

Universidad de las Ciencias Informáticas
Facultad 1



Servicios de comunicación de sistemas clientes con el
AFIS Civil de SAIME,
en software libre

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autores: Yilian Rodríguez Grille

Yunierki Verdecia Viltres

Tutor: Ing. Yurdik Cervantes Mendoza

Ciudad Habana, 27 de junio de 2008

Año 50 de la Revolución

"Sean capaces siempre de sentir, en lo más hondo, cualquier injusticia realizada contra cualquiera, en cualquier parte del mundo. Es la cualidad más linda del revolucionario."

Ernesto Che Guevara.

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Yilian Rodríguez Grille

Yurdik Cervantes Mendoza

Yunierki Verdecia Viltres

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Servicios de comunicación de sistemas clientes con el AFIS Civil de SAIME, en software libre.

Autor(es): Yilian Rodríguez Grille

Yunierki Verdecia Viltres

Ing. Yurdik Cervantes Mendoza

Firma

Fecha

AGRADECIMIENTOS

A mis padres, tías y familia en general por apoyarme y confiar en mí.

*A mis amigas Arye, Lassie, Yaisel, Lisbeth y amigos Agni, Elpidio, Erik, por estar siempre ahí,
dispuestos para mí.*

A mi amor H₂O, por hacerme CRECER.

A Rikki, Ismy, Nuñes, Yoanna, Ana, por acompañarme durante estos años de carrera.

Al 1105, por ser un grupo único.

A la UCI, por darme malos y buenos momentos.

*A todos los profesores que me enseñaron en esta universidad, y a mi tutor que ayudó en la
confeción de esta tesis.*

A mi comandante Fidel, por darme esta oportunidad con esta Revolución.

*A todos los que de una manera u otra estuvieron conmigo durante este largo camino que se
tornó interminable.*

YLLITS

A mi familia por apoyarme en todos mis años de estudio y por la confianza que tienen en mí.

*A todos los integrantes del proyecto Identidad que de alguna u otra manera me han ayudado,
principalmente a mi tutor Yurdik Cervantes a pesar de encontrarse en Venezuela y que ha sido
mi guía para el desarrollo de mis conocimientos desde que empecé a formar parte del
proyecto Identidad.*

A mis amigos Hubert y Alexis por ayudarme con la tesis.

A mi amigo Diuber y a su novia por ser tan preocupados.

A mi amigo Dayron y a su mamá por apoyarme en todo este tiempo.

A todos los compañeros y amigos que de alguna otra forma me preguntaron por la tesis.

*A la dirección de la Revolución y al comandante en jefe Fidel Castro por construir una
universidad como la UCI.*

A la UCI por formarme como un profesional.

YUNTERKI

DEDICATORIA

A mi mamá Riselia y mi papá Linardo.

YLLITS

A mis padres queridos por guiarme por el camino correcto, por la confianza que depositaron en mí y por ser lo máspreciado que tengo.

A mis abuelos del alma Will y Martha por su cariño que me han dado y por ser mis ejemplos.

A mi hermanita linda por quererme tanto.

A mis tíos y tías por todos los consejos que me dieron.

A mis queridos hermanos Liubar y Ray

A Peki, los Mellis y tía Yola por apoyarme siempre.

A toda mi familia y mis amistades

YUNTERKI

RESUMEN

El AFIS Civil (Sistema Automatizado de Identificación de Huellas Dactilares) en la República Bolivariana de Venezuela, tiene una comunicación asincrónica con las aplicaciones que están siendo desarrolladas con SAIME (Servicio Autónomo de Identificación Migración y Extranjería). Hoy en día solo estas aplicaciones se benefician de las funcionalidades de este sistema, para la identificación biométrica de los ciudadanos. Los sistemas o aplicaciones clientes externos a SAIME pueden beneficiarse de estas funcionalidades siempre que posean servicios que permitan una comunicación con el AFIS.

Basados en la necesidad de difundir las funcionalidades del AFIS hacia todos los sistemas clientes que lo requieran, se resuelve desarrollar varios servicios. Dentro de estos se encuentran los servicios web, que son los encargados de crear las verificaciones y llevar un control de las mismas y por otra parte los servicios o demonios que son los encargados de realizar toda la comunicación con el AFIS Civil para el envío de los datos biométricos de una persona, para conocer su identidad. De esta forma se obtiene una solución que puede ser usada por muchos clientes y que tiene como objetivo conocer o verificar la identidad de los ciudadanos.

