bajo en la situación actual y será modificado hasta un nivel alto al implementar una herramienta configurable ya desarrollada.

Facilidad en la implementación de nuevos requerimientos, y rápida reestructuración de procesos, tiene un nivel medio y será modificado hasta un nivel alto al implementar una herramienta de rápida configuración.

Rapidez y facilidad en el desarrollo, menor cantidad de empleados involucrados, tiene un nivel medio bajo y será modificado hasta un nivel alto al implementar una herramienta de rápida configuración que solo requiera de un empleado.

Eficaz creación de notificaciones y reportes disponibles para los clientes, cuando lo necesiten, tiene un nivel bajo y será modificado hasta un nivel alto al implementar una herramienta que permita crear las notificaciones que se desean realizar, al finalizar cada integración de datos y adjuntar los registros resultantes en archivos CSV.

CAPITULO II. Análisis de involucrados

2.01 Requerimientos

2.01.01 Descripción del sistema actual

La empresa Plus Projects S.A. dedicada al desarrollo de aplicaciones web y personalización de herramientas CRM & BPM, tiene la necesidad de crear un proceso de integración de datos diferente para cada cliente que contrata sus servicios, ya que los mismos tienen diferentes lógicas de negocio y el ingreso de información depende de la misma.

La empresa designa tiempo y recursos en el cronograma de trabajo del proyecto, para la programación y pruebas de funcionamiento de estos procesos de integración.

Por lo general el proceso para la creación de estas integraciones de datos, sigue el siguiente flujo de información.

- Conexión a servidor de base de datos (Cliente)
- Recuperar información a integrar a los sistemas CRM.
- Almacenar información en una base de datos para auditoría.
- Guardar, actualizar, eliminar información según la lógica de negocio en los sistemas CRM.
- Notificar los registros que no cumplen las condiciones del negocio.

Todos estos procesos se los programan desde el inicio en la mayoría de los casos.

2.01.02 Visión y alcance

Por medio de la creación de un sistema de fácil instalación que permita la configuración de conexiones a servidores, selección de bases de datos, construcción de sentencias SQL, condiciones para la ejecución de las sentencias SQL y notificaciones dependientes de valores booleanos resultantes de la ejecución de procesos, se podría disminuir de forma notable el tiempo de entrega de los proyectos de la empresa, al prescindir del diseño y desarrollo de una integración de datos programada, además mantendría una misma estructura para cada proceso de integración permitiendo seguir un estándar que permita mejorar los procesos o facilite la solución de problemas.

2.01.03 Entrevistas

Ver Anexo A.03

2.01.04 Matriz de requerimientos

Ver Anexo A.04

2.02 Mapeo de involucrados

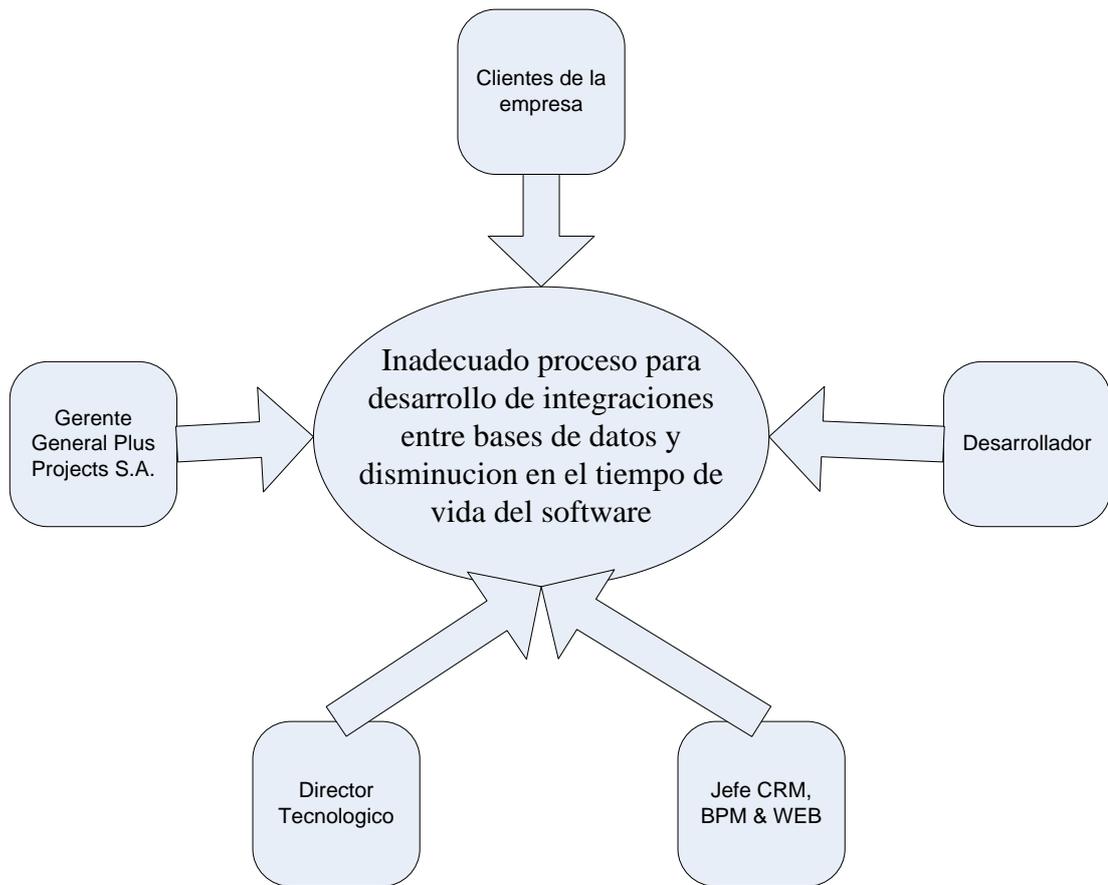


Figura 1. Mapa de involucrados: Detalla de manera visual todos los actores que influyen directa o indirectamente para lograr nuestro objetivo.

2.03 Matriz de análisis de involucrados.

Ver Anexo A.05

CAPITULO III. Problemas y objetivos

3.01 Árbol de problemas.

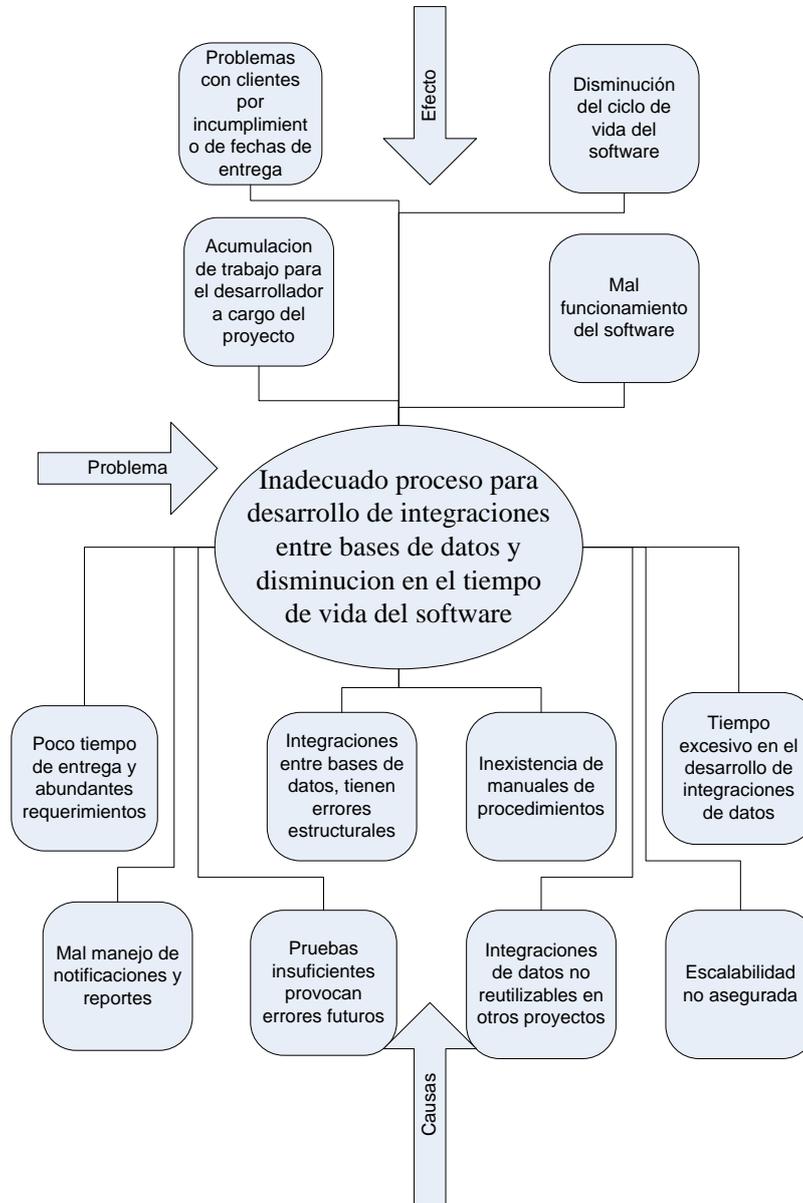


Figura 2. Análisis del problema: Realiza un análisis de las posibles causas y efectos que influyen en el problema central.

Existen varias causas que inciden directa o indirectamente en el inadecuado desarrollo de integraciones de datos, el poco tiempo de entrega de los proyectos da como resultado un producto con debilidades a nivel de integraciones de datos, ya

que se las desarrolla sin pensar en posibles cambios en la lógica de negocio y dejando obsoleto en poco tiempo el proceso de integración.

3.02 Árbol de objetivos.

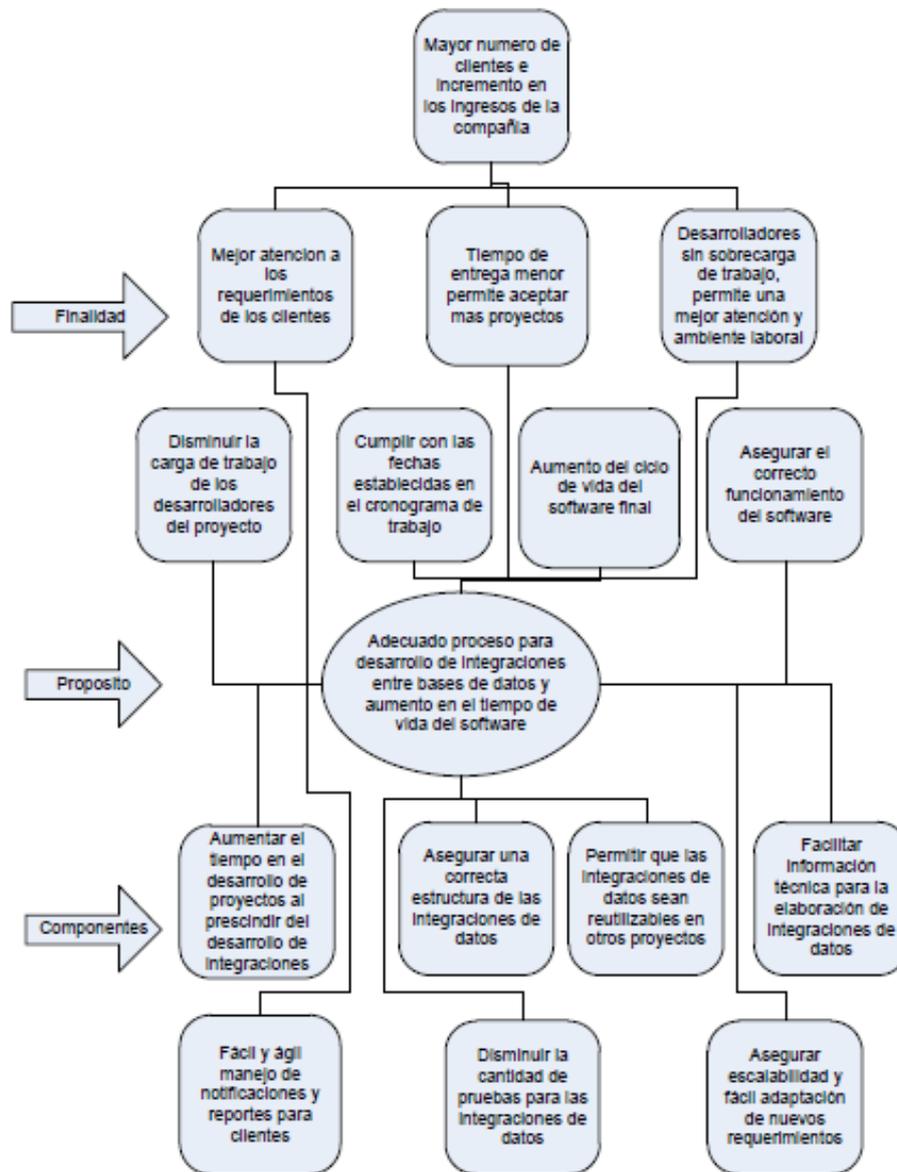


Figura 3. Análisis de objetivos: Listamos los componentes que ayudaran a cumplir el propósito planteado y los resultados esperados para el proyecto.

Al aumentar el tiempo de entrega de los proyectos y encontrar un nuevo proceso de creación de integraciones de datos, se tendrá como finalidad mejorar la atención a los

otros requerimientos de los clientes, mejorar o cumplir el tiempo de entrega de proyectos y evitar la sobrecarga y agotamiento del personal del área de desarrollo.

3.03 Diagrama de casos de uso.

El Diagrama de casos de uso muestra un conjunto de actividades, actores implicados y sus relaciones. Este diagrama es fundamental ya que nos ayuda a visualizar el funcionamiento en el modelado y organización del sistema.

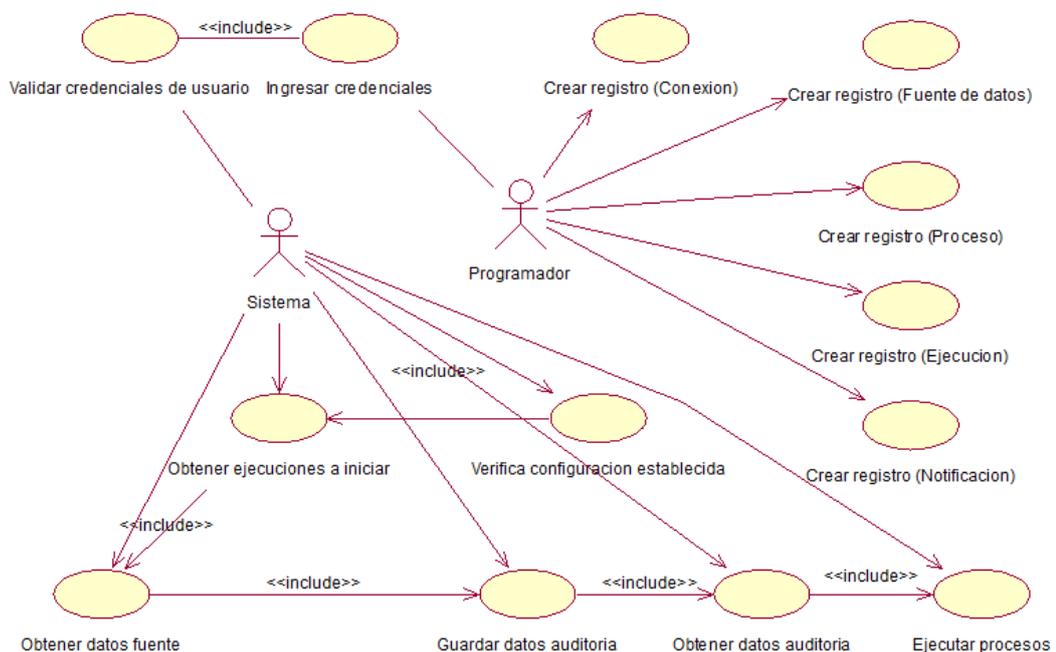


Figura 4. Diagrama general: Este diagrama ilustra los actores involucrados y los casos de uso que estos actores realizan para el funcionamiento del sistema.

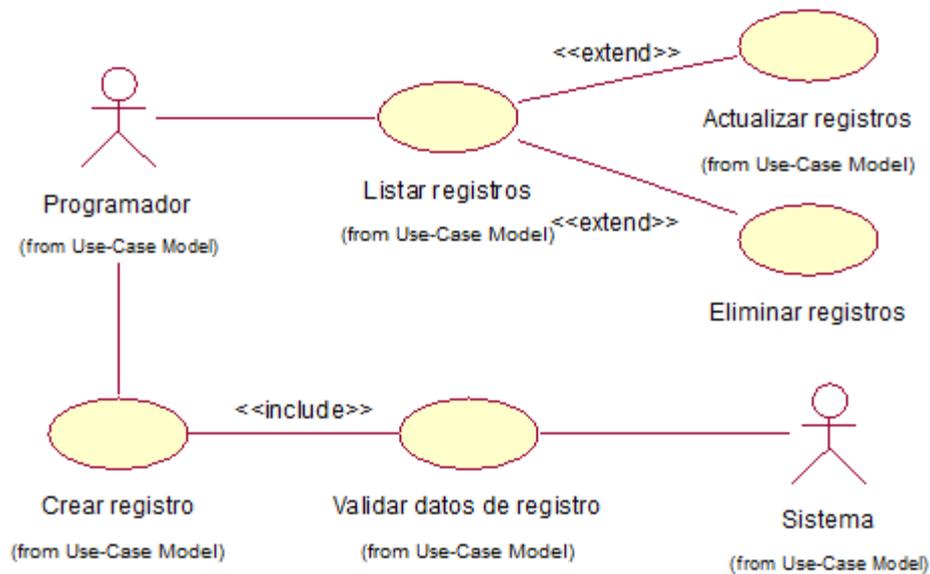


Figura 5. Diagrama de mantenimiento: Este diagrama ilustra los actores involucrados y los casos de uso que estos actores realizan para el mantenimiento de tablas del sistema.

3.04 Diagramas de Realización

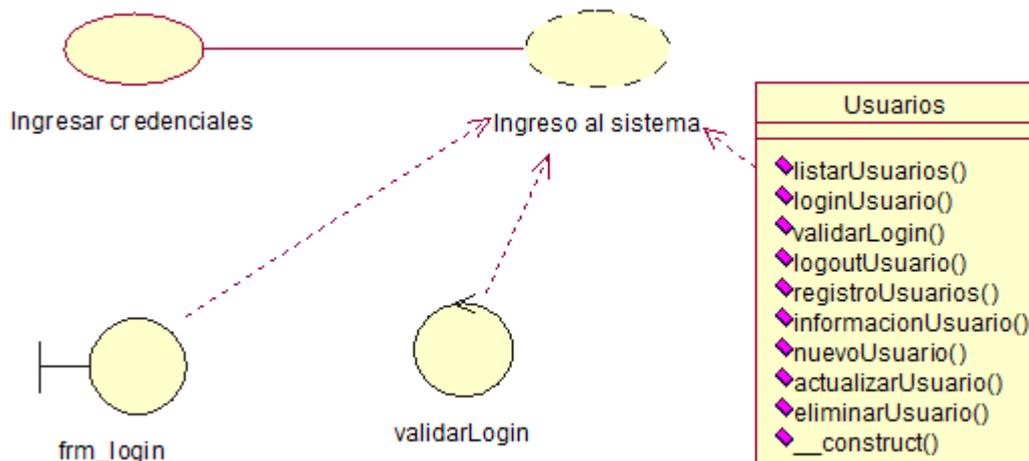


Figura 6. Ingresar credenciales: Para ingresar al sistema el usuario deberá ingresar sus credenciales, para posteriormente realizar una validación de los datos y permitir o negar el ingreso al sistema.

Tabla N° 5: Ingresar credenciales.

ID	CU001
Nombre	Ingresar credenciales
Actores	Programador, Sistema
Precondiciones:	1. Controlar el ingreso al sistema para usuarios registrados
Flujo de Eventos:	1. Ingresar datos 2. Verificación de credenciales 3. Permitir o negar acceso

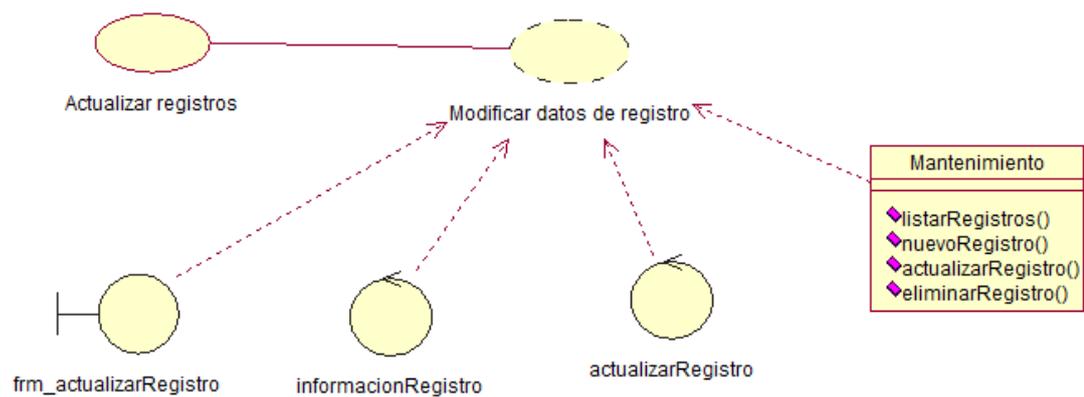


Figura 7. Actualizar registro: Para actualizar un registro se deberá seleccionarlo y posteriormente modificar la información que se desea actualizar, luego se realizara una validación de campos que definirán si el registro puede o no ser actualizado.

Tabla N° 6: Actualizar registros.

ID	CU002
Nombre	Actualizar registros
Actores	Programador, Sistema
Precondiciones:	1. Permitir la actualización de los datos de registros existentes
Flujo de	1. Obtener y mostrar datos de registros
Eventos:	2. Modificar y enviar información modificada
	3. Actualizar información de registros
	4. Notificar actualización exitosa.
Flujo	1. Notificar que el registro no se pudo actualizar
Alternativo.	
Pos condiciones.	1. El flujo alternativo entrara en funcionalidad en caso de alguna contrariedad durante el proceso.

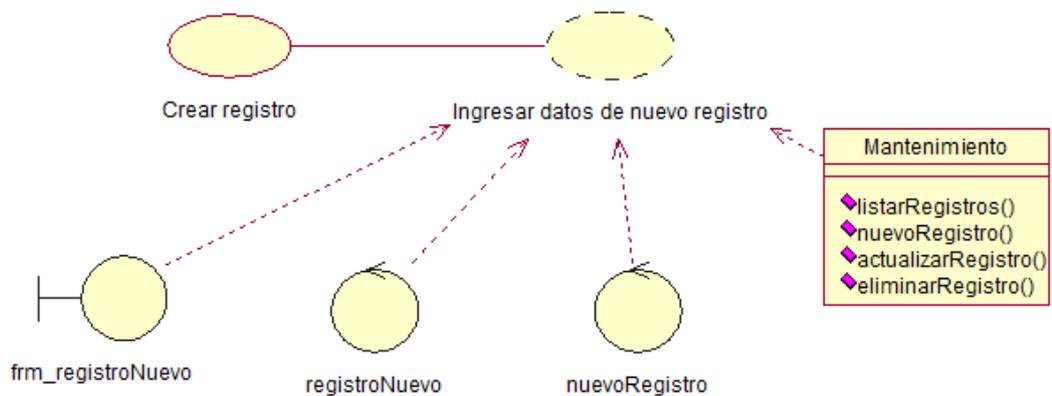


Figura 8. Crear registro: Para crear un registro se deberá ingresar los datos del registro que se desea crear, luego se realizara una validación de campos que definirán si el registro puede o no ser creado.

Tabla N° 7: Crear registró.

ID	CU003
Nombre	Crear registro
Actores	Programador, Sistema
Precondiciones:	1. Permitir la creación de nuevos registros.
Flujo de	1. Ingresar y enviar información de registro nuevo
Eventos:	2. Registrar nuevo en tabla solicitada
	3. Notificar registro exitoso
Flujo	1. Notificar que el registro no se pudo crear
Alternativo.	
Pos condiciones.	1. El flujo alternativo entrara en funcionalidad en caso de alguna contrariedad durante el proceso.

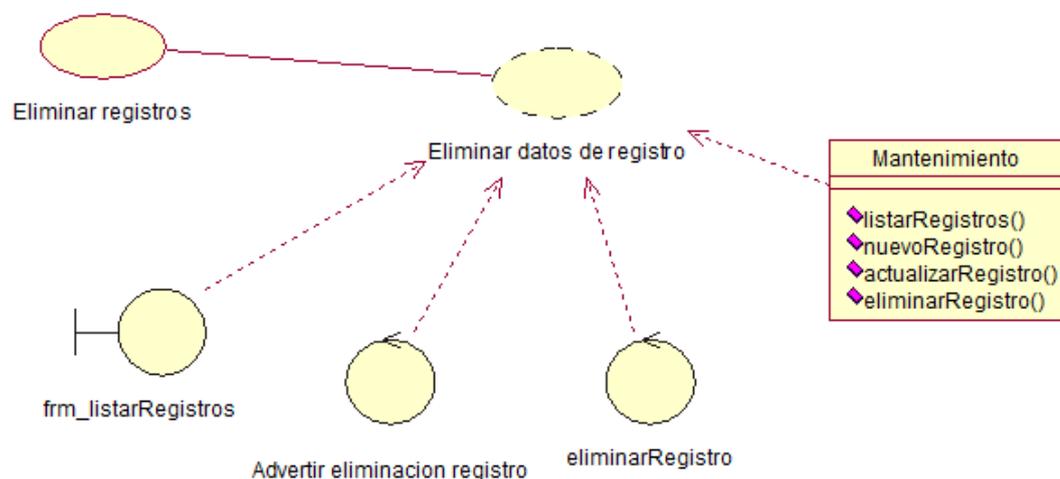


Figura 9. Eliminar registros: Para eliminar un registro se deberá seleccionarlo, luego se realizara una comprobación de si en realidad se desea eliminar, esto definirá si el registro puede o no ser eliminado.

Tabla N° 8: Eliminar registró.

ID	CU004
Nombre	Eliminar registro
Actores	Programador, Sistema
Precondiciones:	1. Permitir la eliminación de registros.
Flujo de	1. Seleccionar registro a eliminar
Eventos:	2. Confirmar eliminación y enviar información
	3. Eliminar registro
	4. Notificar eliminación exitosa
Flujo	1. Notificar que el registro no pudo ser eliminado
Alternativo.	
Pos condiciones.	1. El flujo alternativo entrara en funcionalidad en caso de alguna contrariedad durante el proceso.

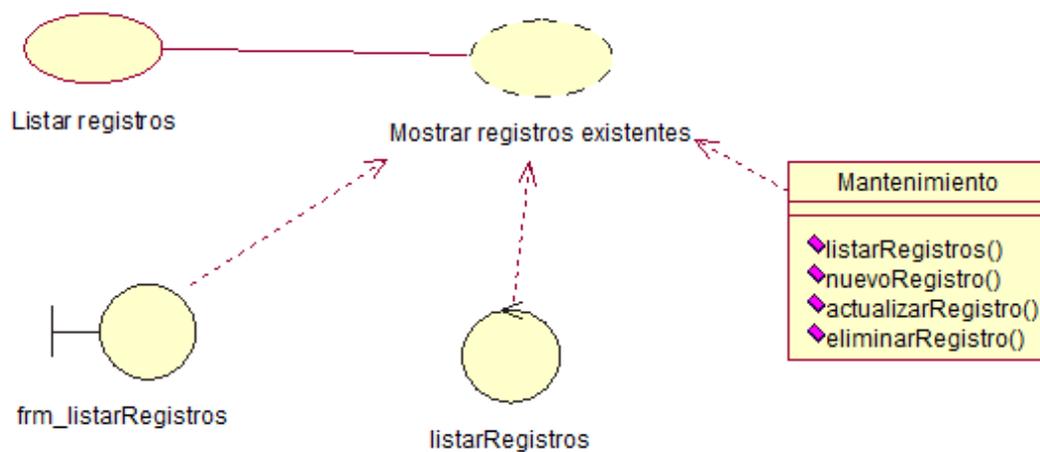


Figura 10. Listar registros: Para listar registros se deberá seleccionar la opción, se verificara si existen registros y se mostrara la información.

Tabla N° 9: Listar registros.

ID	CU005
Nombre	Listar registros
Actores	Programador, Sistema
Precondiciones:	<ol style="list-style-type: none"> 1. Permitir la visualización de los registros existentes 2. Permitir la selección para actualización o eliminación
Flujo de	1. Seleccionar listar registros.
Eventos:	2. Seleccionar registros a actualizar o eliminar
Flujo	3. Notificar que no hay registros creados
Alternativo.	
Pos condiciones.	1. El flujo alternativo entrara en funcionalidad en caso de alguna contrariedad durante el proceso.

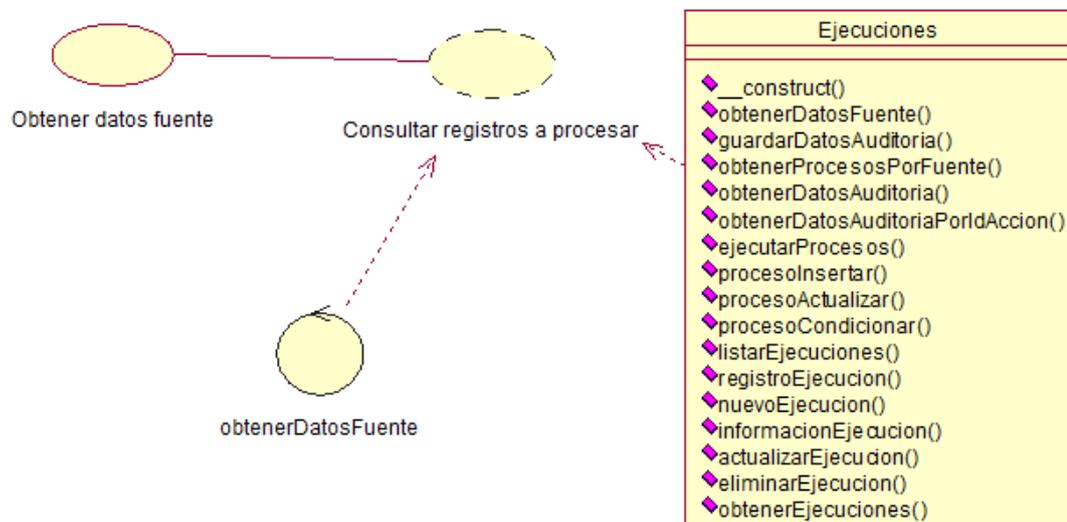


Figura 11. Obtener datos fuente: Para obtener los datos fuente a ser procesados, se consultaran las configuraciones de fuente de datos y se ejecutara el query respectivo.

Tabla N° 10: Obtener datos fuente.

ID	CU006
Nombre	Obtener datos fuente
Actores	Sistema
Precondiciones:	1. Debe obtener la información con las condiciones y preferencias establecidas
Flujo de	1. Consultar condiciones y preferencias
Eventos:	2. Construir el query de consulta y obtener información
Flujo	1. Notificar que no se pudo obtener la información
Alternativo.	
Pos condiciones.	1. El flujo alternativo entrara en funcionalidad en caso de alguna contrariedad durante el proceso.

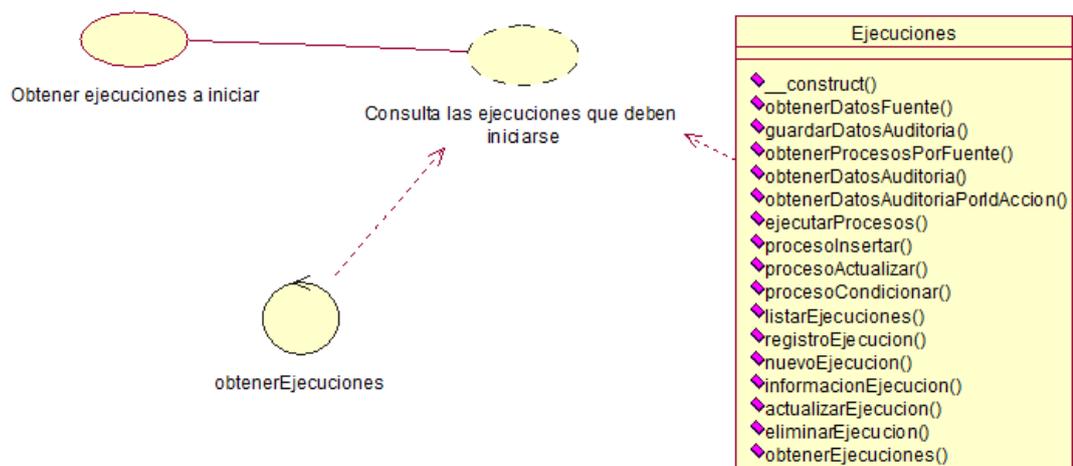


Figura 12. Obtener ejecuciones a iniciar: Consultara si existen ejecuciones a ser iniciadas en un determinado intervalo de tiempo

Tabla N° 11: Obtener ejecuciones a iniciar.

ID	CU007
Nombre	Obtener ejecuciones a iniciar
Actores	Sistema
Precondiciones:	1. Permitir la consulta de ejecuciones que deben ser iniciadas según el intervalo de tiempo establecido
Flujo de	1. Obtener hora y fecha del sistema
Eventos:	2. Verificar si existen ejecuciones programadas para la hora y fecha actual o anterior
Flujo	1. Notificar que no existen ejecuciones y finalizar el proceso
Alternativo.	
Pos condiciones.	1. El flujo alternativo entrara en funcionalidad en caso de alguna contrariedad durante el proceso.

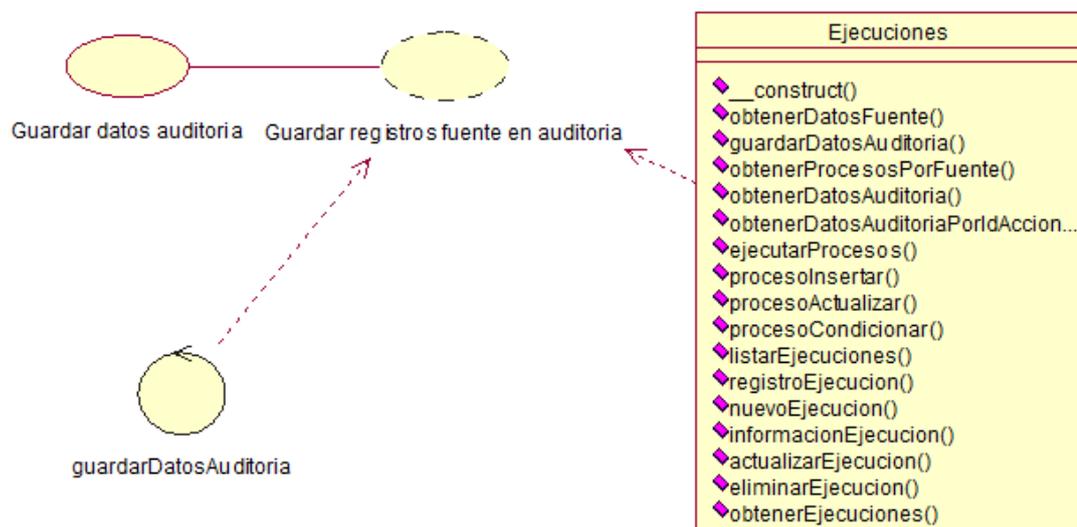


Figura 13. Guardar datos auditoria: Divide los registros obtenidos de la fuente de datos en campo=>valor y registra el ingreso de cada fila en las tablas auditoria_header y auditoria_action.

Tabla N° 12: Guardar datos auditoria.

ID	CU008
Nombre	Guardar datos auditoria
Actores	Sistema
Precondiciones:	1. Permitir que se registre la información entrante a ser integrada
Flujo de	1. Registrar el ingreso del registro, con estado 0
Eventos:	2. Dividir el registro en campo y valor
	3. Almacenar el campo y el valor como un registro en auditoria_action
Flujo	1. Notificar que no hay registros a guardar y finalizar el proceso
Alternativo.	
Pos condiciones.	1. El flujo alternativo entrara en funcionalidad en caso de alguna contrariedad durante el proceso.

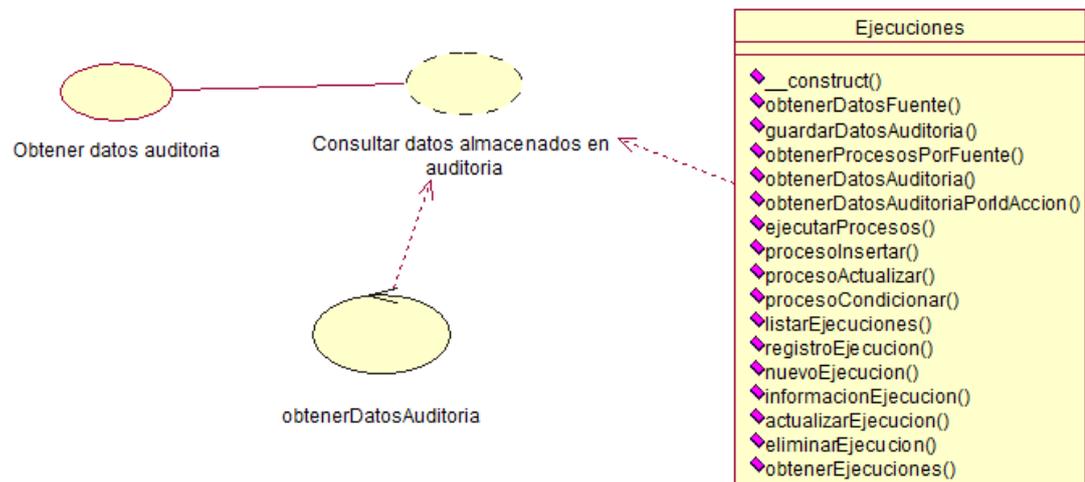


Figura 14. Obtener datos auditoria: Obtener registros ingresados en la base de datos de auditoría y convertir en objetos cada registro, preparándolo para el proceso.

Tabla N° 13: Obtener datos auditoria.

ID	CU009
Nombre	Obtener datos auditoria
Actores	Sistema
Precondiciones:	<ol style="list-style-type: none"> 1. Permitir la consulta de cada registro a ser procesado 2. Establecer un objeto de clase con los datos del registro obtenido
Flujo de	1. Consulta registros con estado igual a 0 en tabla auditoria_cabecera
Eventos:	2. Consultar datos almacenados del registro en tabla auditoria_detalle
Flujo	1. Notificar que no hay registros a procesar y finalizar el proceso
Alternativo.	
Pos condiciones.	1. El flujo alternativo entrara en funcionalidad en caso de alguna contrariedad durante el proceso.

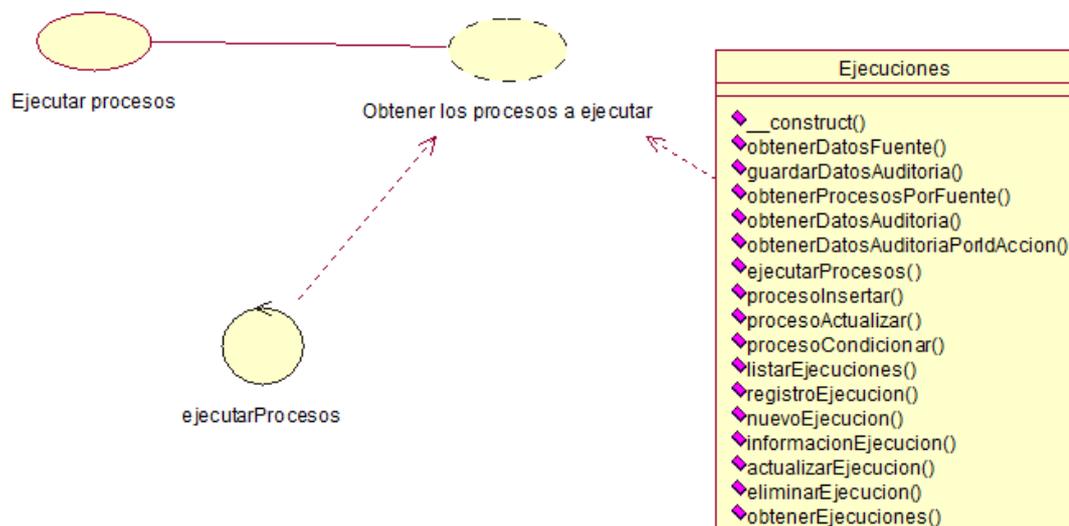


Figura 15. Ejecutar procesos: Obtiene los procesos por los que debe pasar cada registro, realiza inserciones, actualizaciones, selecciones o condiciones según se haya establecido anteriormente con cada registro existente en auditoria_cabecera.

Tabla N° 14: Ejecutar procesos.

ID	CU010
Nombre	Ejecutar procesos
Actores	Sistema
Precondiciones:	1. Permitir la ejecución de los procesos pre configurados en la herramienta
Flujo de	1. Obtener información de procesos a ejecutar
Eventos:	2. Ejecutar proceso por cada registro obtenido de la fuente de datos y que aun no fue procesado 3. Registrar resultados en tabla auditoria_procesos
Flujo	1. Notificar que no existen procesos a ser ejecutados y finalizar el proceso
Alternativo.	general
Pos condiciones.	1. El flujo alternativo entrara en funcionalidad en caso de alguna contrariedad durante el proceso.

3.05 Diagramas de Secuencia del sistema

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso de realización.