En este trabajo de diploma se explican conceptos relacionados con el tema, se hace un análisis del sistema a desarrollar y se expresan los resultados del estudio realizado para el desarrollo e implementación de los servicios web y los servicios de comunicación.

PALABRAS CLAVES

AFIS: Sistema Automatizado de Identificación de Huellas Dactilares.

SAIME: Servicio Autónomo de Identificación, Migración y Extranjería.

Servicio Web: Colección de protocolos y estándares que sirve para intercambiar datos entre aplicaciones.

Servicios o Demonios: Proceso informático que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario.

TABLA DE CONTENIDOS

AGRADECIMIENTOS..... V

DEDICATORIA VI

RESUMEN..... VII

INTRODUCCIÓN..... 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA..... 5

 1.1 Introducción 5

 1.2 Biometría 5

 1.2.1 Identificación por huellas digitales 6

 1.2.2 Componentes biométricos..... 6

 1.2.3 Dispositivos biométricos..... 7

 1.2.4 Estándares asociados a tecnologías biométricas..... 8

 1.2.5 Procesos de Autenticación e Identificación biométrica 9

 1.3 Soluciones existentes sobre identificación biométrica 9

 1.3.1 Plataforma Biométrica HOMINI 10

 1.3.2 MegaMatcher SDK..... 11

 1.3.3 Biometric SDK 11

 1.4 El AFIS 12

 1.4.1 Sistema Automatizado de Identificación de Huellas Dactilares (AFIS) 12

 1.4.2 La evolución del AFIS 12

 1.4.3 AFIS civil y criminal 12

 1.4.4 Arquitectura de los sistemas AFIS..... 13

 1.5 Servicios de comunicación con el AFIS Civil de SAIME 13

 1.5.1 Servicio Local AFIS (SLA) 14

 1.5.2 Servicio Central AFIS (SCA) 14

 1.5.3 Arquitectura interna e Interacción entre los servicios de comunicación 16

 1.6 Servicios Web 17

 1.6.1 ¿Qué son los Servicios Web? 17

 1.6.2 ¿Para qué sirven?..... 18

 1.6.3 Estándares que realizan el intercambio de mensajes en los Servicios Web 18

 1.6.3.1 XML (Extensible Markup Language) 18

 1.6.3.2 SOAP (Simple Object Access Protocol) 19

 1.6.3.3 WSDL (Web Services Description Language) 19

 1.6.3.4 NuSOAP..... 19

 1.7 Servicios o demonios 20

 1.7.1 Características..... 20

1.8	Estándares y protocolos de Internet	21
1.8.1	MIME (Multipurpose Internet Mail Extensions).....	21
1.8.2	IMAP (Internet Message Access Protocol).....	21
1.8.3	SMTP (Simple Mail Transfer Protocol).....	22
1.9	Tendencias y tecnologías actuales	22
1.9.1	RUP como metodología de desarrollo de software	23
1.9.2	UML como lenguaje de modelado de sistemas de software	23
1.9.3	Herramienta Visual Paradigm.....	24
1.9.4	Lenguaje de programación para los Servicios Web: PHP	24
1.9.5	Lenguaje de programación Python	25
1.9.6	Plataforma y entorno de desarrollo. Ubuntu y Eclipse.....	26
1.9.7	Servidor web Apache	26
1.9.8	Servidores de correos	27
1.9.9	PostgreSQL como gestor de Base de Datos.....	28
1.10	Conclusiones	28
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA		30
2.1	Introducción	30
2.2	Descripción de los procesos del negocio	30
2.2.1	Breve Descripción Del Negocio	30
2.2.2	Problemas fundamentales detectados.....	31
2.3	Modelo de Dominio	31
2.4	Diagrama de Clases del Modelo de Dominio	32
2.5	Glosario de Términos del Modelo de Dominio.....	32
2.6	Sistema.....	34
2.6.1	Propuesta del sistema.....	34
2.7	Especificación de requisitos	35
2.7.1	Requerimientos Funcionales.....	35
2.7.2	Requisitos No Funcionales.....	38
2.8	Diagrama de Casos de Usos de Sistema	40
2.9	Definición de los casos de usos del sistema	41
2.9.1	Actores del sistema.....	41
2.10	Descripción Textual de CUS.....	42
2.11	Conclusiones	46
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....		47
3.1	Introducción	47
3.2	Análisis	47