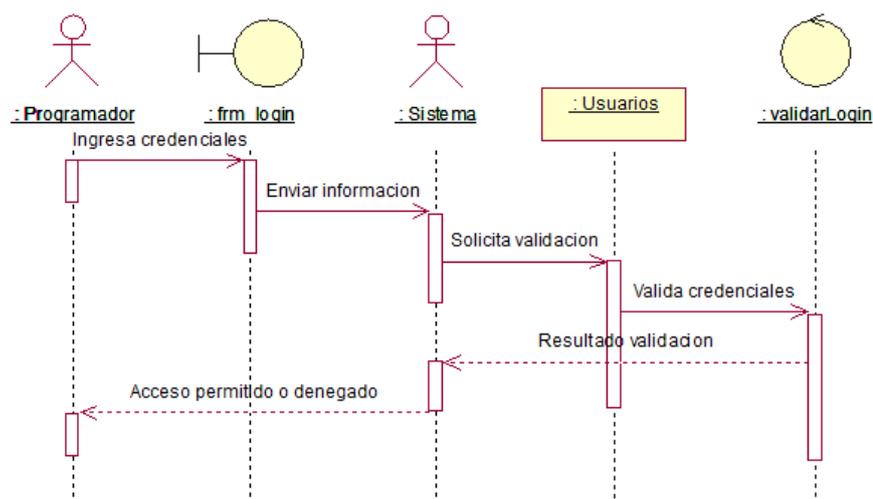


Figura 16. CU001:

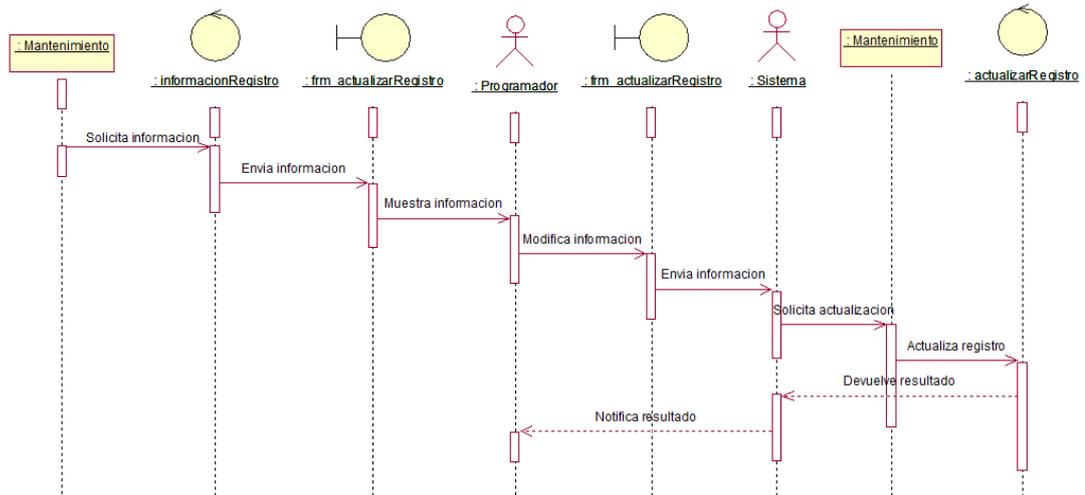


Figura 17. CU001:

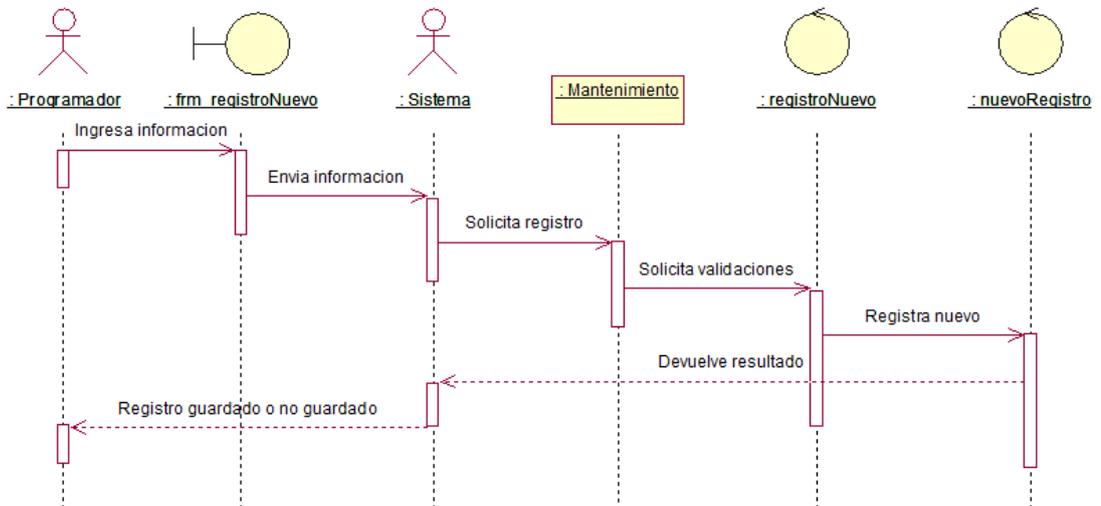


Figura 18. CU002:

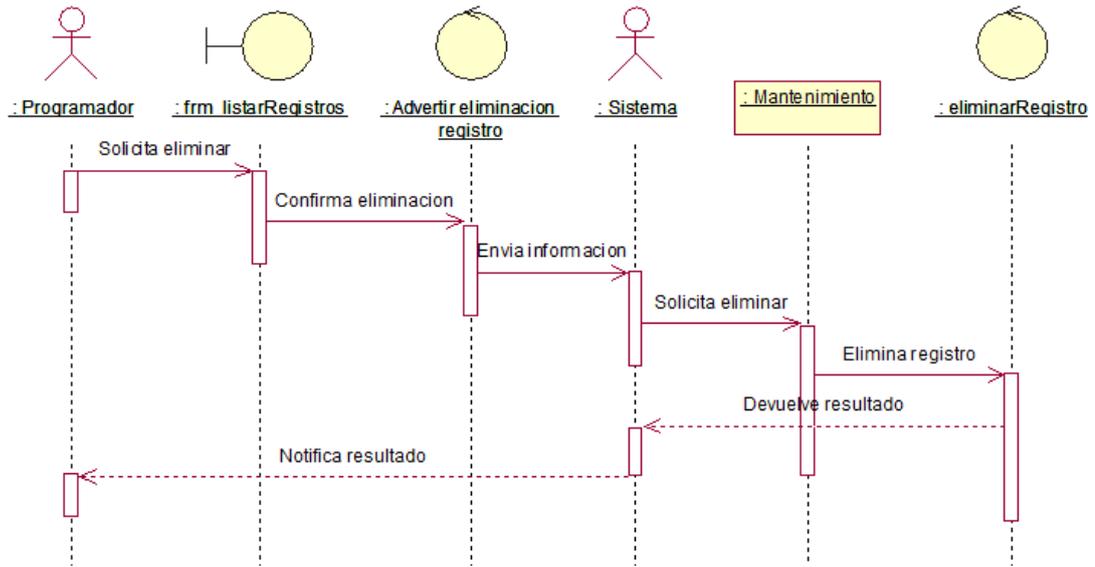


Figura 19. CU003:

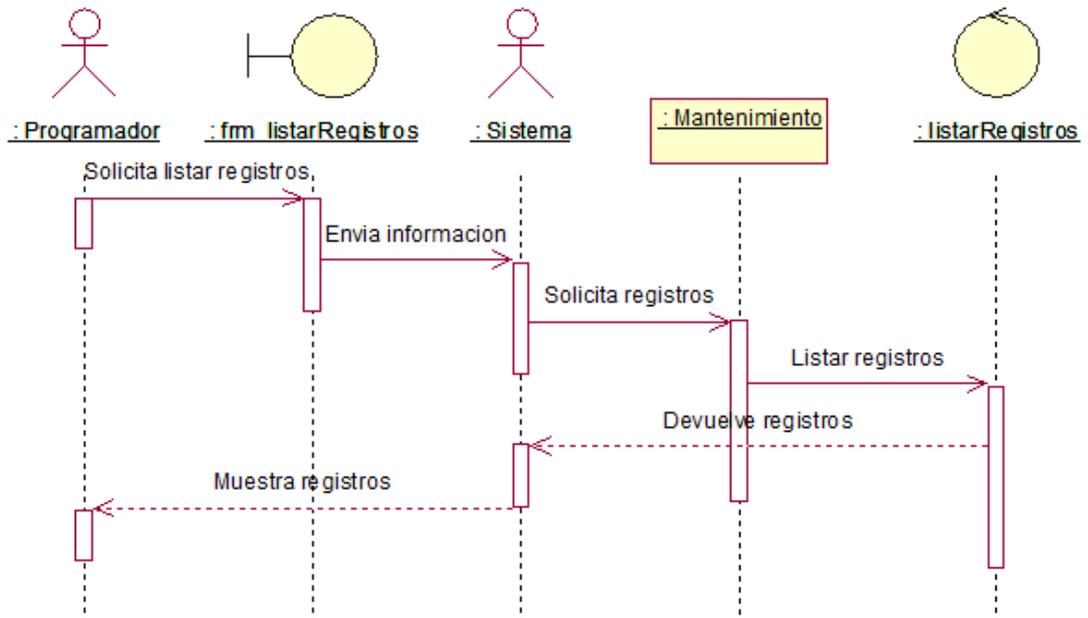


Figura 20. CU004:

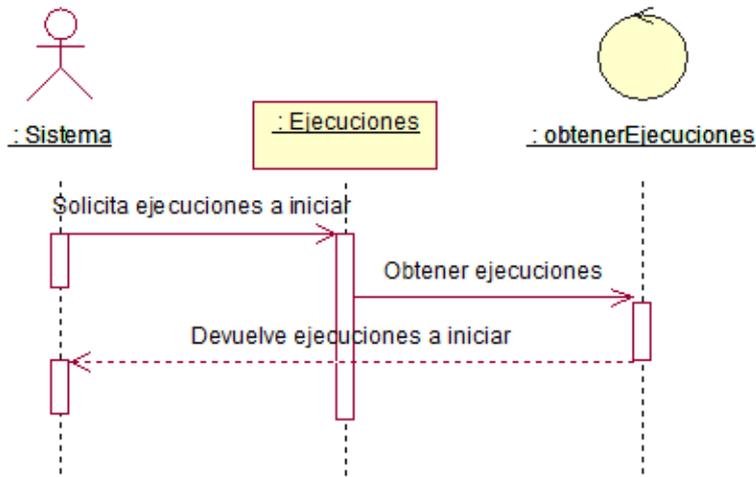


Figura 21. CU005:

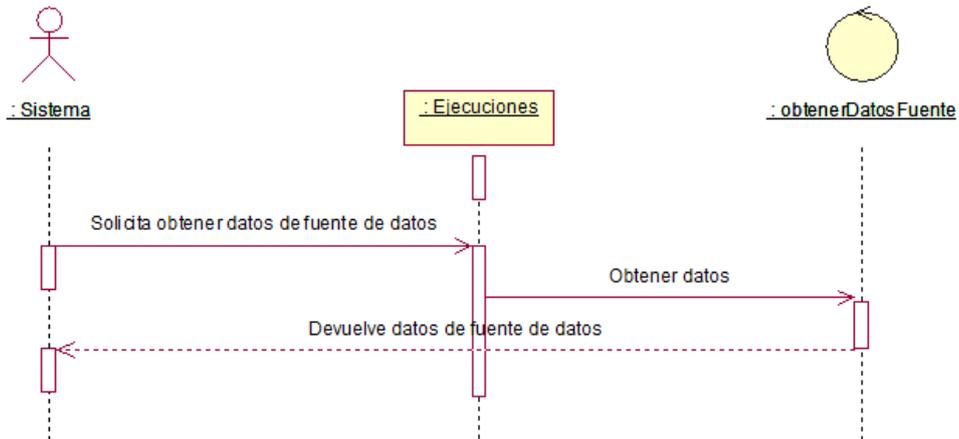


Figura 22. CU006:

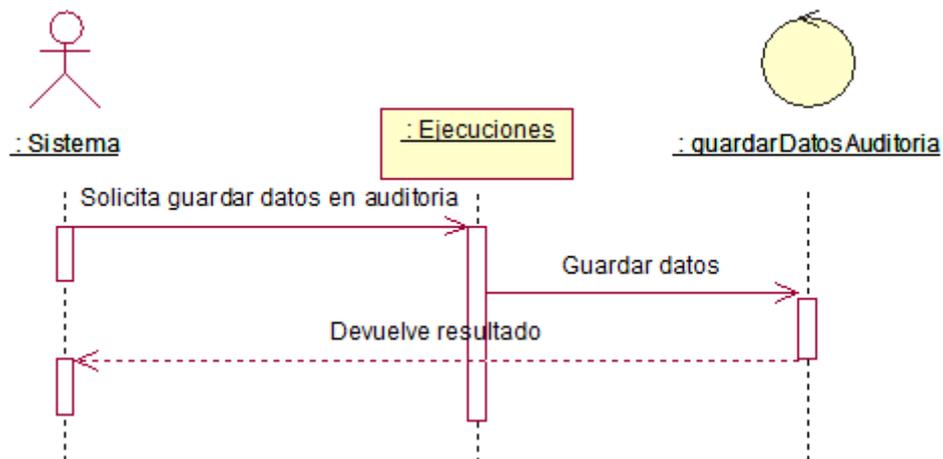


Figura 23. CU007:

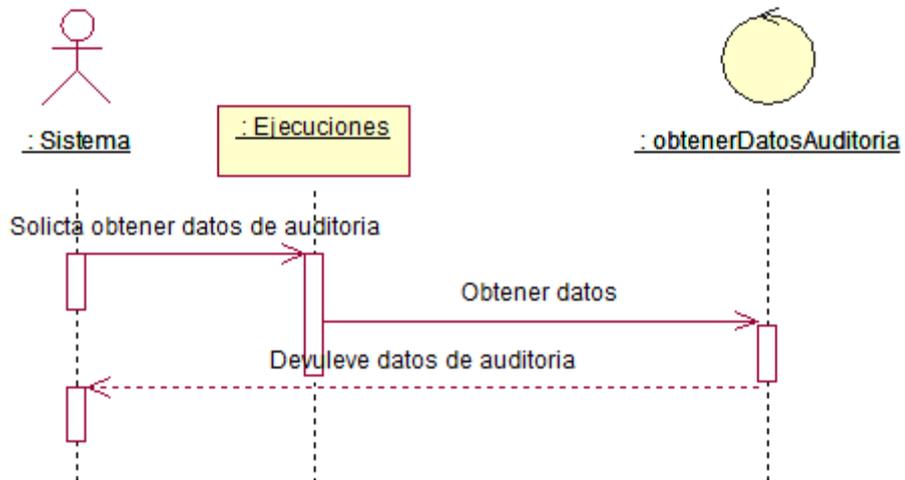


Figura 24. CU008:

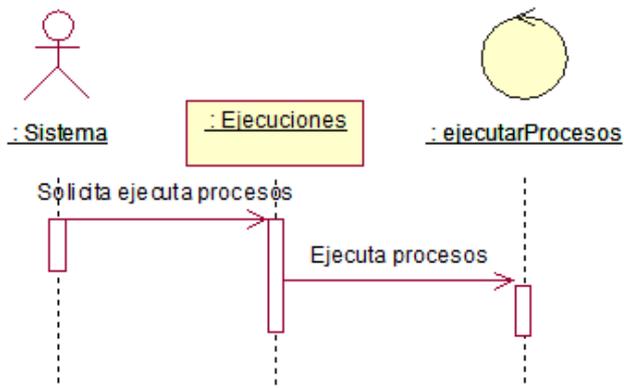


Figura 25. CU009:

3.06 Especificación de casos de uso

Tabla N° 15: Ingresar credenciales.

Caso de uso	Ingresar credenciales
Identificado	Módulo de seguridad que permite o niega el ingreso al sistema y a sus
r	funcionalidades
Curso típico de eventos	1) Ingresar credenciales
	2) Valida credenciales
	3) Permite o niega el ingreso al usuario
Usuario	Programador
Actividad	1) Ingresar nombre de usuario
que hace el	2) Ingresar contraseña
usuario con	3) Presiona botón ingresar
identificado	
r de orden	
Actividad	1) Verificar que el usuario y contraseña sean correctos
que hace el	2) Permitir el ingreso al sistema
usuario con	
identificado	
r de orden	
Cursos alternativos	
	Notificar que las credenciales son incorrectas
	Cerrar sesión de usuario

Tabla N° 16: Actualizar registró.

Caso de uso	Actualizar registro
Identificador	Permite la actualización del registro de una tabla al ser previamente seleccionado
Curso típico de eventos	1) Actualizar el registro seleccionado
Usuario	Programador
Actividad que hace el usuario con identificador de orden	<ol style="list-style-type: none"> 1) Selecciona registro a actualizar 2) Modifica la información que desea actualizar 3) Presiona el botón actualizar
Actividad que hace el usuario con identificador de orden	<ol style="list-style-type: none"> 1) Obtiene información actual del registro. 2) Verifica que la información este completa 3) Actualiza la información del registro
Cursos alternativos	
	Notifica que el registro no se actualizo

Tabla N° 17: Crear registró.

Caso de uso	Crear registro
Identificador	Permitir la creación de nuevos registros en alguna tabla
Curso típico de eventos	1) Crear nuevo registro en alguna tabla
Usuario	Programador
Actividad que hace el usuario con identificador de orden	<ol style="list-style-type: none"> 1) Presiona botón Nuevo. 2) Ingresa información requerida. 3) Presiona el botón Guardar.
Actividad que hace el usuario con identificador de orden	<ol style="list-style-type: none"> 1) Obtiene información ingresada. 2) Verifica calidad de información 3) Realiza validaciones de formulario 4) Guarda la información.
Cursos alternativos	
Notifica que el registro no se creo	

Tabla N° 18: Eliminar registró.

Caso de uso	Eliminar registro
Identificador	Permitir la eliminación de registros al seleccionarlos previamente
Curso típico de eventos	1) Eliminar registro seleccionado
Usuario	Programador
Actividad que hace el usuario con identificador de orden	1) Selecciona el registro a eliminar 2) Presiona el botón eliminar
Actividad que hace el usuario con identificador de orden	1) Realiza confirmación de eliminación. 2) Elimina el registro. 3) Elimina relaciones existentes con el registro.
Cursos alternativos	
Notifica que el registro no se eliminó.	

Tabla N° 19: Listar registros.

Caso de uso	Listar registros
Identificador	Permitir la visualización en lista de todos los registros de alguna tabla
Curso típico de eventos	1) Seleccionar la sección del menú a la cual se desea ingresar y ver los registros.
Usuario	Programador
Actividad que hace el usuario con identificador de orden	1) Selecciona opción en menú principal.
Actividad que hace el usuario con identificador de orden	1) Verifica si existen registros en la tabla seleccionada. 2) Muestra la información de todos los registros
Cursos alternativos	
Notifica que no existen registros a listar.	

Tabla N° 20: Obtener ejecuciones a iniciar.

Caso de uso	Obtener ejecuciones a iniciar
Identificador	Debe verificar si existen ejecuciones planificadas en un intervalo de tiempo establecido entre la fecha y hora actual o anterior
Curso típico de eventos	1) Consultar información de ejecuciones
Usuario	Sistema
Actividad que hace el usuario con identificador de orden	<ol style="list-style-type: none"> 1) Consulta tabla "ejecuciones" condicionando la fecha de inicio de cada registro. 2) Obtiene ejecuciones que deben ser iniciadas. 3) Pone en cola las ejecuciones a realizar.
Cursos alternativos	
	Devuelve valor false y finaliza la ejecución del proceso

Tabla N° 21: Obtener datos fuente.

Caso de uso	Obtener datos fuente
Identificador	Permitir la construcción de consultas sql a partir de las configuraciones de alguna fuente de datos.
Curso típico de eventos previa.	1) Consultar la información de una tabla según configuración previa.
Usuario	Sistema
Actividad que hace el usuario con identificador de orden	<ol style="list-style-type: none"> 1) Consulta configuración de fuente de datos. 2) Establece conexión a servidor fuente. 3) Construye querys de consulta según configuración. 4) Obtiene información a procesar.
Cursos alternativos	
Devuelve valor false si no se pudo conectar a servidor	
Devuelve valor false si no existen resultados	
Finaliza el proceso	

Tabla N° 22: Guardar datos auditoria.

Caso de uso	Guardar datos auditoria
Identificador	Almacenar la información de los registros a ser procesados y que se obtuvieron de la fuente de datos.
Curso típico de eventos posterior de la información.	1) Guardar información en tablas de auditoría para análisis posterior de la información.
Usuario	Sistema
Actividad que hace el usuario con identificador de orden	<ol style="list-style-type: none"> 1) Registrar ingreso de cada fila obtenida de la fuente de datos. 2) Dividir en campo y valor la información de cada fila 3) Insertar en auditoria detalle cada campo y valor de cada fila.
Cursos alternativos	
Devolver false si no se pudo guardar la información	
Finalizar el proceso	

Tabla N° 23: Obtener datos auditoria.