3.2.1	Modelo de clases del análisis.....	47
3.3	Diseño	48
3.3.1.	Modelo de clases del diseño	49
3.3.1.1.	Diagramas de clases del diseño por paquetes del Servicio Web	50
3.3.1.1.	Descripción de las clases del diseño del Servicio Web.....	53
3.3.1.2.	Diagrama de secuencia del diseño	61
3.3.2.	Descripción de la arquitectura de los Servicios de Comunicación	61
3.3.2.1.	Modelo de diseño de los Servicios de Comunicación	62
3.3.3.	Diseño de la Base de Datos	71
3.3.3.1.	Descripción de las tablas del modelo de datos.....	74
3.4.	Conclusiones	77
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.....		78
4.1.	Introducción	78
4.2.	Diagrama de Despliegue	78
4.2.1.	Descripción del Diagrama de Despliegue	79
4.3.	Modelo de implementación.....	79
4.3.1.	Diagrama de Componentes.....	79
4.3.2.	Descripción de los paquetes de componentes.....	84
4.1.	Prueba	85
4.1.1.	Prueba de Caja Negra	85
4.1.2.	Prueba de Caja Blanca	86
4.1.	Conclusiones	88
CONCLUSIONES		89
RECOMENDACIONES		90
BIBLIOGRAFÍA CITADA		91
BIBLIOGRAFIA CONSULTADA		93
GLOSARIO.....		94
ANEXOS I		97
ANEXOS II.....		99
ANEXOS III.....		105

ÍNDICE DE TABLAS

Tabla 1 Descripción de los actores del sistema	41
Tabla 2: Descripción del CUS "Crear Verificación"	42
Tabla 3: Descripción del CUS "Enviar Solicitudes"	43
Tabla 4: Descripción del CUS "Recibir Transacciones"	44
Tabla 5: Descripción de Clase interfaz. CI_Cliente	53
Tabla 6: Descripción de Clase interfaz. CI_Administrador	54
Tabla 7: Descripción de Clase controladora. CC_Transacción.....	54
Tabla 8: Descripción de Clase controladora. CC_Verificación	55
Tabla 9: Descripción de Clase controladora. Conexión.....	55
Tabla 10: Descripción de Clase controladora. Dao_Solicitud.....	56
Tabla 11: Descripción de Clase controladora. Dao_Transaccion	56
Tabla 12: Descripción de Clase entidad. Transacción	57
Tabla 13: Descripción de Clase entidad. Solicitud	58
Tabla 14: Descripción de Clase entidad. Respuesta.....	58
Tabla 15: Descripción de Clase entidad. Verificación	59
Tabla 16: Descripción de Clase entidad. Imagen.....	59
Tabla 17: Descripción de la clase del diseño del paquete de Persistencia. ConexionCent	65
Tabla 18: Descripción de la clase del diseño del paquete de Persistencia. Conexión.....	66
Tabla 19: Descripción de la clase del diseño del paquete de Persistencia. LectorDB.....	66
Tabla 20: Descripción de la clase del diseño del paquete de Persistencia. EscritorDB.....	67
Tabla 21: Descripción de la clase del diseño del paquete de Persistencia. LectorXML	68
Tabla 22: Descripción de la clase del diseño del paquete de Persistencia. EscritorXML	68
Tabla 23: Descripción de la clase del diseño del paquete de Persistencia. LectorIMAP	68
Tabla 24: Descripción de la clase del diseño del paquete de Persistencia. EscritorSMTP.....	69
Tabla 25: Descripción de la clase del diseño del paquete de Persistencia. ServicioLocal.....	69
Tabla 26: Descripción de la clase del diseño del paquete de Persistencia. ServicioCentral.....	69
Tabla 27: Descripción de la tabla de nTipoVerificación del modelo entidad-relación	74
Tabla 28: Descripción de la tabla de dVerificaciónAfis del modelo entidad-relación	75
Tabla 29: Descripción de la tabla de dSolicitudAfis del modelo entidad-relación	75
Tabla 30: Descripción de la tabla de dRespuestaAfis del modelo entidad-relación.....	76
Tabla 31: Descripción de la tabla de dImagenTranAfis del modelo entidad-relación	76
Tabla 32: Descripción de la tabla de nOrigenDatos del modelo entidad-relación	77
Tabla 33: Descripción de caso de prueba para el CU "Crear Verificación"	85
Tabla 34: Descripción de caso de prueba para el CU "Enviar Solicitud"	85