Caso de uso	Obtener datos auditoria
Identificador	Consultar la información que debe ser procesada por medio de la condición de un estado de integración.
Curso típico de eventos	1) Obtener la información a integrar que no han sido procesados anteriormente.
Usuario	Sistema
Actividad que hace el usuario con identificador de orden	<ol style="list-style-type: none"> 1) Consultar registros con estado de integración igual a pendiente. 2) Obtener detalle de campo y valor por cada registro. 3) Crear objetos PHP para procesar la información de cada registro.
Cursos alternativos	
Devolver false si no existen resultados.	
Finalizar el proceso	

Tabla N° 24: Ejecutar procesos.

Caso de uso	Ejecutar procesos
Identificador	Ejecutar las inserciones, actualizaciones, consultas y condiciones a ejecutarse por cada registro de la fuente de datos.
Curso típico de eventos	1) Ejecutar procesos por cada registro existente
Usuario	Sistema
Actividad que hace el usuario con identificador de orden	<ol style="list-style-type: none"> 1) Obtener procesos a ejecutar. 2) Ejecutar los procesos pasando la información obtenida de cada registro de la fuente de datos. 3) Registrar estado del proceso. 4) Registrar estado de integración.
Cursos alternativos	
Devuelve false si no existen procesos	
Devuelve false si el proceso no se ejecutó.	
Finaliza la ejecución.	

CAPITULO IV. Análisis de alternativas

4.01 Matriz de análisis de alternativas.

Ver Anexo A.06

4.02 Matriz de análisis de impacto de los objetivos.

Tabla N° 26: Matriz de análisis de impacto de los objetivos.

Objetivos	Factibilidad de lograrse	Impacto ambiental	Relevancia	Sostenibilidad	Total
Mejor atención a requerimientos de clientes	4	1	4	4	13
Menor tiempo de entrega de proyectos	4	2	4	4	14
Desarrolladores sin sobrecarga de trabajo	4	3	4	4	15
Cumplimiento de fechas establecidas en cronograma	4	1	4	4	13
Aumento del ciclo de vida del software	4	4	4	4	16
Correcto funcionamiento del software	4	2	4	4	14
Total	24	13	24	24	85

4.03 Estándares para el diseño de clases

La arquitectura del sistema es Modelo-Vista-Controlador, esta arquitectura permite separar las capas de presentación, datos y lógica de negocio.

En el sistema se manejan clases únicamente para la capa de lógica de negocio y datos.

Para la creación de clases en la capa de lógica de negocios se maneja la siguiente estandarización.

- El nombre debe ser significativo.
- El nombre de la clase debe ser escrito en minúsculas a excepción de la primera letra.

Valido: Usuarios, Roles, Conexiones.

No valido: usuarios, roles, CONEXIONES.

Para la creación de clases en la capa de datos se maneja la siguiente estandarización.

- El nombre debe ser significativo.
- El nombre de la clase debe ser escrito en minúsculas a excepción de la primera letra.
- Después del nombre de la clase se debe aumentar `_model`.

4.04 Diagrama de Clases

Un **diagrama de clases** es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, orientados a objetos.

De este diagrama parte para que se pueda modelar el Físico, Lógico y generar el script para la base de datos.

Ver Anexo A.01.

4.05 Modelo Lógico – Físico

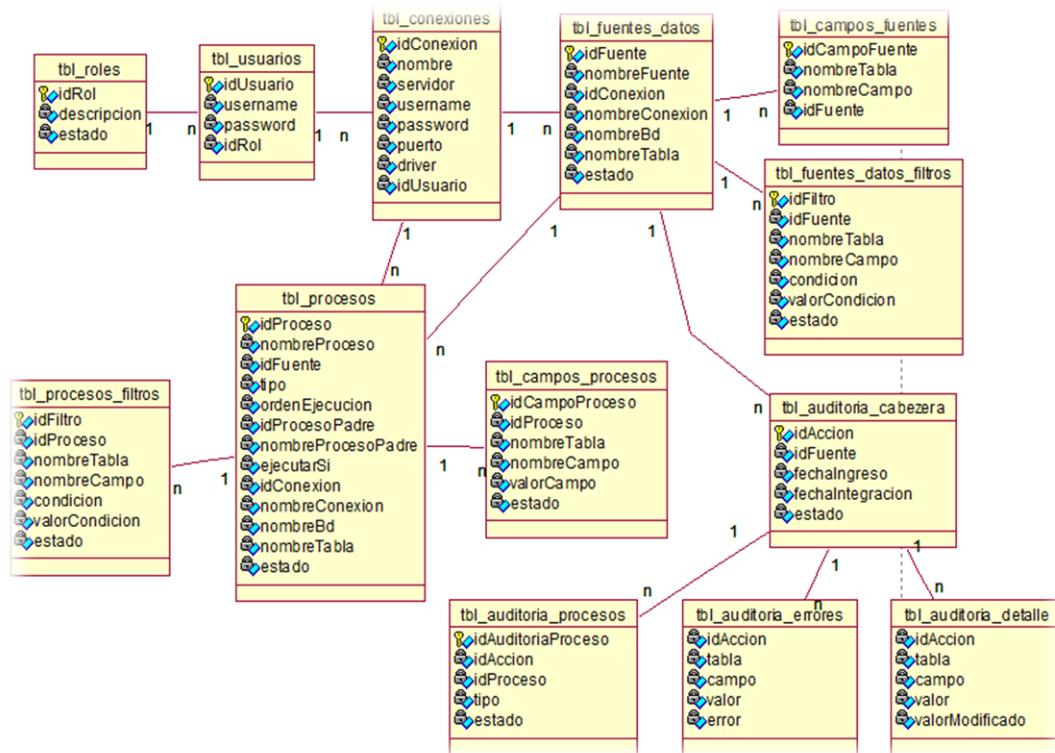


Figura 26. Modelo lógico: Este diagrama representa las clases utilizadas para el funcionamiento del sistema, estas clases actúan de forma independiente dependiendo del módulo a manipular.

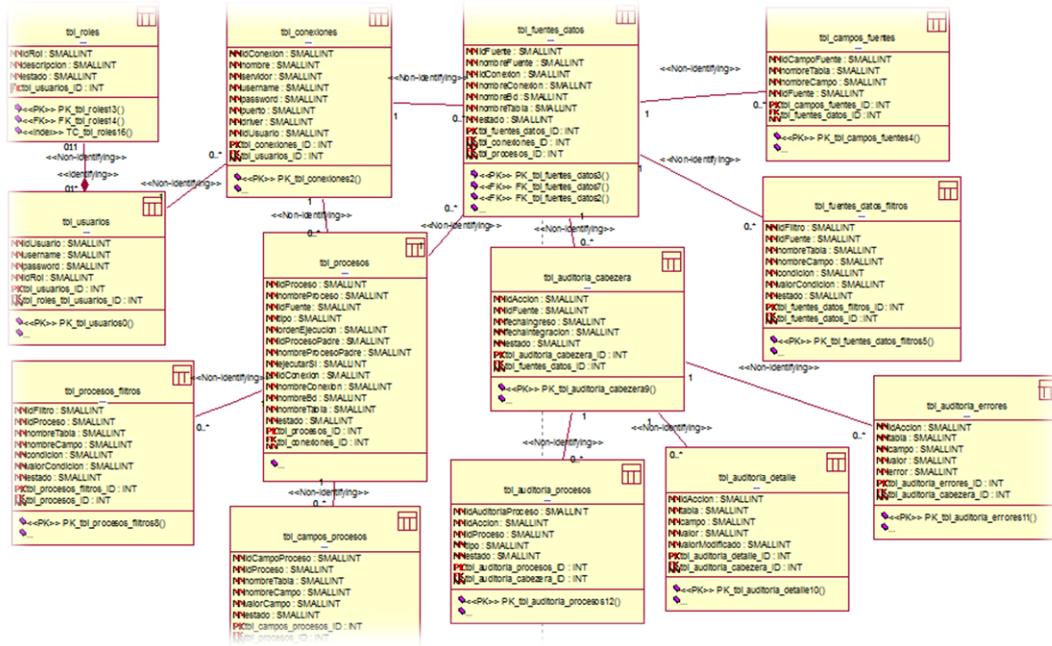


Figura 27. Modelo físico: Este diagrama representa el modelo de base de datos, manejando herencia y tipos de datos utilizados en los atributos de cada clase.

4.06 Diagrama de Componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

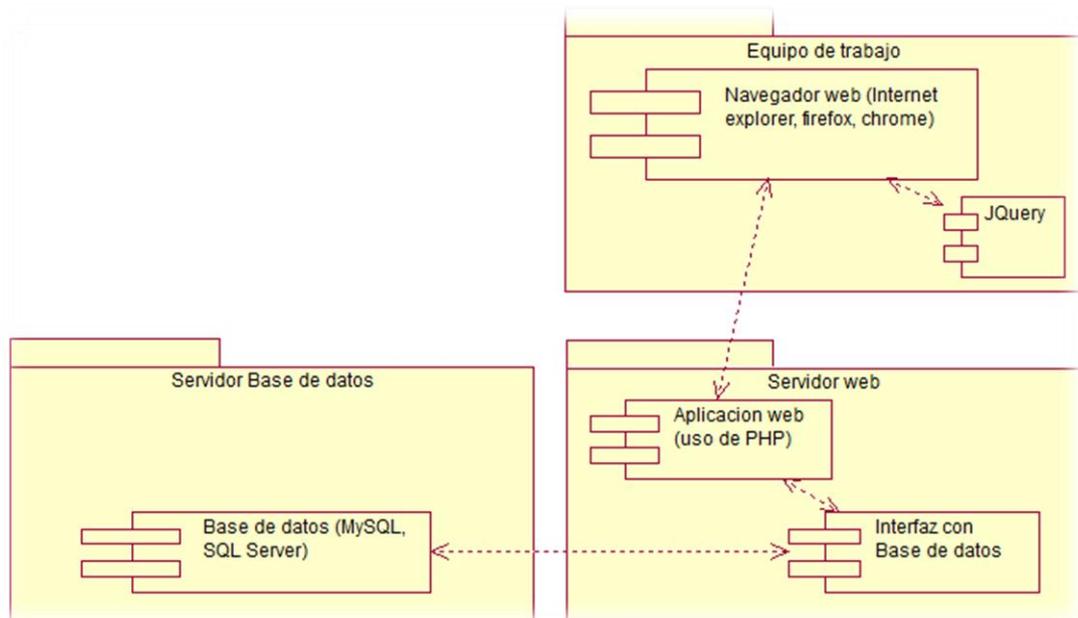


Figura 28. Diagrama de componentes: Por medio de este diagrama se representan los componentes necesarios para el funcionamiento del sistema, este proceso se divide en 3 capas, las cuales son servidor de base de datos, servidor de aplicaciones y navegador de cliente.

4.07 Diagrama de estrategias.

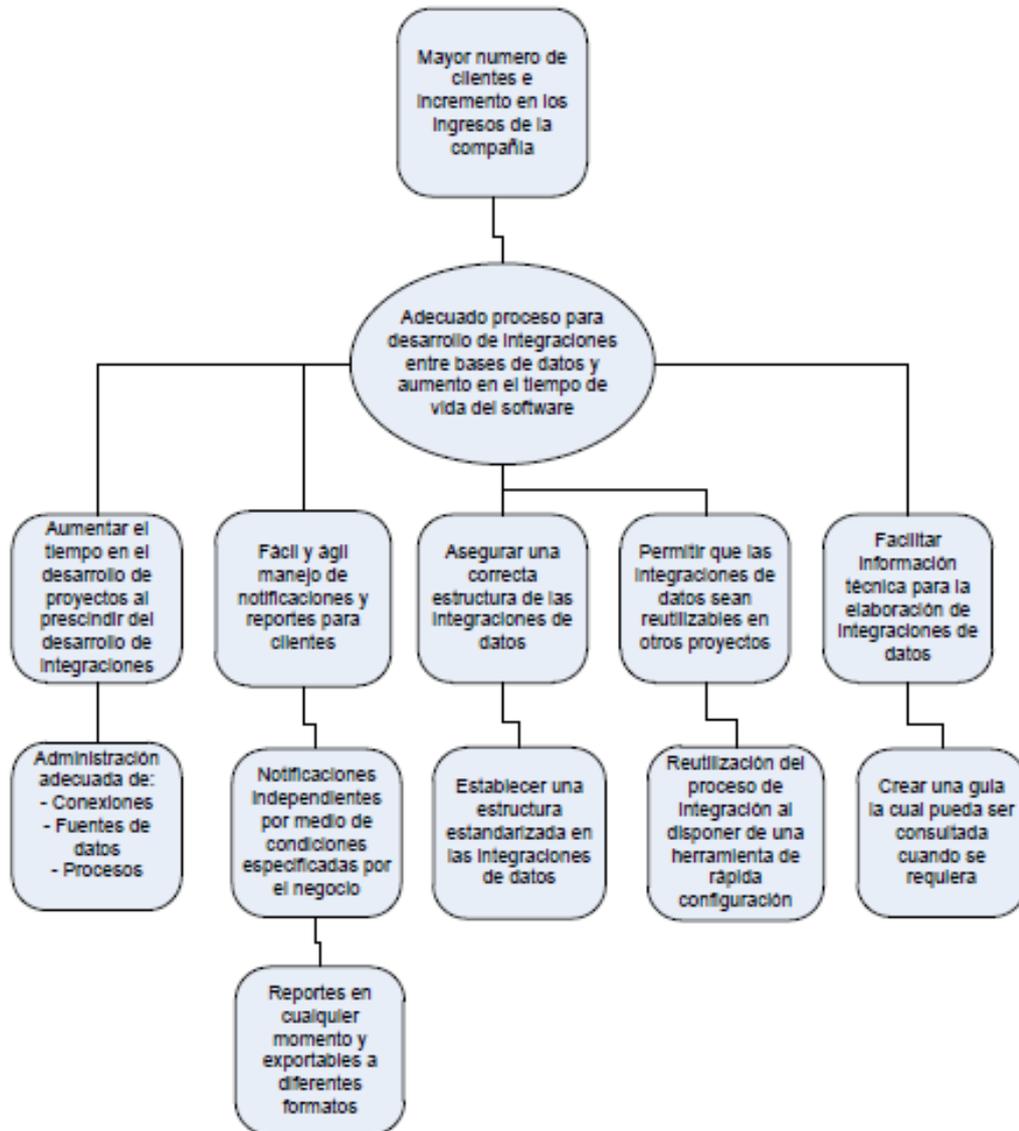


Figura 29. Diagrama de estrategias: Listamos las estrategias que ayudaran a cumplir el propósito planteado y los resultados esperados para el proyecto.

4.08 Matriz de marco lógico

Ver Anexo A.07

CAPITULO 5. Propuesta

5.01 Especificación de estándares de Programación

Dentro del desarrollo de la programación se va a utilizar distintos tipos de objetos y lenguajes los cuales se los maneja con la siguiente nomenclatura que se detalla a continuación.

5.01.01 Estándares para clases

La arquitectura del sistema es Modelo-Vista-Controlador, esta arquitectura permite separar las capas de presentación, datos y lógica de negocio.

En el sistema se manejan clases únicamente para la capa de lógica de negocio y datos.

Para la creación de clases en la capa de lógica de negocios se maneja la siguiente estandarización.

- El nombre debe ser significativo.
- El nombre de la clase debe ser escrito en minúsculas a excepción de la primera letra.

Valido: Usuarios, Roles, Conexiones.

No valido: usuarios, roles, CONEXIONES.

Para la creación de clases en la capa de datos se maneja la siguiente estandarización.

- El nombre debe ser significativo.
- El nombre de la clase debe ser escrito en minúsculas a excepción de la primera letra.
- Después del nombre de la clase se debe aumentar `_model`.

Valido: Usuarios_model, Roles_model, Conexiones_model.

No valido: Usuarios, RolesModel, conexiones_model.

5.01.02 Estándares para funciones

Para la creación de funciones en todas las clases se debe mantener la siguiente estandarización.

- El nombre debe ser significativo.
- El nombre de la función debe ser escrito en minúsculas a excepción de la primera letra de cada palabra desde la segunda.
- Las funciones deben devolver únicamente un valor.

```
public function registroUsuarios() {  
    $session = $this->session->all_userdata();  
    if (isset($session['username'])) {  
        $this->load->view('usuarios/registroUsuarios');  
    } else {  
        $this->index();  
    }  
}
```

Figura 30. Estándar funciones: Representa la forma correcta de nombrar una función, según los estándares antes expuestos.

5.01.03 Estándares para variables

Para la creación o declaración de variables se debe mantener la siguiente estandarización.

- El nombre debe ser significativo, haciendo mención al valor que contendrá.
- El nombre de la variable debe ser escrito en minúsculas a excepción de la primera letra de cada palabra desde la segunda.
- Las variables deben ser eliminadas cuando ya no sean necesarias.

```
$accion = $this->uri->segment(3);  
$idUserario = $this->uri->segment(4);
```

Figura 31. Estándar variables: Representa la forma correcta para declarar y utilizar variables, según los estándares ante expuestos.

5.01.04 Estándares para nombres de controles

Para la creación de controles se debe mantener la siguiente estandarización.

- El nombre debe ser significativo, haciendo mención al valor que se ingresara o mostrara en el control.
- El nombre del control debe ser escrito en minúsculas a excepción de la primera letra de cada palabra desde la segunda.

```
<input type="text" name="username" value="" id="username" placeholder="Nombre de usuario" class="form-control"/>
```

Figura 32. Estándar controles: Representa la forma correcta de nombrar los controles HTML a utilizar en la interfaz de usuario.

5.01.05 Estándares para nombres de archivos

Para la creación de archivos se debe mantener la siguiente estandarización.

- El nombre debe ser significativo, haciendo mención al contenido del archivo.
- El nombre del archivo debe ser escrito en minúsculas a excepción de la primera letra de cada palabra desde la segunda.
- La extensión del archivo debe ser especificada.

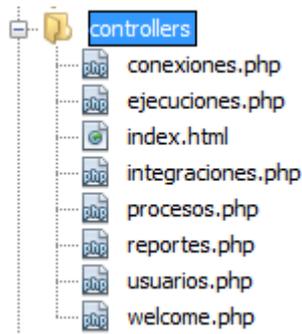


Figura 33. Estándar archivos: Representa la forma correcta de nombrar los archivos que contienen el código fuente del sistema.

5.01.06 Componentes HTML.

Para la creación de una aplicación web, es indispensable la utilización de codificación HTML, la cual es interpretada por el navegador web, permitiendo así el uso de la herramienta.

Las siguientes etiquetas son utilizadas en la capa de presentación de esta herramienta.

Ver Anexo A.08

5.01.07 Variables Globales

Se puede usar variables globales ya que pueden facilitar la programación y evitar el uso excesivo de variables locales.

Las variables globales a utilizar son las siguientes.

- \$_SESSION
- \$_POST
- \$_GET
- \$_SERVER

```
$datos = array(  
    'username' => $this->input->post('username'),  
    'password' => $this->input->post('password')  
);
```

Figura 34. Variables globales: Representa la forma correcta de utilizar variables de sesión, según los estándares antes expuestos.

5.01.08 Estándares de Base de Datos

Los objetos de base de datos deberán estar debidamente documentados con las siguientes propiedades.

5.01.09 Tipos de Datos

La descripción de los tipos de datos a utilizar dentro del diseño de la base de datos debe ser clara y precisa.

Tabla N° 28: Tipos de Datos

Tipo de Datos	Característica
Int	Número entero con o sin signo. Con signo el rango de valores va desde -2147483648 a 2147483647. Sin signo el rango va desde 0 a 429.4967.295
Date	Tipo fecha, almacena una fecha. El rango de valores va desde el 1 de enero del 1001 al 31 de diciembre de 9999. El formato de almacenamiento es de año-mes-día
DateTime	Combinación de fecha y hora. El rango de valores va desde el 1 de enero del 1001 a las 0 horas, 0 minutos y 0 segundos al 31 de diciembre del 9999 a las 23 horas, 59 minutos y 59 segundos. El formato de almacenamiento es de año-mes-día horas: minutos: segundos
Time	Almacena una hora. El rango de horas va desde -838 horas, 59 minutos y 59 segundos a 838, 59 minutos y 59 segundos. El formato de almacenamiento es de 'HH:MM:SS'
VarChar(n)	Almacena una cadena de longitud variable. La cadena podrá contener desde 0 a 255 caracteres.

5.01.10 Tablas

Para los nombres de las tablas se deben mantener las siguientes estandarizaciones.

- El nombre de la tabla debe ser significativo y debe hacer mención a los datos contenidos en la misma.
- Las palabras que conformen el nombre de la tabla deben estar separadas por un guion bajo (“_”).
- La codificación de las tablas debe ser UTF-8.

5.01.11 Campos

El nombre de un campo debe ser escrito en minúsculas excepto la primera letra de cada palabra, empezando desde la segunda.

Deben ser establecidos los tipos de datos que albergaran estos campos, teniendo un límite, un valor por defecto y se debe especificar si se aceptan nulos o no.

Se deberán identificar las claves primarias, las mismas que no deberán permitir valores nulos y de preferencia ser de tipo INT.

Tabla N° 29: Nombre de Campos

Nombre	Nomenclatura	Tipo	Allow Null	PK
Identificador de Usuario	idUsuario	Int	No	Si
Fecha de creación	fechaCreacion	datetime	No	No

5.02 Diseño de Interfaces de usuario

Ingreso al sistema

Ingrese nombre de usuario

Ingrese password de usuario

Ingresar

Figura 35. Inicio de sesión: Se debe ingresar las credenciales de usuario como son: nombre de usuario y password, las cuales permitirán el ingreso al sistema.

Home Usuarios Conexiones Fuentes de datos Ejecuciones Reportes

Lista de usuarios existentes

Id Usuario	Username	Accion	
4	rolo	Actualizar	Eliminar

Figura 36. Listar Usuarios: Interfaz para visualización y selección de usuarios.

Home Usuarios Conexiones Fuentes de datos Ejecuciones Reportes

Registro nuevos usuarios

Ingrese nombre de usuario

Ingrese password de usuario

Confirme password de usuario

Guardar

Figura 37. Crear usuario: Se deben ingresar los datos necesarios para la creación de un nuevo usuario como son Nombre de usuario, password y confirmación de password.

Home Usuarios Conexiones Fuentes de datos Ejecuciones Reportes

Lista de conexiones existentes

Id Conexion	Nombre	Tipo	Servidor	Driver	Accion	
24	Sql server 2	fuentes	ROLEINS	sqlsrv	Actualizar	Eliminar
25	Mysql	fuentes	localhost	mysql	Actualizar	Eliminar
26	Mysql destino	destino	localhost	mysql	Actualizar	Eliminar
27	plus crm	fuentes	pluscrm	mssql	Actualizar	Eliminar

Figura 38. Listar conexiones: Permite visualizar y seleccionar las conexiones creadas.

Home Usuarios Conexiones Fuentes de datos Ejecuciones Reportes

Registro de nuevas conexiones

Seleccione tipo de servidor	Servidor fuente
Ingrese nombre de conexion	Nombre de conexion
Ingrese nombre de servidor	Nombre de servidor
Ingrese username de servidor	Username de servidor
Ingrese password de servidor	Password de servidor
Ingrese puerto de servidor	Puerto de servidor
Seleccione driver de conexion	MySQL

Guardar

Figura 39. Crear conexión: Permite la creación de nuevas conexiones al ingresar las credenciales de los servidores de base de datos necesarios.

Home Usuarios Conexiones Fuentes de datos Ejecuciones Reportes

Lista de fuentes de datos existentes

	Id Fuente	Nombre	Conexion	Estado	Accion
Ver Procesos	1	fuelle 1	Mysql	Activo	Actualizar Eliminar
Ver Procesos	2	prueba	Mysql	Activo	Actualizar Eliminar
Ver Procesos	3	db_sugar_crm	Mysql	Activo	Actualizar Eliminar

Figura 40. Listar fuente de datos: Permite la visualización y selección de fuentes de datos creadas

Home Usuarios Conexiones Fuentes de datos Ejecuciones Reportes

Registro de nuevas fuentes de datos

Nombre fuente de datos	Tabla	Campo	Condicion	Valor	Tablas
<input type="text" value="Nombre fuente de datos"/>					<input type="text"/>
Seleccione conexion					Campos Filtros
<input type="text"/>					<input type="text"/>
Bases de datos					Campos
<input type="text"/>					<input type="text"/>

Guardar

Figura 41. Crear fuente de datos: Permite la creación de nuevas fuentes de datos al ingresar los campos necesarios para formar internamente una sentencia SQL de tipo SELECT

Home Usuarios Conexiones Fuentes de datos Ejecuciones Reportes

Registro de proceso insertar

Nombre proceso	Tabla	Destino	Fuente	Tablas
<input type="text" value="Nombre fuente de datos"/>				<input type="text"/>
Conexion				Campos
<input type="text"/>				<input type="text"/>
Bases de datos				Ejecutar si
<input type="text"/>				<input type="text" value="Fuente de datos es True"/>

Guardar

Figura 43. Crear proceso: Permite la creación de procesos insertar al ingresar los campos necesarios para formar internamente una sentencia SQL de tipo INSERT

Home Usuarios Conexiones Fuentes de datos Ejecuciones Reportes

Lista de ejecuciones existentes

Id Ejecucion	Nombre	Nombre Fuente	Fecha inicio	Hora inicio	Fecha fin	Hora fin	Accion
1	ejecucion prueba3	fuentes 1	2014-03-29	12:00:00	2014-03-29	00:00:00	Actualizar Eliminar
2	prueba noti	fuentes 1	2014-03-30	01:01:00	2014-03-30	01:02:00	Actualizar Eliminar
3	ejecutar a sugar	db_sugar_crm	2014-03-31	02:01:00	2014-03-31	02:02:00	Actualizar Eliminar

Para iniciar las ejecuciones automaticamente debe crear una tarea programada de windows, que ejecute un archivo por lotes con los siguiente comandos de ejemplo:
cd C:\wamp\www\HerramientaIntegraciones2
C:\wamp\bin\php\php5.3.13\php.exe -c C:\wamp\bin\php\php5.3.13\php.ini -f index.php ejecuciones exec

Figura 44. Listar ejecuciones: Permite la visualización y selección de ejecuciones creadas.

Home Usuarios Conexiones Fuentes de datos Ejecuciones Reportes

Registro de nuevas ejecuciones

Ingrese nombre de ejecucion	<input type="text" value="Nombre de ejecucion"/>
Seleccione fuente de datos	<input type="text" value="fuente 1"/>
Ingrese fecha de inicio	<input type="text" value="mm/dd/aaaa"/>
Ingrese hora de inicio	<input type="text" value="--:-- --"/>
Ingrese fecha de finalizacion	<input type="text" value="mm/dd/aaaa"/>
Ingrese hora de finalizacion	<input type="text" value="--:-- --"/>
Configuracion adicional	<input type="button" value="Programar notificaciones"/>
<input type="button" value="Guardar"/>	

Figura45. Crear ejecución: Permite la creación de nuevas ejecuciones al ingresar los datos necesarios para la definición de los rangos de ejecución.

5.03 Especificación de pruebas de unidad

Tabla N° 30: Prueba de unidad 1

Modulo	Ingresar al sistema
Identificador	PruUni1
Curso típico de eventos	
Ingresa credenciales, Valida credenciales, Permite o niega el ingreso al usuario	
Usuario	Sistema
Ingresar nombre de usuario, Ingresa contraseña, Presiona botón ingresar	Verifica que el usuario y contraseña sean correctos., Permitir el ingreso al sistema, Cargar datos dependiendo del usuario
Cursos alternativos	
Notificar que las credenciales son incorrectas, Negar ingreso al sistema	

Tabla N° 31: Prueba de unidad 2

Modulo	Crear Registros
Identificador	PruUni2
Curso típico de eventos	
Ingresar información de registro, Guardar información	
Usuario	Sistema
Ingresar información de registro nuevo	Validar información requerida, guardar información
Cursos alternativos	
Notificar que la información es incorrecta, No guardar información	

Tabla N° 32: Prueba de unidad 3

Modulo	Actualizar Registro
Identificador	PruUni3
Curso típico de eventos	
Ingresar información de registro, Actualizar información	
Usuario	Sistema
Seleccionar registro, Ingresar información de registro a actualizar	Validar información requerida, actualizar información de registro seleccionado
Cursos alternativos	
Notificar que la información es incorrecta, No guardar información	

Tabla N° 33: Prueba de unidad 4

Modulo	Eliminar Registro
Identificador	PruUni4
Curso típico de eventos	
Eliminar información	
Usuario	Sistema
Seleccionar registro, presionar acción de eliminar	Confirmar eliminación, eliminar registro, eliminar información relacionada al registro
Cursos alternativos	
Notificar que no se pudo eliminar, No eliminar información	

5.04 Especificación de pruebas de aceptación

Tabla N° 34: Prueba de aceptación 1

Identificador de la prueba	PruAccept1
Método a probar	Conexión a MySQL y Sql Server, Configuración de conexiones a servidores
Objetivo de la prueba	Verificar el cumplimiento de requerimientos funcionales
Datos de entrada	
	<ol style="list-style-type: none"> 1) Credenciales de acceso a servidor MySQL y Sql Server 2) Selección de driver de conexión Mysql y Sql Server
Resultado esperado	
	<ol style="list-style-type: none"> 1) La conexión es exitosa 2) Se guarda la información de la conexión creada
Comentarios	
	<p>Al crear o actualizar una conexión se verifica que la conexión sea posible antes de guardarla, esta verificación intenta conectar con el servidor y obtiene una cadena de conexión o valor booleano FALSE</p>

Tabla N° 35: Prueba de aceptación 2

Identificador de la prueba	PruAcept2
Método a probar	Creación y consulta de fuentes de datos por medio de la construcción de sentencias SQL tipo select
Objetivo de la prueba	Verificar el cumplimiento de requerimientos funcionales
Datos de entrada	
	<ul style="list-style-type: none"> 3) Selección de conexión 4) Selección de base de datos 5) Selección de tabla 6) Selección de campos 7) Configurar filtros 8) Configurar procesos por registros obtenidos 9) Configurar ejecución de fuente de datos y procesos
Resultado esperado	
	<ul style="list-style-type: none"> 3) La ejecución se inicia a la hora planificada 4) El estado resultante de la ejecución es TRUE
Comentarios	
	Para verificar la consulta SQL correctamente se realizo el proceso completo de configuración y se verifico el numero de registros obtenidos, junto con el estado resultante de la ejecución

Tabla N° 36: Prueba de aceptación 3

Identificador de la prueba	PruAccept3
Método a probar	Creación y ejecución de procesos por cada registro proveniente de la fuente de datos
Objetivo de la prueba	Verificar el cumplimiento de requerimientos funcionales
Datos de entrada	
	<ul style="list-style-type: none"> 10) Selección de fuente de datos 11) Creación de proceso tipo Consultar 12) Creación de proceso tipo Insertar 13) Creación de proceso tipo Actualizar 14) Configurar ejecución de fuente de datos y procesos
Resultado esperado	
	<ul style="list-style-type: none"> 5) La ejecución se inicia a la hora planificada 6) El estado resultante de la integración de cada registro es TRUE
Comentarios	
	<p>Para verificar la creación y ejecución de procesos por cada registro existente, se creó a partir de una fuente de datos los procesos que se ejecutarán. Estos procesos devuelven un estado de TRUE o FALSE dependiendo del resultado de la ejecución de los mismos.</p>

Tabla N° 37: Prueba de aceptación 4

Identificador de la prueba	PruAccept4
Método a probar	La información a integrar se registra en una base de datos de auditoría
Objetivo de la prueba	Verificar el cumplimiento de requerimientos funcionales
Datos de entrada	
	15) Configurar ejecución de fuente de datos y procesos
Resultado esperado	
	7) La ejecución se inicia a la hora planificada
	8) Se registra el inicio de la ejecución
	9) Se registra el ingresos de cada registro de la fuente de datos en la tabla auditoria_cabecera
	10) Se registra cada campo y valor ingresado por cada registro en la tabla auditoria_detalle
Comentarios	
	Al iniciarse la ejecución planificada, se crea un registro por cada fila a integrar.

Tabla N° 38: Prueba de aceptación 5

Identificador de la prueba	PruAccept5
Método a probar	Notificar por email los registros obtenidos de los procesos seleccionados
Objetivo de la prueba	Verificar el cumplimiento de requerimientos funcionales
Datos de entrada	
	<ul style="list-style-type: none"> 16) Configurar ejecución de fuente de datos y procesos 17) Establecer destinatarios de notificaciones 18) Establecer proceso a notificar 19) Establecer estado de proceso a notificar
Resultado esperado	
	<ul style="list-style-type: none"> 11) La ejecución se inicia a la hora planificada 12) Se registra el envío de la notificación. 13) Se recibe notificación y archivo CSV con información
Comentarios	
	Al finalizar la ejecución se verifica si se debe realizar alguna notificación, se realiza una consulta por medio del proceso a notificar y el resultado del mismo. Se obtienen los datos guardados en auditoria y se genera un archivo tipo CSV a enviarse al destinatario.

5.05 Especificación de pruebas de carga

Tabla N° 39: Prueba de carga 1

Identificador de la prueba	PruCarga1
Método a probar	Cantidad de información a procesarse en una ejecución
Objetivo de la prueba	Obtener la cantidad de registros máximos que pueden ser procesados
Datos de entrada	
	20) Configurar ejecución de fuente de datos y procesos
	21) Fuente de datos contiene 100 registros
Resultado esperado	
	14) La ejecución se inicia a la hora planificada
	15) Se realiza la integración de 100 registros
	16) Se verifica el estado de integración de cada registro
Comentarios	
	Al finalizar la ejecución se verifica si el estado resultante coincide con la integración de datos realizada.

Tabla N° 40: Prueba de carga 2

Identificador de la prueba	PruCarga2
Método a probar	Cantidad de información a procesarse en una ejecución
Objetivo de la prueba	Obtener la cantidad de registros máximos que pueden ser procesados
Datos de entrada	
	22) Configurar ejecución de fuente de datos y procesos
	23) Fuente de datos contiene 1000 registros
Resultado esperado	
	17) La ejecución se inicia a la hora planificada
	18) Se realiza la integración de 1000 registros
	19) Se verifica el estado de integración de cada registro
Comentarios	
	Al finalizar la ejecución se verifica si el estado resultante coincide con la integración de datos realizada.

Tabla N° 41: Prueba de carga 2

Identificador de la prueba	PruCarga1
Método a probar	Cantidad de información a procesarse en una ejecución
Objetivo de la prueba	Obtener la cantidad de registros máximos que pueden ser procesados
Datos de entrada	
	24) Configurar ejecución de fuente de datos y procesos
	25) Fuente de datos contiene 10000 registros
Resultado esperado	
	20) La ejecución se inicia a la hora planificada
	21) Se realiza la integración de 10000 registros
	22) Se verifica el estado de integración de cada registro
Comentarios	
	Al finalizar la ejecución se verifica si el estado resultante coincide con la integración de datos realizada.

5.06 Configuración del ambiente mínima/ideal

Para la puesta en marcha del sistema se requiere de componentes mínimos en el servidor a utilizar.

- Windows XP 32bits, Windows Server 2008 32bits
- MySql 5.1.0 o Sql Server 2008
- Servidor web Apache 2.2
- PHP 5.2
- Memoria RAM 2GB
- Procesador Intel Atom de 1.67GHz

CAPITULO VI. Aspectos administrativos

6.01 Recursos

Tabla N° 42: Recursos

TIPO	RECURSO
Recurso Intangible	Computador
Recurso Intangible	Impresora
Recurso Renovable	Hojas
Recurso Tangible	Transporte
Recurso Tangible	Alimentación
Recurso Didáctico	Internet
Recurso Humano	Conocimiento
Recurso Humano	Experiencia
Recurso Económico	Proyecto

6.02 Presupuesto

Tabla N° 43: Presupuesto

RECURSO	VALOR UNITARIO	VALOR TOTAL
(Seminario, Tutorías	722,00	722,00
Impresiones B/N	0,10	30,00
Impresiones Color	0.25	40,00
Internet		60,00
Transporte	2,50	75,00
Alimentación	2,00	60,00
Otros Gastos		150,00
Anillado y Empastado		80,00
Total		1217,00

6.03 Cronograma

Ver Anexo A.09

CAPITULO VII. Conclusiones y Recomendaciones

7.01 Conclusiones

I.- Con la ayuda del sistema herramienta de integraciones se logra mantener una organización y estandarización en el flujo de datos correspondiente a integraciones entre bases de datos de distintas plataformas.

II.- Por medio del sistema herramienta de integraciones se logra reducir el tiempo de desarrollo de proyectos que requieran comunicación de datos entre distintas bases de datos, al permitir un funcionamiento basado en configuraciones según los requerimientos de negocio lo soliciten.

III.- Se demostró que la automatización de procesos de programación que involucran manejo de datos, es posible y de gran utilidad.

IV.- Se demostró que las notificaciones respectivas a los actores involucrados en el negocio, son moldeables, seguras y efectivas al enviar información sobre el cumplimiento de condiciones y procesos por medio de archivos CSV y manejo de servidores SMTP.

V.- Los sistemas basados en web con programación php, permitirán la utilización de la herramienta en distintos sistemas operativos, ya que un servidor intérprete de php puede ser instalado en diversos ambientes.

VI.- El sistema al simular procesos utilizados en programación de integraciones de datos, permitirá que las modificaciones sean completamente visuales y fáciles de realizar y no será necesario interpretar código fuente escrito para entender su funcionamiento.

7.02 Recomendaciones

I.- Utilizar las funciones preestablecidas del sistema, para mejorar la experiencia de usuario y flujo correcto de información entre bases de datos.

II.- Al existir información dependiente de ciertas configuraciones, las mismas no podrán mantenerse al realizar modificaciones, se recomienda organizar un flujo de datos validado antes de crear fuentes de datos o procesos.

III.- La herramienta está basada en active record para el manejo de bases de datos, el uso de la herramienta permitirá que la información sea procesada con los mayores estándares de programación php existentes en la actualidad.

IV.- Para impedir que el proceso de ejecución se realice contantemente y perjudique al performance del servidor, se recomienda manejar un estado de integración en la fuente de datos, el cual al finalizar los procesos de integración se actualice a un valor diferente por medio del cual el sistema identifique si ya fue procesado el registro.

V.- Para adaptarse al performance más básico y no consumir tanta memoria se recomienda que se realice una configuración previa del número de registros a procesarse de forma paginada, el valor por defecto es de 100 registros a la vez, hasta completar el total de registros de la fuente de datos.

VI.- Para evitar inconvenientes al momento de ejecutar los procesos de inserción y actualización se recomienda dar permisos suficientes al usuario de base de datos configurado en la conexión a los servidores.

ANEXOS

A.01

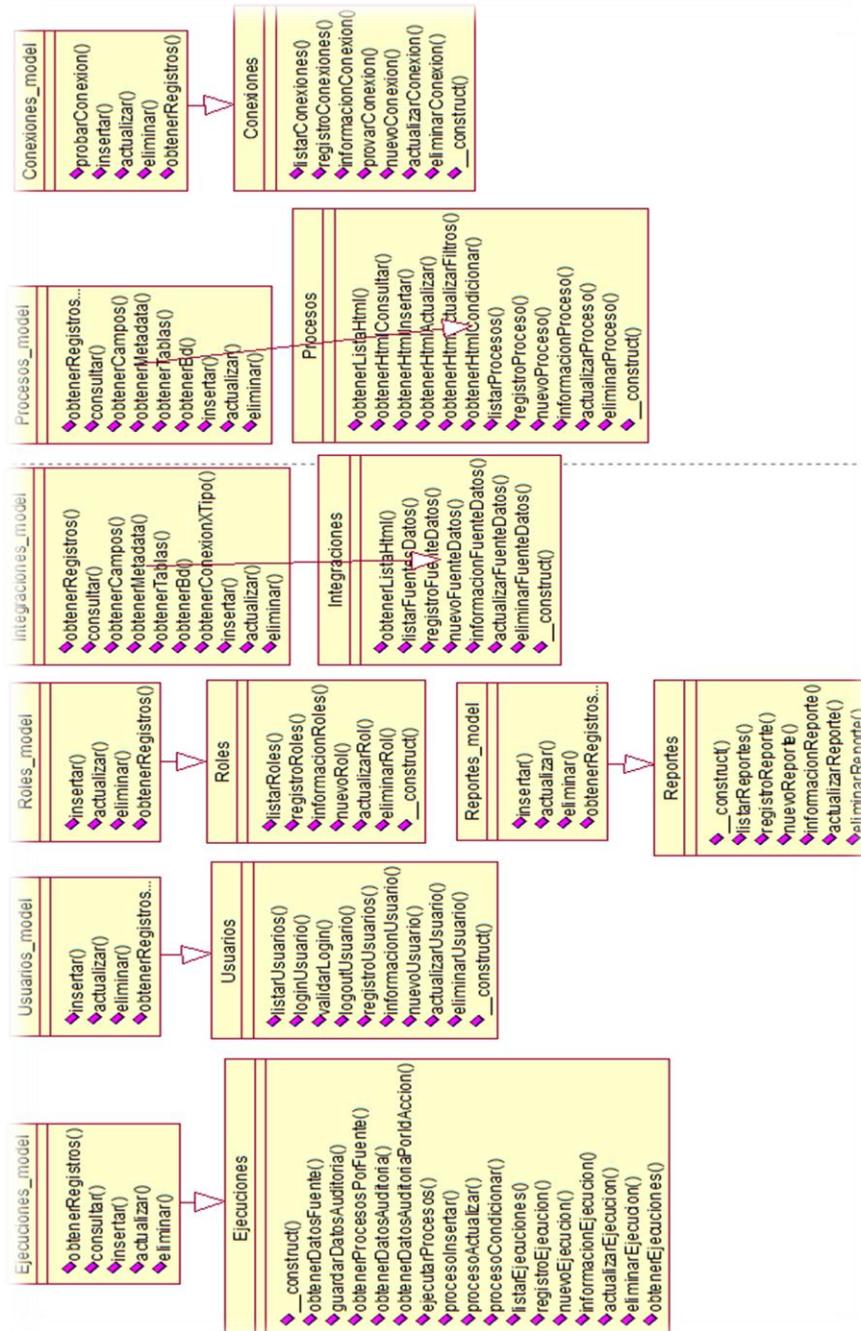


Figura46. Diagrama de clases: Permite visualizar las clases utilizadas para el funcionamiento y correcta estructura del sistema.