Tabla 35: Descripción de caso de prueba para el CU “Recibir Transacciones”	86
Tabla 36: Operacionalización de las Variables	97
Tabla 37: Recursos humanos.....	98
Tabla 38: Recursos materiales.....	98
Tabla 39: Descripción del CUS “Mostrar Listado de Verificaciones”	99
Tabla 40: Descripción del CUS “Generar Listado de Transacciones”	100
Tabla 41: Descripción del CUS “Generar Reportes de Solicitudes”	101
Tabla 42: Descripción CUS “Solicitar Envío de Solicitud SRQ”	102
Tabla 43: Descripción del CUS “Recibir Respuestas AFIS”	102
Tabla 44: Descripción del CUS “Enviar Transacciones”	103
Tabla 45: Descripción de Clase controladora. Validator.....	105
Tabla 46: Descripción de la tabla de nTipoTransAfis del modelo entidad-relación.....	110
Tabla 47: Descripción de la tabla de dSolicIdAtAFIS del modelo entidad-relación	110
Tabla 48: Descripción de la tabla de dSolicIRQAfis del modelo entidad-relación	110
Tabla 49: Descripción de la tabla de dSolicSRQAfis del modelo entidad-relación.....	110
Tabla 50: Descripción de la tabla de nServicioAfis del modelo entidad-relación.....	110
Tabla 51: Descripción de la tabla de dSolicImagenAfis del modelo entidad-relación	111
Tabla 52: Descripción de la tabla de nEstadoTranAfis del modelo entidad-relación	111
Tabla 53: Descripción de la tabla de dResplmagenAfis del modelo entidad-relación	111
Tabla 54: Descripción de la tabla de dTRespErrAfis del modelo entidad-relación	111
Tabla 55: Descripción de la tabla de nRespEstado del modelo entidad-relación	111
Tabla 56: Descripción de la tabla de dTRespSTRAfis del modelo entidad-relación.....	112
Tabla 57: Descripción de la tabla de dTRespACKAfis del modelo entidad-relación.....	112
Tabla 58: Descripción de la tabla de dTRespEIRAfis del modelo entidad-relación	112
Tabla 59: Descripción de la tabla de dTRespEARAfis del modelo entidad-relación.....	112
Tabla 60: Descripción de la tabla de dTRespIMRAfis del modelo entidad-relación.....	113

ÍNDICE DE FIGURAS

Figura 1.1: Interacción simplificada de una Oficina Regional con el AFIS Civil.....	14
Figura 1.2 Forma general del proceso de verificación biométrica.....	15
Figura 1.3: Modelo de interacción entre dos servicios de comunicación AFIS.....	16
Figura 2.1: Modelo de Dominio	32
Figura 2.2: Propuesta del Sistema	35
Figura 2.3: Diagrama de CUS para el paquete Servicio de Interfaz con el Cliente y Administrador (Servicio Web)...	40
Figura 2.4: Diagrama de CUS del paquete Servicio de Comunicación AFIS (SCA).....	41
Figura 2.5: Diagrama de CUS del paquete Servicio de Distribución AFIS (SDA).....	41
Figura 3.1: Diagramas de clases del análisis para CUS “Crear Verificación” del paquete Servicio de Interfaz con el Cliente y Administrador (Servicio Web)	47
Figura 3.2: Diagramas de clases del análisis del paquete Servicio de Comunicación AFIS.....	48
Figura 3.3: Diagramas de clases del análisis del paquete Servicio de Distribución AFIS.....	48
Figura 3.4: Diagrama de paquetes que agrupa las clases del diseño para el Servicio Web.....	49
Figura 3.5: Diagrama de clases del diseño paquete de Presentación, CI_Cliente	50
Figura 3.6: Diagrama de clases del diseño paquete de Presentación, CI_Administrador.....	50
Figura 3.7: Diagrama de clases del diseño paquete de Entidades	51
Figura 3.8: Diagrama de clases del diseño paquete de Acceso a Datos	52
Figura 3.9: Diagrama de clases del diseño paquete de Gestión de Interfaz, clases controladora	52
Figura 3.10: Diagrama de secuencia del diseño del caso de uso Crear Verificación	61
Figura 3.11 Arquitectura de comunicación interna entre los servicios de comunicación	62
Figura 3.12 Diagrama de Paquetes de los Servicios de Comunicación.....	63
Figura 3.13 Diagrama de clases del diseño del paquete Entidades de los Servicios de Comunicación	64
Figura 3.14 Diagrama de clases del diseño del paquete Persistencia de los Servicios de Comunicación.....	65
Figura 3.15 Diagrama de clases del diseño del paquete Servicios de los Servicios de Comunicación.....	65
Figura 3.16: Diagrama de secuencia del diseño del caso de uso “Enviar Solicitudes AFIS”	70
Figura 3.17: Diagrama de secuencia del diseño del caso de uso “Recibir Respuestas AFIS”	71
Figura 3.18: Diagrama de modelo de datos, (parte I).....	72
Figura 3.19: Diagrama de modelo de datos, (parte II).....	73
Figura 3.20 Diagrama de modelo de datos, (parte III).....	74
Figura 4.1: Diagrama de despliegue.....	78
Figura 4.2: Diagrama de Paquetes de Componentes del servicio web.....	80