A.02

Tabla N° 1. Matriz de Fuerzas T.

Análisis de fuerzas					
Situación empeorada	Situación actual			Situación mejorada	
Disminución considerable en la cartera de clientes por tiempo de entrega excesivo	Inadecuado proceso para desarrollo de integraciones entre bases de datos y disminución en el tiempo de vida del software			Estandarización del desarrollo de integraciones, disminución en el tiempo de entrega de proyectos	
Fuerzas impulsadoras	I	PC	I	PC	Fuerzas bloqueadoras
Manejo de estándares para operatividad de transacciones y Active record que permitirán el desarrollo confiable de integraciones y facilidad en el soporte técnico	2	5	5	2	Inadecuado manejo de estándares en el desarrollo de integraciones, cada programador utiliza diferente funcionalidad.
Prescindir del desarrollo de integraciones y disminuir el tiempo de entrega	1	5	5	1	Exceso de tiempo contemplado en el cronograma de trabajo
Facilidad en la implementación de nuevos requerimientos, y rápida reestructuración de procesos, por medio de actualización de configuraciones.	3	5	5	3	Disminución del tiempo de vida del software, al no asegurar escalabilidad

Rapidez y facilidad en el desarrollo, menor cantidad de empleados involucrados	2	5	5	2	Utilización de recursos humanos y sobrecarga de trabajo
Eficaz creación de notificaciones que dependan de un suceso en particular y reportes tipo csv disponibles para los clientes y adjuntos por correo electrónico	1	5	5	1	Limitación en reportes y notificaciones generadas para los clientes

A.03

Tabla N° 2: Entrevista Wladimir Pazmiño (Director tecnológico).

Diseño Entrevista		
identificador: Ent01		
Preguntas	Objetivos	Análisis posterior
¿Cuáles son los sistemas que más utilizan integraciones de datos?	Identificar los productos de la empresa a los que enfocar la solución	Los sistemas que requieren mayor utilización de integraciones de datos son: Los sistemas CRM
¿Qué sistema CRM personaliza la empresa?	Conocer el nombre de la herramienta para su investigación posterior	El sistema personalizado y comercializado por la empresa es SugarCRM
¿Cuáles son las áreas de las empresas clientes involucradas en los proyectos SugarCRM?	Conocer las áreas que interfieren en el desarrollo de los proyectos	Las áreas involucradas de las empresas clientes son: marketing, CRM y Sistemas
¿Cuál es la principal utilidad del sistema SugarCRM para el área de Marketing?	Conocer las principales acciones que requiere el área de marketing	El área de marketing se favorece de la información que

		recauda sobre los clientes y de sus intereses
¿Cuál es la principal utilidad del sistema SugarCRM para el área de CRM?	Conocer las principales acciones que requiere el área de CRM	El área de CRM puede proporcionar mejor servicio a los clientes por medio de la información recaudada y campañas
¿Cuál es la principal utilidad del sistema SugarCRM para el área de sistemas?	Conocer las principales acciones que requiere el área de sistemas	El área de sistemas se favorece de la calidad de la información de los clientes y la constante actualización de datos
¿Cuáles son las bases de datos involucradas en las integraciones de datos para SugarCRM?	Conocer los motores de bases de datos más utilizados por los clientes	Las bases de datos más utilizadas son: Sql Server, MySQL
¿Cuál sería la utilidad al disminuir el tiempo de desarrollo de integraciones de datos?	Conocer los principales beneficios de reducir el tiempo de programación.	Se podría mejorar los tiempos de entrega en los proyectos, se tendría ventaja en las licitaciones al reducir el tiempo y los recursos, se podría utilizar ese tiempo para otros proyectos
¿Quiénes serían los usuarios del sistema?	Conocer los tipos de usuarios que debe tener el sistema	El sistema sería dedicado solo para el uso de los programadores
¿Cuáles son los lenguajes que utilizan para el desarrollo de las integraciones de datos?	Conocer el lenguaje de programación en el cual realizar el sistema	El lenguaje de programación utilizado es PHP ya que SugarCRM es basado en WEB y maneja este lenguaje

A.04

Tabla N° 3: Matriz de requerimientos.

Matriz de requerimientos					
Identificador	Descripción	Fuente	Prioridad	Estado	Usuarios involucrados
Requerimientos Funcionales					
REQF1	El sistema debe permitir la conexión a MySql y Sql server	Wladimir Pazmiño	Media	Validado	Programador
REQF2	Debe permitir la configuración de conexiones a los servidores	Wladimir Pazmiño	Alta	Validado	Programador
REQF3	Debe permitir sentencias sql "Select"	Wladimir Pazmiño	Alta	Validado	Programador
REQF4	Debe permitir la creación de fuentes de datos	Wladimir Pazmiño	Alta	Validado	Programador
REQF5	Debe permitir procesar cada registro proveniente de una fuente	Wladimir Pazmiño	Alta	Validado	Programador
REQF6	Los procesos deben permitir la creación de sentencias Sql "Insert", "Update", "Condiciones"	Wladimir Pazmiño	Alta	Validado	Programador
REQF7	La información a integrar debe registrarse en una base de datos de auditoria	Wladimir Pazmiño	Alta	Validado	Programador
REQF8	Se debe poder notificar por email los registros resultantes de los procesos	Wladimir Pazmiño	Alta	Validado	Programador
Requerimientos no funcionales					

REQNF1	El sistema debe ser confiable	Wladimir Pazmiño	Alta	Validado	Programado
REQNF2	El sistema debe controlar el acceso para mayor seguridad	Wladimir Pazmiño	Alta	Validado	Programado
REQNF3	El sistema debe permitir escalabilidad	Wladimir Pazmiño	Alta	Validado	Programado
REQNF4	El sistema debe tener facilidad de uso	Wladimir Pazmiño	Alta	Validado	Programado

A.05

Tabla N° 4: Matriz de análisis de involucrados.

Actores Involucrados	Interés sobre el problema central	Problemas percibidos	Recursos y mandatos	Interés sobre el proyecto	Conflictos potenciales
Gerente General	Garantizar a los clientes y al personal las herramientas necesarias para cumplir con sus necesidades	Existen quejas por incumplimiento en los cronogramas	R1. Humano R2. Administrativo R3. Económico M1. Manejar los recursos eficientemente	Evitar inconvenientes por incumplimiento de plazos y mejorar la organización entre áreas	Pérdida de clientes para la compañía
Director tecnológico	Levantamiento de información, análisis y diseño del software y	Falta de personal por exceso de requerimientos	R1. Humano R2. Administrativo R3. Técnico R4. Comercial M1.	Mejorar tiempos de entrega para dar un valor agregado al producto final	Perdida de información y mal manejo de reportes

	manejo de áreas tecnológicas		Estandarizar procesos		
Jefe CRM, BPM & WEB	Controlar las tareas de los desarrolladores y dar información necesaria sobre los proyectos	Insatisfacción de los desarrolladores y clientes	R1. Humano R2. Técnico M1. Manejar un proceso común para las integraciones	Mejorar la salud laboral de los empleados, disminuir la sobrecarga de trabajo	Se deben trabajar horas extras para cumplir con cronogramas
Desarrollador	Desarrollo e implementación de los requerimientos	Realizar desarrollo de integraciones por cada proyecto, dificultad al implementar nuevos requerimientos	R1. Técnico. M1. Mejorar el proceso de integración de datos para facilitar el soporte técnico	Disminuir la carga de trabajo para cumplir con las fechas de entrega establecidas	El soporte técnico no lo realiza siempre el mismo desarrollado, hay un estándar con el cual se faciliten los cambios a realizar
Clientes de la empresa	Utilización y petición de nuevos requerimientos	Existen problemas en el funcionamiento de los reportes y de la información	R1. Usuario final M1. Tener un mejor servicio y más rápido	Disponer de reportes cuando se los necesite y que se cumplan los plazos establecidos	No se puede utilizar la herramienta si hay problemas con la información

A.06

Tabla N° 25: Matriz de análisis de alternativas.

Objetivos	Impacto sobre el propósito	Factibilidad técnica	Factibilidad financiera	Factibilidad social	Total	Categoría
Mejor atención a requerimientos de clientes	3	3	3	3	12	Alta
Menor tiempo de entrega de proyectos	3	3	3	3	12	Alta
Desarrolladores sin sobrecarga de trabajo	3	3	2	3	11	Alta
Cumplimiento de fechas establecidas en cronograma	3	3	2	3	11	Alta
Aumento del ciclo de vida del software	3	3	3	3	12	Alta
Correcto funcionamiento del software	3	3	3	3	12	Alta
Total	18	18	16	18	70	

A.07

Tabla N° 27: Análisis del marco lógico.

SUPUESTOS	RESUMEN NARRATIVO	INDICADORES	MEDIOS DE	SUPUESTOS
El personal de la empresa cambian de actitud en los nuevos procesos de organización	FINALIDAD DEL PROYECTO Mejorar el tiempo dedicado a la programación de integraciones	El decremento del tiempo empleado en la creación de procesos de integración de datos se reducirá en un 50% La carga de trabajo de los recursos, disminuirá en un 30% al evitar la programación y configurar visualmente	Disminución de quejas o problemas en la lógica de negocio. Recursos humanos sin mucha carga de trabajo Reportes de la información integrada, a los encargados de su análisis y corrección	La persona encargada de realizar las integraciones termine sus labores intempestivamente.
	PROPÓSITO DEL PROYECTO Permitir la configuración de integraciones de datos, de una forma sencilla y sin involucrar	La rapidez de entrega de procesos de integración de datos.	Encuestas realizadas a la persona encargada y mejora en el performance del servidor	La lógica del negocio cambie de forma brusca
	COMPONENTES DEL PROYECTO <ul style="list-style-type: none"> Optimización en el uso del recurso en las operaciones de la empresa 	La fiabilidad de funcionamiento de los recursos utilizados		Los recursos utilizados deterioren en el proceso de alguna actividad

RESUMEN NARRATIVO	INDICADORES	MEDIOS DE VERIFICACIÓN
ACTIVIDADES DEL PROYECTO	El nivel de aceptación en el proceso tiene buena utilización del recurso a la vez buena selección de información se reflejará en la optimización de la actividad.	Verificación visual del comportamiento de cada uno de los usuarios.
Capacitar a los desarrolladores de software, para el uso de la herramienta.		
Realizar control del uso de recursos.		

A.08

Tabla N° 27: Componentes HTML.

DETALLE DEL COMANDO	ETIQUETA DE APERTURA	ETIQUETA DE CIERRE
Principio de Documento	<html>	</html>
Encabezado	<head	</head>
Título	<title>	</title>
Cuerpo del documento	<body>	</body>
links	<a>	

Imagen Individual		
Menú		
Ítem menú		
Etiqueta	<label>	</label>
Contenedor	<div>	</div>
Salto de línea	 	
Lista desplegable y opciones	<select><option>	</option></select>
Tabla	<table><tr><td>	</td></tr></table>
Campo password	<input type="password"/>	
Campo texto	<input type="text"/>	
Botón submit	<input type="submit"/>	
Botón	<input type="button"/>	

A.09

Tabla N° 33: Presupuesto

Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin
Programada manualmente	Proyecto de Grado	122 días	mié 30/10/13	jue 17/04/14
Programada manualmente	Capítulo I	12 días	mié 30/10/13	jue 14/11/13
Programada manualmente	Contexto	4 días	mié 30/10/13	lun 04/11/13
Programada manualmente	Justificación	4 días	mar 05/11/13	vie 08/11/13

Programada manualmente	Definición Matriz T	4 días	lun 11/11/13	jue 14/11/13
Programada manualmente	Capítulo II	9 días	vie 15/11/13	mié 27/11/13
Programada manualmente	Mapeo de Involucrados	6 días	vie 15/11/13	vie 22/11/13
Programada manualmente	Matriz de análisis involucrados	3 días	lun 25/11/13	mié 27/11/13
Programada manualmente	Capítulo III	9 días	jue 28/11/13	mar 10/12/13
Programada manualmente	Árbol de Problemas	5 días	jue 28/11/13	mié 04/12/13
Programada manualmente	Árbol de Objetivos	4 días	mié 04/12/13	mar 10/12/13
Programada manualmente	Capítulo IV	24 días	mié 11/12/13	lun 13/01/14
Programada manualmente	Matriz de análisis alternativas	5 días	mié 11/12/13	mar 17/12/13
Programada manualmente	Matriz de análisis de impacto de objetivos	10 días	mié 18/12/13	mar 31/12/13
Programada manualmente	Diagrama de estrategias	4 días	mié 01/01/14	lun 06/01/14
Programada manualmente	Matriz de Marco Lógico	5 días	mar 07/01/14	lun 13/01/14
Programada manualmente	Capítulo V	48 días	mar 14/01/14	jue 20/03/14
Programada manualmente	Justificación Técnica	7 días	mar 14/01/14	mié 22/01/14
Programada manualmente	Análisis y Diseño	10 días	jue 23/01/14	mié 05/02/14

Programada manualmente	Desarrollo	26 días	lun 03/02/14	lun 10/03/14
Programada manualmente	Pruebas	12,5 días	mié 05/03/14	vie 21/03/14
Programada manualmente	Capítulo VI	9 días	vie 21/03/14	mié 02/04/14
Programada manualmente	Recursos	2 días	vie 21/03/14	lun 24/03/14
Programada manualmente	Presupuesto	4 días	mar 25/03/14	vie 28/03/14
Programada manualmente	Cronograma	3 días	lun 31/03/14	mié 02/04/14
Programada manualmente	Capítulo VII	3 días	jue 03/04/14	lun 07/04/14
Programada manualmente	Conclusiones	2 días	jue 03/04/14	vie 04/04/14
Programada manualmente	Recomendaciones	2 días	sáb 05/04/14	dom 06/04/14
Programada manualmente	Índices	9 días	lun 07/04/14	jue 17/04/14

8.01 Manual técnico

8.01.01 Instalación de WAMP SERVER 2.4 (Apache, PHP 5, MySql 5)

Descargar instalador de la siguiente dirección web.

- <http://www.wampserver.com/en/>

Al finalizar la descarga ejecutar el archivo y seguir los siguientes pasos.

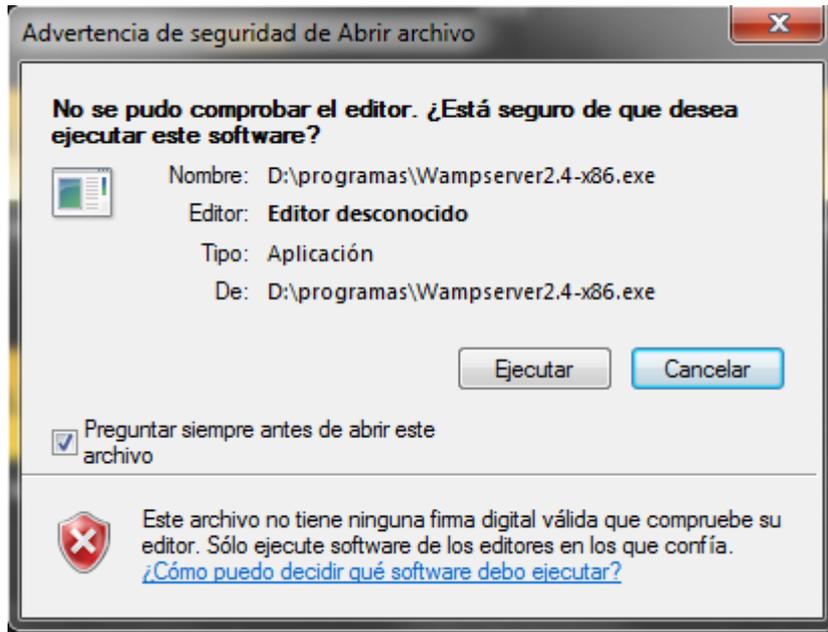


Figura. 46 Advertencia de seguridad: Presionar el botón Ejecutar para abrir el instalador.

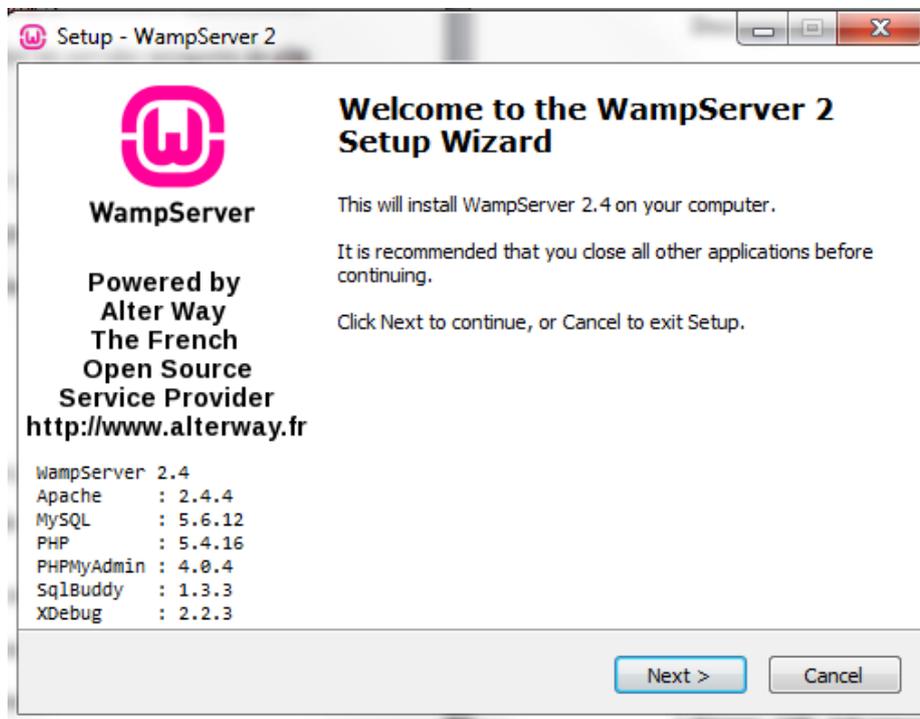


Figura. 47 Inicio de instalación WAMP SERVER 2.4: Presionar en botón Next

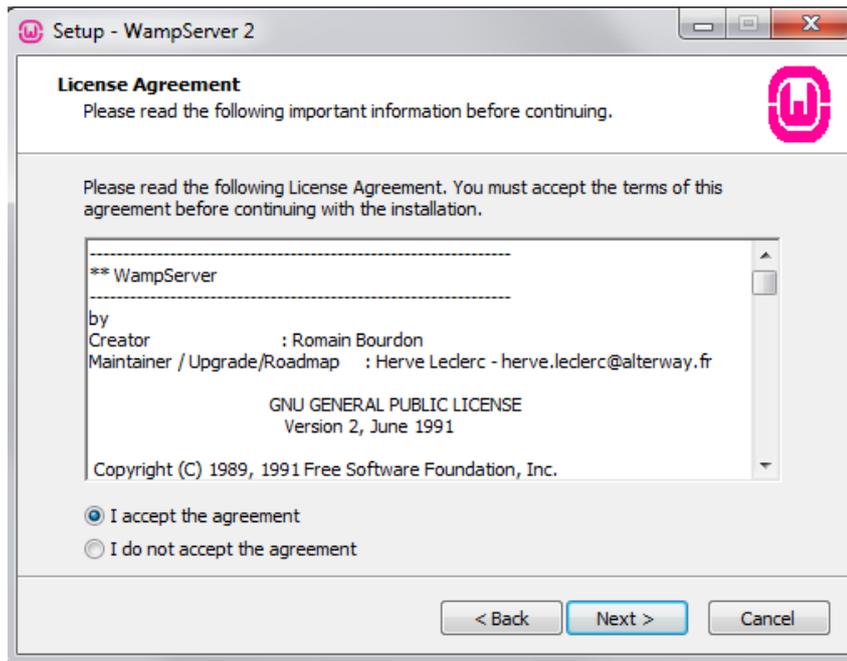


Figura. 48 Contrato de licencia: Seleccionar acepto términos y condiciones y presionar el botón Next