Figura 4.3 Diagrama de paquetes de componentes de los Servicios de Comunicación	80
Figura 4.4: Diagrama de componentes para el paquete de clases interfaces de WS_Cliente.....	81
Figura 4.5: Diagrama de componentes para el paquete de clases interfaces de WS_Administrador	81
Figura 4.6: Diagrama de componentes para el paquete de gestión de interfaz (Servicio Web).....	81
Figura 4.7: Diagrama de componentes para el paquete Acceso a Datos (Servicio Web).....	82
Figura 4.8: Diagrama de componentes para el paquete clases entidades (Servicio Web).....	82
Figura 4.9 Diagrama de componentes para el paquete Persistencia de los Servicios de Comunicación.....	83
Figura 4.10 Diagrama de componentes para el paquete Transacciones de los Servicios de Comunicación.....	83
Figura 4.11 Diagrama de componentes para el paquete Servicios de los SC.....	84
Figura 4.12 Código para el CU Crear Verificación del Servicio web	87
Figura II.1 Diagrama de paquetes del sistema.....	99
Figura III.1 Diagrama de clases del análisis para el sub-paquete servicio web con el cliente	105
Figura III.2: Diagrama de clases del análisis para el paquete de servicios web	105
Figura III.3: Diagrama de clases del análisis para el sub-paquete servicio web con el administrador	105
Figura III.4 Diagrama de secuencia del diseño del caso de uso “Mostrar Listado de Transacciones”. Escenario Obtener Solicitudes.....	107
Figura III.5: Diagrama de secuencia del diseño del caso de uso “Mostrar Listado de Transacciones”. Escenario Obtener Respuestas	107
Figura III.6 Diagrama de secuencia del diseño del caso de uso “Mostrar Listado de Verificaciones”	107
Figura III.7: Diagrama de secuencia del diseño del caso de uso “Obtener Resultados de Verificaciones”	108
Figura III.8: Diagrama general del modelo entidad relación (modelo de datos)	109

INTRODUCCIÓN

El Ministerio del Poder Popular para las Relaciones del Interior y Justicia (MPPRIJ) de la República Bolivariana de Venezuela, ha implantado un sistema biométrico AFIS (Sistema Automatizado de Identificación de Huellas Dactilares), para garantizar la integridad de los procesos internos del país que necesitan de mayor seguridad y confiabilidad. Dentro de estos se encuentran los procesos de Identificación, Migración y Extranjería, que están siendo automatizados con el desarrollo del proyecto SAIME (Servicio Autónomo de Identificación, Migración y Extranjería).

La función principal del AFIS Civil es aumentar el nivel de seguridad y confiabilidad de dichos procesos, al detectar de forma automatizada los intentos de usurpación de identidad y/o de múltiples registros de una misma persona bajo diferentes identidades; de modo que permite conocer o verificar la identidad de los ciudadanos mediante sus datos biométricos dactilares.

Para utilizar los servicios de verificación de huellas contra el AFIS Civil, se necesita que los sistemas o aplicaciones clientes capturen los datos biométricos dactilares (huellas) de los usuarios, verifiquen su calidad y se comuniquen de manera asincrónica con este. Como parte de esta comunicación deben ser enviados mensajes al AFIS, llamados "Solicitudes AFIS" y este debe recibirlas, procesarlas y enviar el resultado de vuelta, mediante mensajes llamados "Respuestas AFIS". Ambos tipos de mensajes son referidos como "Transacciones AFIS" y suelen utilizarse por SAIME para el transporte de huellas a otros procesos distantes y participar de esta manera en flujos alternativos intermedios a toda esta comunicación.

Las aplicaciones de SAIME que se encuentran distribuidas por las distintas Oficinas Regionales del país, cuentan con un conjunto de componentes biométricos, servicios de comunicación y herramientas de configuración que regulan todo el funcionamiento provisto, permitiendo la captura y envío de las huellas del ciudadano hacia el AFIS Civil para su inclusión. Una vez presentes estos datos biométricos en el AFIS, este puede brindarle al sistema SAIME la funcionalidad necesaria para la seguridad de sus procesos. Los sistemas externos pueden beneficiarse de estas funcionalidades siempre que tengan componentes y servicios similares.

El ministerio desea ofrecer las funcionalidades del AFIS Civil a otros sistemas externos a SAIME, en aras de contribuir a la informatización de la sociedad y con esto elevar el nivel de vida de los ciudadanos. De esta manera estos sistemas podrán mandar transacciones al AFIS para conocer o verificar la identidad de un ciudadano. Por esta razón es necesario desarrollar un conjunto de servicios de comunicación, que permitan a cualquier sistema externo a SAIME enviar solicitudes al AFIS Civil, y obtener las repuestas del mismo.

Estos servicios no existen en software libre ni en propietario. Se considera su desarrollo en software libre considerando las necesidades expuestas en el decreto 3390 publicado en la gaceta oficial venezolana N° 38.095 de fecha 28/12/2004 y como claramente especifica el Artículo 1. “La Administración Pública Nacional empleará prioritariamente Software Libre desarrollado con Estándares Abiertos, en sus sistemas, proyectos y servicios informáticos. A tales fines, todos los órganos y entes de la Administración Pública Nacional iniciarán los procesos de migración gradual y progresiva de éstos hacia el Software Libre desarrollado con Estándares Abiertos.”

Dada la situación existente, el presente trabajo tiene el fin de dar respuesta al siguiente problema científico:

¿Cómo automatizar en Software Libre servicios que permitan la transmisión de los datos biométricos dactilares de un usuario desde un sistema cliente hacia el AFIS Civil de SAIME?

Por lo tanto el objeto de estudio son los servicios de comunicación, y el campo de acción: son los servicios de comunicación para la transmisión de los datos biométricos en Software Libre.

Para dar solución a la situación científica existente se traza como objetivo general: desarrollar servicios de comunicación de sistemas clientes con el AFIS Civil de SAIME, en Software Libre.

Teniendo en cuenta el objetivo general se enuncian los siguientes objetivos específicos:

- Implementar una interfaz de comunicación con los sistemas clientes para permitirles crear las solicitudes y monitorear su estado (servicio web).
- Implementar un servicio que permita el envío de solicitudes desde los sistemas clientes hacia el servicio central.
- Implementar un servicio central que permita la recepción y reenvío de las solicitudes hacia al AFIS Civil de SAIME.
- Implementar una interfaz para el monitoreo de las solicitudes y respuestas a nivel central.

Para dar cumplimiento a los objetivos específicos se proponen las siguientes tareas:

- Investigar acerca de los estándares y formatos MIME, WSQ y sobre los protocolos de internet IMAP y SMTP.
- Realizar un estudio acerca de los servicios Web, XML, SOAP y las tecnologías relacionadas.
- Realizar un estudio acerca de los servicios o demonios que corren en sistemas UNIX.
- Definir los casos de uso y actores del negocio.

- Realizar el levantamiento de requisitos y modelación del sistema.
- Realizar el análisis y diseño de los servicios y herramientas a desarrollar.
- Implementar y probar.

Como solución al problema planteado se refiere la siguiente hipótesis: *con el desarrollo de servicios de comunicación en Software Libre, se logrará el envío de solicitudes de identificación y autenticación desde cualquier sistema cliente hacia el AFIS Civil.*

Los *servicios de comunicación*, más el *envío de solicitudes de identificación y autenticación desde cualquier sistema cliente hacia el AFIS Civil.*, son las variables reflejadas en la hipótesis. Su operacionalización se puede ver en Anexos I.

Los métodos teóricos apoyados en el proceso de análisis y síntesis permiten estudiar las características del objeto de investigación (*los servicios de comunicación para la transmisión de los datos biométricos*) que no son observables directamente.