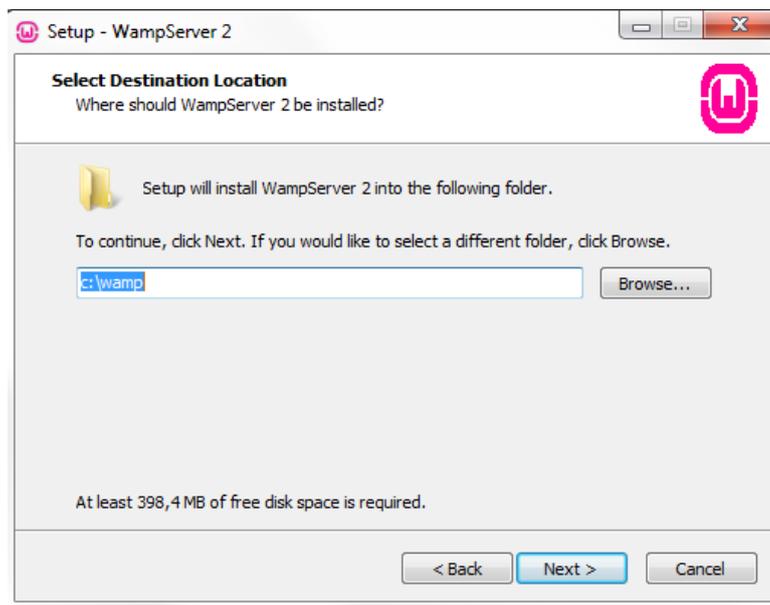


Figura. 49 Instalar en ubicación: Dejar ubicación predeterminada y presionar Next

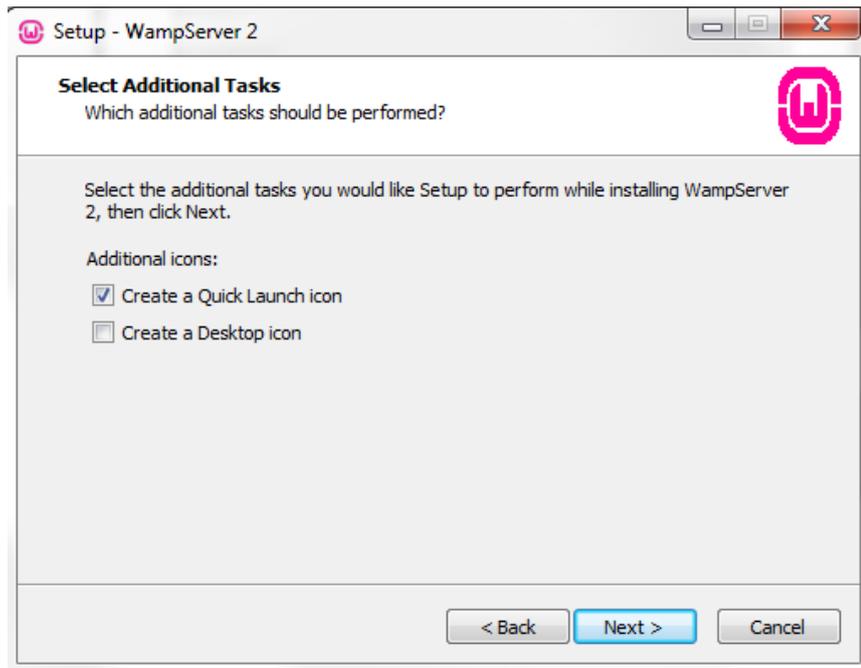


Figura. 50 Seleccionar tareas adicionales: Seleccionar la tarea deseada y presionar Next

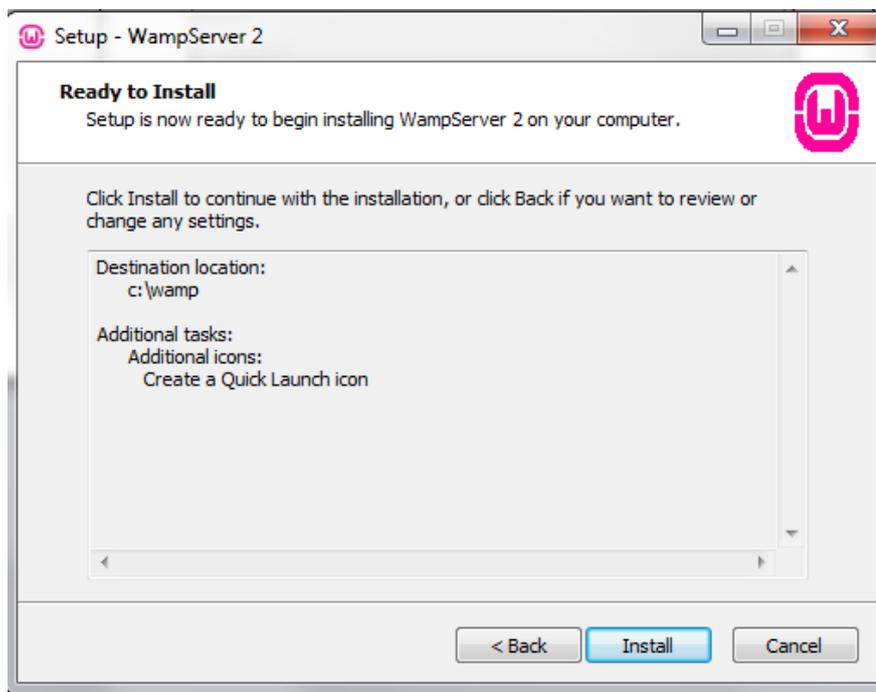


Figura 51. Listo para instalar: Presionar Install

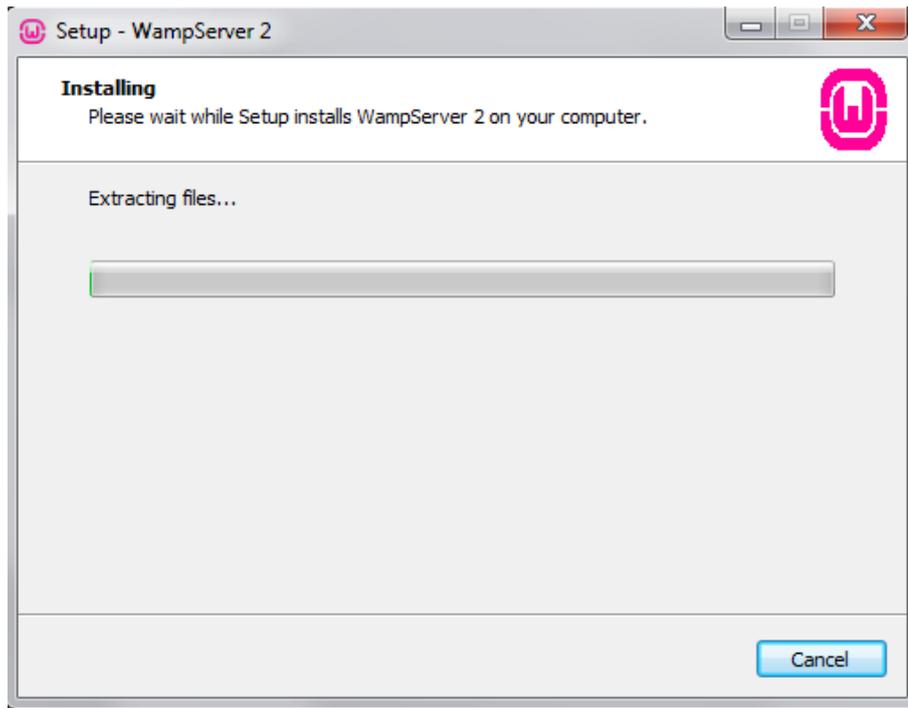


Figura 52. Instalando: Esperar a que la instalación finalice

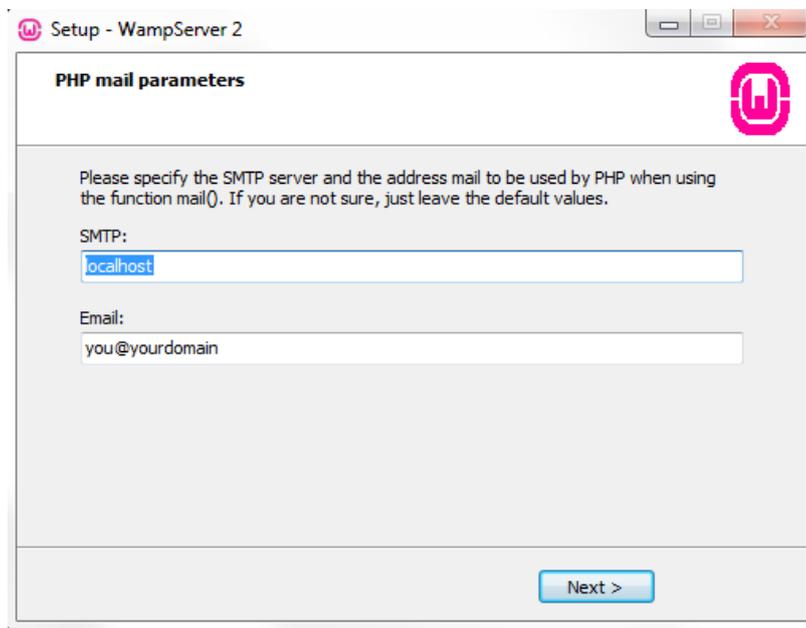


Figura 53. Parámetros de email: Configurar credenciales o dejarlas por default y presionar Next

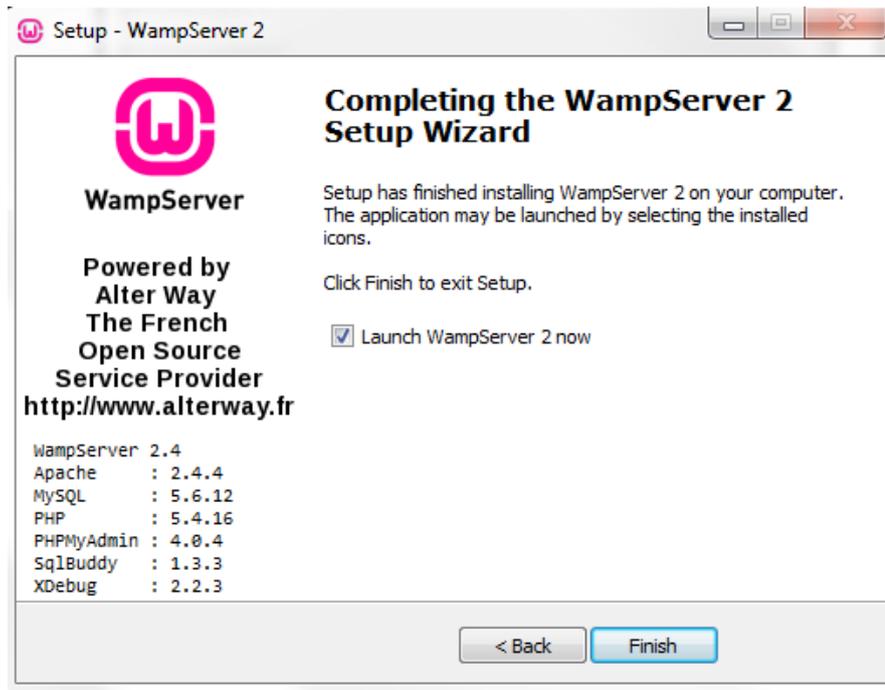


Figura 54. Instalación completa: Presionar Finish



Figura 55. Icono para ejecutar WAMP SERVER: Se debe dar doble click en este icono para iniciar el servicio

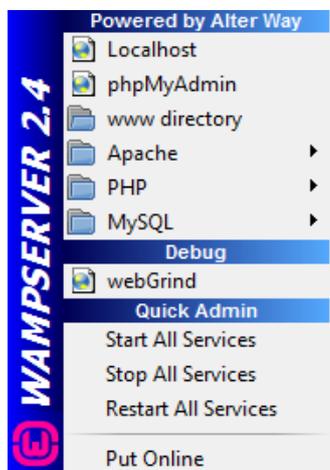


Figura 56. Ventana de administración de servicios: Ir a iconos ocultos en la barra de herramientas de Windows y dar click en el logo de wamp server.

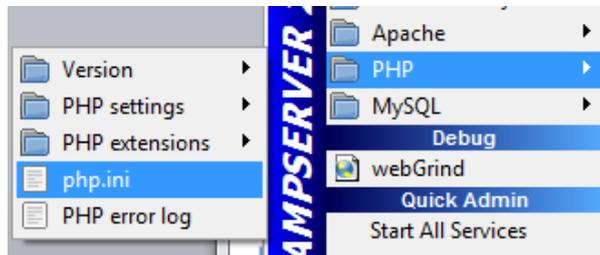


Figura 57. Abrir archivo de configuración de php: Ingresar en el menú PHP y dar click sobre php.ini

```
;extension=php_mssql.dll
extension=php_mysql.dll
extension=php_mysqli.dll
;extension=php_oci8.dll
;extension=php_oci8_11g.dll
extension=php_openssl.dll
```

Figura 58. Extensiones activas: Para el correcto funcionamiento de la herramienta se deben des comentar las extensiones que están sin “;” al principio y reiniciar WAMP SERVER

8.02 Script base de datos MySql

```
CREATE DATABASE `herramienta_integraciones` ;
```

```
USE `herramienta_integraciones`;
```

```
DROP TABLE IF EXISTS `auditoria_cabezera`;
```

```
CREATE TABLE `auditoria_cabezera` (
```

```
  `idAccion` INT(11) NOT NULL AUTO_INCREMENT,
```

```
`idFuente` INT(11) NOT NULL,  
  
`idAuditoriaEjecucion` INT(11) NOT NULL,  
  
`fechaIngreso` DATETIME NOT NULL,  
  
`fechaIntegracion` DATETIME DEFAULT NULL,  
  
`estado` VARCHAR(20) NOT NULL,  
  
PRIMARY KEY (`idAccion`)  
  
) ENGINE=INNODB DEFAULT CHARSET=utf8;  
  
DROP TABLE IF EXISTS `auditoria_detalle`;  
  
CREATE TABLE `auditoria_detalle` (  
  
`idAccion` INT(11) NOT NULL,  
  
`tabla` VARCHAR(100) DEFAULT NULL,  
  
`campo` VARCHAR(100) NOT NULL,  
  
`valor` TEXT NOT NULL,  
  
`valorModificado` TEXT  
  
) ENGINE=INNODB DEFAULT CHARSET=utf8;  
  
DROP TABLE IF EXISTS `auditoria_ejecuciones`;  
  
CREATE TABLE `auditoria_ejecuciones` (  
  
`idAuditoriaEjecucion` INT(11) NOT NULL AUTO_INCREMENT,  
  
`idEjecucion` INT(11) NOT NULL,
```

```
`fechaInicio` DATETIME NOT NULL,  
  
`fechaFin` DATETIME DEFAULT NULL,  
  
`estado` VARCHAR(20) NOT NULL,  
  
PRIMARY KEY (`idAuditoriaEjecucion`)  
  
) ENGINE=INNODB DEFAULT CHARSET=utf8;  
  
DROP TABLE IF EXISTS `auditoria_errores`;  
  
CREATE TABLE `auditoria_errores` (  
  
`idAccion` INT(11) NOT NULL,  
  
`tabla` VARCHAR(100) NOT NULL,  
  
`campo` VARCHAR(100) NOT NULL,  
  
`valor` TEXT,  
  
`error` TEXT NOT NULL  
  
) ENGINE=INNODB DEFAULT CHARSET=utf8;  
  
DROP TABLE IF EXISTS `auditoria_procesos`;  
  
CREATE TABLE `auditoria_procesos` (  
  
`idAuditoriaProceso` INT(11) NOT NULL AUTO_INCREMENT,  
  
`idAccion` INT(11) NOT NULL,  
  
`idProceso` INT(11) NOT NULL,  
  
`tipo` VARCHAR(20) NOT NULL,
```

```
`estado` VARCHAR(20) NOT NULL,  
  
PRIMARY KEY (`idAuditoriaProceso`)  
  
) ENGINE=INNODB DEFAULT CHARSET=utf8;  
  
DROP TABLE IF EXISTS `campos_fuentes`;  
  
CREATE TABLE `campos_fuentes` (  
  
  `idCampoFuente` INT(11) NOT NULL AUTO_INCREMENT,  
  
  `nombreTabla` VARCHAR(100) NOT NULL,  
  
  `nombreCampo` VARCHAR(100) NOT NULL,  
  
  `idFuente` INT(11) NOT NULL,  
  
  PRIMARY KEY (`idCampoFuente`)  
  
) ENGINE=INNODB DEFAULT CHARSET=utf8;  
  
DROP TABLE IF EXISTS `campos_procesos`;  
  
CREATE TABLE `campos_procesos` (  
  
  `idCampoProceso` INT(11) NOT NULL AUTO_INCREMENT,  
  
  `idProceso` INT(11) NOT NULL,  
  
  `nombreTabla` VARCHAR(100) NOT NULL,  
  
  `nombreCampo` VARCHAR(100) NOT NULL,  
  
  `valorCampo` VARCHAR(255) DEFAULT NULL,  
  
  `estado` VARCHAR(20) NOT NULL,
```

```
PRIMARY KEY (`idCampoProceso`)  
  
) ENGINE=INNODB DEFAULT CHARSET=utf8;  
  
DROP TABLE IF EXISTS `ci_sessions`;  
  
CREATE TABLE `ci_sessions` (  
  
    `session_id` VARCHAR(40) NOT NULL DEFAULT '0',  
  
    `ip_address` VARCHAR(45) NOT NULL DEFAULT '0',  
  
    `user_agent` VARCHAR(120) NOT NULL,  
  
    `last_activity` INT(10) UNSIGNED NOT NULL DEFAULT '0',  
  
    `user_data` TEXT NOT NULL,  
  
    PRIMARY KEY (`session_id`),  
  
    KEY `last_activity_idx` (`last_activity`)  
  
) ENGINE=INNODB DEFAULT CHARSET=utf8;  
  
DROP TABLE IF EXISTS `conexiones`;  
  
CREATE TABLE `conexiones` (  
  
    `idConexion` INT(11) NOT NULL AUTO_INCREMENT,  
  
    `nombre` VARCHAR(50) NOT NULL,  
  
    `servidor` VARCHAR(100) NOT NULL,  
  
    `username` VARCHAR(50) NOT NULL,  
  
    `password` VARCHAR(50) DEFAULT NULL,
```

```
`puerto` VARCHAR(50) DEFAULT NULL,  
  
`driver` VARCHAR(50) NOT NULL,  
  
`idUserio` INT(11) NOT NULL,  
  
`tipo` VARCHAR(20) NOT NULL,  
  
PRIMARY KEY (`idConexion`)  
  
) ENGINE=INNODB DEFAULT CHARSET=utf8;  
  
DROP TABLE IF EXISTS `ejecuciones`;  
  
CREATE TABLE `ejecuciones` (  
  
`idEjecucion` INT(11) NOT NULL,  
  
`nombreEjecucion` VARCHAR(100) NOT NULL,  
  
`idUserio` INT(11) NOT NULL,  
  
`idFuente` INT(11) NOT NULL,  
  
`nombreFuente` VARCHAR(100) NOT NULL,  
  
`fechaInicio` DATE NOT NULL,  
  
`horaInicio` TIME NOT NULL,  
  
`fechaFin` DATE DEFAULT NULL,  
  
`horaFin` TIME DEFAULT NULL,  
  
`estado` VARCHAR(20) NOT NULL,  
  
PRIMARY KEY (`idEjecucion`)
```

```
) ENGINE=INNODB DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `fuentes_datos`;

CREATE TABLE `fuentes_datos` (

  `idFuente` INT(11) NOT NULL,

  `nombreFuente` VARCHAR(100) NOT NULL,

  `idUserio` INT(11) NOT NULL,

  `idConexion` INT(11) NOT NULL,

  `nombreConexion` VARCHAR(100) NOT NULL,

  `nombreBd` VARCHAR(100) NOT NULL,

  `nombreTabla` VARCHAR(100) NOT NULL,

  `nombreCampo` VARCHAR(100) NOT NULL,

  `condicion` VARCHAR(100) NOT NULL,

  `valorCondicion` VARCHAR(100) DEFAULT NULL,

  `estado` VARCHAR(20) NOT NULL

) ENGINE=INNODB DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `fuentes_datos_filtros`;

CREATE TABLE `fuentes_datos_filtros` (

  `idFiltro` INT(11) NOT NULL AUTO_INCREMENT,

  `idFuente` INT(11) NOT NULL,
```

```
`nombreTabla` VARCHAR(100) NOT NULL,  
  
`nombreCampo` VARCHAR(100) NOT NULL,  
  
`condicion` VARCHAR(100) NOT NULL,  
  
`valorCondicion` VARCHAR(100) DEFAULT NULL,  
  
`estado` VARCHAR(20) NOT NULL,  
  
PRIMARY KEY (`idFiltro`)  
  
) ENGINE=INNODB DEFAULT CHARSET=utf8;  
  
DROP TABLE IF EXISTS `notificaciones`;  
  
CREATE TABLE `notificaciones` (  
  
`idNotificacion` INT(11) NOT NULL AUTO_INCREMENT,  
  
`idEjecucion` INT(11) NOT NULL,  
  
`nombreDestinatario` VARCHAR(100) NOT NULL,  
  
`emailDestinatario` VARCHAR(100) NOT NULL,  
  
`idProceso` INT(11) NOT NULL,  
  
`resultadoProceso` VARCHAR(20) NOT NULL,  
  
`tipoEmail` VARCHAR(20) NOT NULL,  
  
`estado` VARCHAR(20) NOT NULL,  
  
PRIMARY KEY (`idNotificacion`)  
  
) ENGINE=INNODB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `procesos`;  
  
CREATE TABLE `procesos` (  
  
    `idProceso` INT(11) NOT NULL,  
  
    `nombreProceso` VARCHAR(100) NOT NULL,  
  
    `idFuente` INT(11) NOT NULL,  
  
    `tipo` VARCHAR(20) NOT NULL,  
  
    `ordenEjecucion` INT(11) NOT NULL,  
  
    `idProcesoPadre` INT(11) NOT NULL,  
  
    `nombreProcesoPadre` VARCHAR(100) DEFAULT NULL,  
  
    `ejecutarSi` VARCHAR(20) DEFAULT NULL,  
  
    `idConexion` INT(11) DEFAULT NULL,  
  
    `nombreConexion` VARCHAR(100) DEFAULT NULL,  
  
    `nombreBd` VARCHAR(100) DEFAULT NULL,  
  
    `nombreTabla` VARCHAR(100) NOT NULL,  
  
    `utilizarResultados` INT(11) NOT NULL,  
  
    `estado` VARCHAR(20) NOT NULL,  
  
    PRIMARY KEY (`idProceso`)  
  
) ENGINE=INNODB DEFAULT CHARSET=utf8;  
  
DROP TABLE IF EXISTS `procesos_filtros`;
```

```
CREATE TABLE `procesos_filtros` (  
  
    `idFiltro` INT(11) NOT NULL AUTO_INCREMENT,  
  
    `idProceso` INT(11) NOT NULL,  
  
    `nombreTabla` VARCHAR(100) NOT NULL,  
  
    `nombreCampo` VARCHAR(100) NOT NULL,  
  
    `condicion` VARCHAR(100) NOT NULL,  
  
    `valorCondicion` VARCHAR(100) DEFAULT NULL,  
  
    `estado` VARCHAR(20) NOT NULL,  
  
    PRIMARY KEY (`idFiltro`)  
  
) ENGINE=INNODB DEFAULT CHARSET=utf8;  
  
DROP TABLE IF EXISTS `usuarios`;  
  
CREATE TABLE `usuarios` (  
  
    `idUsuario` INT(11) NOT NULL AUTO_INCREMENT,  
  
    `username` VARCHAR(20) NOT NULL,  
  
    `password` VARCHAR(20) NOT NULL,  
  
    PRIMARY KEY (`idUsuario`)  
  
) ENGINE=INNODB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;  
  
INSERT INTO `usuarios` (`idUsuario`, `username`, `password`) VALUES  
  
(1,'admin','admin');
```