Básicamente el método teórico, Histórico-Lógico da la posibilidad del análisis histórico del proceso de gestión de información. Y siguiendo la lógica de deducción junto a la hipótesis, el método Hipotético-Deductivo, permite llegar a nuevos conocimientos y predicciones, hasta determinar cuál es la forma más conveniente a utilizar en el futuro *desarrollo de servicios de comunicación de sistemas clientes con el AFIS Civil de SAIME*, y así posteriormente someterlas a verificaciones empíricas.

Para acercarse aún más al fin de los objetivos y siguiendo de manera lógica y coherente la investigación, se utilizaron los métodos empíricos. Los cuales ayudan a extraer de los fenómenos analizados las informaciones que se necesitan sobre ellos a través de la *observación* y del uso de técnicas opináticas como la *entrevista*.

La percepción directa experimentada en el ambiente del negocio de la situación problemática que se plantea, sin intermediarios que deformen los hechos de la realidad objetiva, es lo que brinda el método de la observación, logrando así un registro visual de lo que ocurre en una situación real para analizar cada fase del proceso e ir observando cada tarea que se realice y obtener experiencia de ésta para aplicarla en todas las demás.

Obtener información de un grupo de personas o de una en específico, es el objetivo que se persigue con la entrevista. Dotarse de un conocimiento cualitativo de los fenómenos y de las

características personales del entrevistado, tocando claramente el tema que importa, reflejado en una guía de preguntas para su desarrollo, y cuyas respuestas repercutirán en la investigación.

Apoyándose en la teoría de las probabilidades y su estrecha relación con la teoría de la selección, se escoge una medida que sirve de forma estadística, es decir permite expresar en números la efectividad de lo que se realiza. La *población* sería entonces esa medida general de la que se habla y que en la investigación es expresada como: *Entidades o sistemas externos a SAIME* y se tomará como *muestra* de esta población: *Los Procesos de verificación y transmisión de datos biométricos hacia el AFIS Civil*.

El presente trabajo consta de cuatro capítulos:

Capítulo 1 Fundamentación Teórica: Este capítulo muestra el estudio de soluciones que implementan un grupo de componentes y herramientas para lograr la identificación biométrica, además se hace referencia a las tendencias y tecnologías actuales que se usarán.

Capítulo 2 Características del Sistema: Se describen las actividades del flujo de procesos actuales a través de un modelo de dominio, el cual sirve de base para determinar qué es lo que se va a desarrollar. Se muestran los principales procesos a través de casos de uso, los actores que intervienen, sus relaciones y una descripción de cada una de ellos.

Capítulo 3 Análisis y Diseño del Sistema: Se muestran los diagramas de clases del análisis y del diseño para los casos de usos principales del sistema, junto con los correspondientes diagramas de interacción, además del diseño de la base de datos.

Capítulo 4 Implementación y Prueba: Se muestra el sistema desde la perspectiva de la implementación apoyados en los diagramas de despliegue y de componentes. Se realizan pruebas a los servicios implementados.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

A este capítulo concierne el estudio de soluciones que ya existen y tienen como objetivo principal utilizar las *huellas digitales* para la *verificación* de identidades o la *identificación* biométrica, apoyándose en la administración y el reconocimiento de dichas huellas. Del sistema automatizado de identificación de huellas dactilares (AFIS), se exponen las ideas fundamentales de funcionamiento en la actualidad.

Un acercamiento inicial a temas que están estrechamente vinculados a este estudio los da el presente apartado, con el fin de comprender, esclarecer ideas, conocimientos y conceptos. La arquitectura, por ejemplo, de un sistema de identificación de huellas, que tiene en cuenta básicamente la adquisición, comprensión de las huellas y una interfaz con la que interactúa el cliente, es uno de estos temas. Además de los estándares, formatos y protocolos de Internet que se utilizan con el fin de transportar datos a través de correos electrónicos. MIME, WSQ, SMTP son algunos.

Se brinda además una breve descripción del subsistema que permite a las aplicaciones de SAIME realizar la captura y gestión de los datos biométricos de los ciudadanos, en el momento que se disponen a realizar sus trámites.

Y finalmente se refiere el aprovechamiento de las ventajas que proporcionan metodologías de desarrollo de software como RUP y lenguajes de programación como Python, para desarrollar el análisis e implementación respectivamente. Además de la posibilidad que ofrece Python y el entorno integrado Eclipse de trabajar en Linux ya que son multiplataforma y así obtener el resultado esperado en Software Libre.