8.03 Diccionario de datos

auditoria_cabezera												
Fields												
Field	Type	Collation	N	K	Defa	Extra	Privileges	Com				
			ull	ey	ult			ment				
idAccion	int(11)	(NULL)	N	P	(NU	auto_incr	select,insert,update					
			O	RI	LL)	ement	,references					
idFuente	int(11)	(NULL)	N		(NU		select,insert,update					
			O		LL)		,references					
idAuditoriaEj ecucion	int(11)	(NULL)	N		(NU		select,insert,update					
			O		LL)		,references					
fechaIngreso	datetim e	(NULL)	N		(NU		select,insert,update					
			O		LL)		,references					
fechaIntegrac ion	datetim e	(NULL)	Y		(NU		select,insert,update					
			ES		LL)		,references					
estado	varchar (20)	utf8_gene ral_ci	N		(NU		select,insert,update					
			O		LL)		,references					
Indexes												
Table	No	Key	Se	Colu	Coll	Cardi	Sub	Pac	N	Ind	Com	Inde
	n	name	q	mn	ation	nality	part	ked	ul	ex	ment	x
	uni		in	nam					l	type		com
	que		in	e								ment
			de									
			x									

auditoria_	0	PRIM	1	idAc	A	0	(NU	(NU	BT
cabezera		ARY		cion			LL)	LL)	RE
									E

auditoria_detalle									
Fields									
Field	Type	Collation	Nu	Ke	Defa	Ext	Privileges	Comm	
			ll	y	ult	ra		ent	
idAccion	int(11)	(NULL)	N		(NUL		select,insert,update,re	ferences	
tabla	varchar(100)	utf8_general_ci	YES		(NUL		select,insert,update,re	ferences	
campo	varchar(100)	utf8_general_ci	NO		(NUL		select,insert,update,re	ferences	
valor	text	utf8_general_ci	NO		(NUL		select,insert,update,re	ferences	
valorModificado	text	utf8_general_ci	YES		(NUL		select,insert,update,re	ferences	

auditoria_ejecuciones									
Fields									
Field	Type	Collation	N	K	Defa	Extra	Privileges	Com	
			ull	ey	ult			ment	

idAuditoriaEjecucion	int(11)	(NULL)	N	P	(NULL)	auto_incr							select,insert,update,references
idEjecucion	int(11)	(NULL)	N		(NULL)								select,insert,update,references
fechaInicio	datetime	(NULL)	N		(NULL)								select,insert,update,references
fechaFin	datetime	(NULL)	YES		(NULL)								select,insert,update,references
estado	varchar(20)	utf8_general_ci	N		(NULL)								select,insert,update,references
Indexes													
Table	No	Key	Seq	Column	Collation	Cardinality	Subpart	Packed	Null	Index type	Comment	Index comment	
auditoria_ejecuciones	0	PRI	1	idAuditoriaEjecuciones	A	0	(N)	(N)		BTREE			
		MA		aEjecuciones			UL	UL		RE			
		RY		n			L)	L)		E			

auditoria_errores													
Fields													
Field	Type	Collation	Null	Key	Default	Extra	Privileges					Comment	
idAccion	int(11)	(NULL)	NO		(NULL)		select,insert,update,references						

tabla	varchar(100)	utf8_general_ci	NO	(NULL)	select,insert,update,references
campo	varchar(100)	utf8_general_ci	NO	(NULL)	select,insert,update,references
valor	text	utf8_general_ci	YES	(NULL)	select,insert,update,references
error	text	utf8_general_ci	NO	(NULL)	select,insert,update,references

auditoria_procesos												
Fields												
Field	Type	Collation	N	K	Default	Extra	Privileges	Comment				
idAuditoria	int(11)	(NULL)	N	P	(NULL)	auto_increment	select,insert,update,references					
idProceso	int(11)	(NULL)	N	O	(NULL)		select,insert,update,references					
idAccion	int(11)	(NULL)	N	O	(NULL)		select,insert,update,references					
idProceso	int(11)	(NULL)	N	O	(NULL)		select,insert,update,references					
tipo	varchar(20)	utf8_general_ci	N	O	(NULL)		select,insert,update,references					
estado	varchar(20)	utf8_general_ci	N	O	(NULL)		select,insert,update,references					
Indexes												
Table	No	Key	Seq	Column	Collation	Cardinality	Subpart	Packed	Null	Index	Comment	Index
	uni		in		n				l			com

qu	in	typ	men
e	de	e	t
	x		
auditoria	0	PRI	1
_proceso	MAR		iaProceso
s	Y		

campos_fuentes												
Fields												
Field	Type	Collation	N	K	Defa	Extra	Privileges	Com				
			ull	ey	ult			ment				
idCampoF	int(11)	(NULL)	N	P	(NU	auto_incr	select,insert,update,					
uente			O	RI	LL)	ement	references					
nombreTa	varchar(utf8_gene	N		(NU		select,insert,update,					
bla	100)	ral_ci	O		LL)		references					
nombreCa	varchar(utf8_gene	N		(NU		select,insert,update,					
mpo	100)	ral_ci	O		LL)		references					
idFuente	int(11)	(NULL)	N		(NU		select,insert,update,					
			O		LL)		references					
Indexes												
Table	No	Key	Se	Column	Coll	Cardi	Sub	Pac	N	Ind	Com	Inde
	n	name	q	name	ation	nality	part	ked	ul	ex	ment	x
	uni		in						l	typ		com
	que		in							e		ment
			de									
			x									

campos	0	PRIM	1	idCamp	A	0	(NU	(NU	BT
_fuentes		ARY		oFuente			LL)	LL)	RE
									E

campos_procesos												
Fields												
Field	Type	Collation	Nu	K	Defa	Extra	Privileges	Com				
			ll	ey	ult				ment			
idCampoP	int(11)	(NULL)	N	P	(NU	auto_incr	select,insert,update,					
roceso			O	RI	LL)	ement	references					
idProceso	int(11)	(NULL)	N		(NU		select,insert,update,					
			O		LL)		references					
nombreTa	varchar(utf8_gene	N		(NU		select,insert,update,					
bla	100)	ral_ci	O		LL)		references					
nombreCa	varchar(utf8_gene	N		(NU		select,insert,update,					
mpo	100)	ral_ci	O		LL)		references					
valorCamp	varchar(utf8_gene	Y		(NU		select,insert,update,					
o	255)	ral_ci	ES		LL)		references					
estado	varchar(utf8_gene	N		(NU		select,insert,update,					
	20)	ral_ci	O		LL)		references					
Indexes												
Table	No	Key	Se	Column	Coll	Cardi	Sub	Pac	N	Ind	Com	Inde
	n	name	q	name	atio	nality	part	ked	ul	ex	ment	x
	uni		in		n				l	typ		com
	que		in							e		men
			de									t
			x									

campos_	0	PRI	1	idCamp	A	0	(N	(N	BT
procesos		MAR		oProces			UL	UL	RE
		Y		o			L)	L)	E

ci_sessions												
Fields												
Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment				
session_id	varchar(40)	utf8_general_ci	NO	PRI	0		select,insert,update,references					
ip_address	varchar(45)	utf8_general_ci	NO		0		select,insert,update,references					
user_agent	varchar(200)	utf8_general_ci	NO		(NULL)		select,insert,update,references					
last_activity	int(10) unsigned	(NULL)	NO	MUL	0		select,insert,update,references					
user_data	text	utf8_general_ci	NO		(NULL)		select,insert,update,references					
Indexes												
Table	Non-unique	Key name	Seq	Column name	Collation	Cardinality	Subpart	Packed	Null	Index type	Comment	Index comment
ci_sessions	0	PRIMARY	1	session_id	A	0		(NULL)	(NULL)	BT		
										RE		E

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
ci_ses	1	last_acti	1	last_a	A	0	(NU (NU BT	
sions		vity_idx		ctivity			LL) LL) RE	E
conexion								
es								
Fields								
idCone	int(11)	(NULL)	N	P	(NU	auto_incre	select,insert,update,r	
xion			O	RI	LL)	ment	eferences	
nombre	varchar(50)	utf8_gener al_ci	N		(NU		select,insert,update,r	
			O		LL)		eferences	
servido	varchar(100)	utf8_gener al_ci	N		(NU		select,insert,update,r	
r			O		LL)		eferences	
userna	varchar(50)	utf8_gener al_ci	N		(NU		select,insert,update,r	
me			O		LL)		eferences	
passwo	varchar(50)	utf8_gener al_ci	Y		(NU		select,insert,update,r	
rd			ES		LL)		eferences	
puerto	varchar(50)	utf8_gener al_ci	Y		(NU		select,insert,update,r	
			ES		LL)		eferences	
driver	varchar(50)	utf8_gener al_ci	N		(NU		select,insert,update,r	
			O		LL)		eferences	
idUsuar	int(11)	(NULL)	N		(NU		select,insert,update,r	
io			O		LL)		eferences	
tipo	varchar(20)	utf8_gener al_ci	N		(NU		select,insert,update,r	
			O		LL)		eferences	
Indexes								

Table	No	Key	Se	Colum	Colla	Cardi	Sub	Pack	N	Inde	Com	Inde
	n	name	q	n	tion	nality	part	ed	ul	x	ment	x
	uni		in	name					l	type		com
	que		ind									ment
			ex									
conexiones	0	PRIMARY	1	idConexion	A	0	(NULL)	(NULL)		BTRE		

ejecuciones												
Fields												
Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment				
idEjecucion	int(11)	(NULL)	No	Primary	(NULL)		select,insert,update,references					
nombreEjecucion	varchar(100)	utf8_general_ci	No		(NULL)		select,insert,update,references					
idUsuario	int(11)	(NULL)	No		(NULL)		select,insert,update,references					
idFuente	int(11)	(NULL)	No		(NULL)		select,insert,update,references					
nombreFuente	varchar(100)	utf8_general_ci	No		(NULL)		select,insert,update,references					
fechaInicio	date	(NULL)	No		(NULL)		select,insert,update,references					
horaInicio	time	(NULL)	No		(NULL)		select,insert,update,references					

fechaFin	date	(NULL)	YE	(NUL	select,insert,update,r							
			S	L)	eferences							
horaFin	time	(NULL)	YE	(NUL	select,insert,update,r							
			S	L)	eferences							
estado	varchar(20)	utf8_general_ci	N	(NUL	select,insert,update,r							
			O	L)	eferences							
Indexes												
Table	No	Key	Se	Colum	Coll	Cardi	Sub	Pac	N	Inde	Com	Inde
	n	name	q	n	ation	nality	part	ked	ul	x	ment	x
	uni		in	name					l	type		com
	que		ind									ment
			ex									
ejecuc	0	PRIM	1	idEjec	A	0	(NU	(NU	BT			
iones		ARY		ucion			LL)	LL)	RE			
									E			

fuentes_datos												
Fields												
Field	Type	Collation	Nu	Ke	Defa	Ext	Privileges	Comm				
			ll	y	ult	ra		ent				
idFuente	int(11)	(NULL)	N		(NUL		select,insert,update,r					
			O		L)		eferences					
nombreFue	varchar(100)	utf8_general_ci	N		(NUL		select,insert,update,r					
n			O		L)		eferences					
idUsuario	int(11)	(NULL)	N		(NUL		select,insert,update,r					
			O		L)		eferences					
idConexion	int(11)	(NULL)	N		(NUL		select,insert,update,r					
			O		L)		eferences					

nombreCon	varchar(100)	utf8_general_ci	N	(NULL)	select,insert,update,references
nombreBd	varchar(100)	utf8_general_ci	N	(NULL)	select,insert,update,references
nombreTabla	varchar(100)	utf8_general_ci	N	(NULL)	select,insert,update,references
nombreCampo	varchar(100)	utf8_general_ci	N	(NULL)	select,insert,update,references
condicion	varchar(100)	utf8_general_ci	N	(NULL)	select,insert,update,references
valorCondicion	varchar(100)	utf8_general_ci	YES	(NULL)	select,insert,update,references
estado	varchar(20)	utf8_general_ci	N	(NULL)	select,insert,update,references

--

fuentes_datos_filtros								
Fields								
Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
idFiltro	int(11)	(NULL)	No	Primary	(NULL)	auto_increment	select,insert,update,references	
idFuente	int(11)	(NULL)	No		(NULL)		select,insert,update,references	
nombreTabla	varchar(100)	utf8_general_ci	No		(NULL)		select,insert,update,references	
nombreCampo	varchar(100)	utf8_general_ci	No		(NULL)		select,insert,update,references	

condicion	varchar(100)	utf8_general_ci	N	(NULL)									select,insert,update,references
valorCondicion	varchar(100)	utf8_general_ci	YES	(NULL)									select,insert,update,references
estado	varchar(20)	utf8_general_ci	N	(NULL)									select,insert,update,references
Indexes													
Table	No	Key	Seq	Column	Collation	Cardinality	Subpart	Packed	Null	Index type	Comment	Index comment	
fuentes_datos_filtros	0	PRIMARY	1	idFiltro	A	0	(NULL)	(NULL)	BTREE				

notificaciones													
Fields													
Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment					
idNotificacion	int(11)	(NULL)	NO	PRIMARY	(NULL)	auto_increment	select,insert,update,references						
idEjecucion	int(11)	(NULL)	NO		(NULL)		select,insert,update,references						
nombreDestinatario	varchar(100)	utf8_general_ci	NO		(NULL)		select,insert,update,references						

emailDestinatario	varchar(100)	utf8_general_ci	N	(NULL)	(NULL)	select,insert,update,references
idProceso	int(11)	(NULL)	N	(NULL)	(NULL)	select,insert,update,references
resultadoProceso	varchar(20)	utf8_general_ci	N	(NULL)	(NULL)	select,insert,update,references
tipoEmail	varchar(20)	utf8_general_ci	N	(NULL)	(NULL)	select,insert,update,references
estado	varchar(20)	utf8_general_ci	N	(NULL)	(NULL)	select,insert,update,references

Indexes

Table	No	Key	Seq	Column	Collation	Cardinality	Subpart	Packed	Null	Index type	Comment	Index comment
notificaciones	0	PRIMARY	1	idNotificacion	A	0	(NULL)	(NULL)		BTREE		

procesos

Fields

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
idProceso	int(11)	(NULL)	N	PRI	(NULL)		select,insert,update,references	

nombreProceso	varchar(100)	utf8_general_ci	N	(NO)	select,insert,update,references							
idFuente	int(11)	(NULL)	N	(NO)	select,insert,update,references							
tipo	varchar(20)	utf8_general_ci	N	(NO)	select,insert,update,references							
ordenEjecucion	int(11)	(NULL)	N	(NO)	select,insert,update,references							
idProcesoPadre	int(11)	(NULL)	N	(NO)	select,insert,update,references							
nombreProcesoPadre	varchar(100)	utf8_general_ci	YES	(NO)	select,insert,update,references							
ejecutarSi	varchar(20)	utf8_general_ci	YES	(NO)	select,insert,update,references							
idConexion	int(11)	(NULL)	YES	(NO)	select,insert,update,references							
nombreConexion	varchar(100)	utf8_general_ci	YES	(NO)	select,insert,update,references							
nombreBd	varchar(100)	utf8_general_ci	YES	(NO)	select,insert,update,references							
nombreTabla	varchar(100)	utf8_general_ci	N	(NO)	select,insert,update,references							
utilizarResultados	int(11)	(NULL)	N	(NO)	select,insert,update,references							
estado	varchar(20)	utf8_general_ci	N	(NO)	select,insert,update,references							
Indexes												
Table	Index	Key name	Seq	Column name	Collation	Cardinality	Subpart	Packed	Null	Index type	Comment	Index

uni	ind	com
que	ex	ment
proc	0 PRIM 1 idPro A 0 (NU (NU BT	
esos	ARY ceso	LL) LL) REE

procesos_filtros												
Fields												
Field	Type	Collation	Nu	K	Defa	Extra	Privileges	Com				
			ll	ey	ult			ment				
idFiltro	int(11)	(NULL)	N	P	(NU	auto_incr	select,insert,update,					
			O	RI	LL)	ement	references					
idProceso	int(11)	(NULL)	N		(NU		select,insert,update,					
			O		LL)		references					
nombreTa	varchar(utf8_gene	N		(NU		select,insert,update,					
bla	100)	ral_ci	O		LL)		references					
nombreC	varchar(utf8_gene	N		(NU		select,insert,update,					
ampo	100)	ral_ci	O		LL)		references					
condicion	varchar(utf8_gene	N		(NU		select,insert,update,					
	100)	ral_ci	O		LL)		references					
valorCon	varchar(utf8_gene	Y		(NU		select,insert,update,					
dicion	100)	ral_ci	ES		LL)		references					
estado	varchar(utf8_gene	N		(NU		select,insert,update,					
	20)	ral_ci	O		LL)		references					
Indexes												
Table	No	Key	Se	Col	Coll	Cardi	Sub	Pac	N	Inde	Com	Inde
	n	name	q	umn	ation	nality	part	ked	ul	x	ment	x
	uni		in	nam					l	type		com
	que			e								ment

ind									
ex									
procesos	0	PRIM	1	idFi	A	0	(NU	(NU	BT
_filtros		ARY		ltro			LL)	LL)	RE
									E

usuarios												
Fields												
Field	Type	Collation	Nu	Ke	Defa	Extra	Privileges	Comm				
			ll	y	ult			ent				
idUsu	int(11)	(NULL)	N	P	(NU	auto_incre	select,insert,update,r					
ario			O	RI	LL)	ment	ferences					
userna	varchar	utf8_gener	N		(NU		select,insert,update,r					
me	(20)	al_ci	O		LL)		ferences					
passw	varchar	utf8_gener	N		(NU		select,insert,update,r					
ord	(20)	al_ci	O		LL)		ferences					
Indexes												
Tabl	No	Key	Se	Colu	Colla	Cardin	Sub	Pack	N	Inde	Com	Inde
e	n	name	q	mn	tion	ality	part	ed	ull	x	ment	x
		uni	in	name						type		com
		que	ind									ment
			ex									
usua	0	PRIM	1	idUsu	A	1	(NU	(NU	BT			
rios		ARY		ario			LL)	LL)	REE			