Con la investigación de los puntos antes mencionados y otros, se logrará alcanzar conocimientos necesarios para complementar la investigación e iniciar el desarrollo del nuevo software.

1.2 Biometría

La biometría proviene de las palabras bios (vida) y metron (medida). Biometría se refiere a dos campos diferentes. Uno que la define como una rama de las matemáticas estadísticas que se ocupa del análisis de datos biológicos y que comprende temas como población, medidas físicas, tratamientos de enfermedades, etc, y el otro, concerniente al estudio de métodos automáticos para el

reconocimiento único de humanos basados en uno o más rasgos conductuales. Esta última acepción es la usada en este trabajo.

Los sistemas biométricos incluyen un dispositivo de captación y un software biométrico que interpreta la muestra física y la transforma en una secuencia numérica. En el caso del reconocimiento de la huella digital, se ha de tener en cuenta que en ningún caso se extrae la imagen de la huella, sino una secuencia de números que la representan. Sus aplicaciones abarcan un gran número de sectores; desde el acceso seguro a computadores, redes, protección de ficheros electrónicos, hasta el control horario y control de acceso físico a una sala de acceso restringido. (HOMINI S.A. , 2004)

1.2.1 Identificación por huellas digitales

Todos los seres humanos tienen características morfológicas únicas que los diferencian. La forma de la cara, la geometría de partes del cuerpo como las manos, los ojos y tal vez la más conocida, la huella digital, son algunos rasgos que los diferencian del resto de los seres humanos.

La medición biométrica se ha venido estudiando desde tiempo atrás y es considerada en la actualidad como el método ideal de identificación humana.

La identificación por medio de huellas digitales constituye una de las formas más representativa de la utilización de la biometría. Una huella digital está formada por una serie de surcos. Las terminaciones o bifurcaciones de los mismos son llamados “puntos de minucia”. Cada uno de estos puntos tiene una característica y una posición única, que puede ser medida. Comparando esta distribución es posible obtener la identidad de una persona que intenta acceder a un sistema en general. (HOMINI S.A. , 2004)

1.2.2 Componentes biométricos

Un sistema biométrico común comprende varios componentes: un sensor utilizado para recopilar datos y convertir la información en formato digital. Algoritmos de procesamiento de señal que realizan actividades de control de calidad y generan las plantillas biométricas. Un componente para almacenamiento de datos que contiene la información con la cual se comparan las nuevas plantillas biométricas. Un algoritmo de coincidencia que compara las nuevas plantillas biométricas con una o más de las plantillas almacenadas. Y por último, un proceso de decisión (ya sea automático o manual) que utiliza los resultados del componente de coincidencia para tomar una decisión basada en el sistema. (Argentina, 2007)

El subsistema biométrico de SAIME posee también varios componentes, los cuales trabajan en conjunto para lograr que todos los datos de la persona sean capturados correctamente. Estos componentes se clasifican en, componentes de:

- Captura: El AgenteXLS2 y ClienteAFIS, son los encargados de que las huellas de la persona sean capturados correctamente.
- Compresión: El AgenteWSQ es el que realiza la compresión de las huellas, convirtiéndolas al formato WSQ que es el formato utilizado en el AFIS y otros sistemas biométricos para optimizar el espacio de almacenamiento.
- Comunicación: AgenteCorreo, AgenteEnvío y AgenteControlFlujo, son algunos de los principales componentes, pues son los intermediarios de comunicación con los servicios AFIS de los cuales se hablará más adelante.
- Configuración: AgenteCambioCfg, AgenteCambioCfgC, son los intermediarios para el chequeo de los cambios de configuración de los servicios AFIS. AgenteConfCDB y AgenteConfSDB, son los encargados de portar los datos de configuración que son persistentes en la base de datos del cliente y el servidor respectivamente.

Todos estos componentes sirven de base de estudio para diseñar componentes y servicios similares en software libre.

1.2.3 Dispositivos biométricos

Los dispositivos biométricos tienen la función de medir o de extraer las características personales únicas para identificar a una persona. Esta tecnología se ha utilizado por décadas, específicamente en seguridad y control de acceso. Existen dos categorías de dispositivos biométricos: fisiológicos y de comportamiento. Los de comportamiento identifican características únicas, como la firma de una persona. Los de identificación fisiológica se basan en la medición de características únicas del cuerpo tales como: los detalles de las huellas digitales, patrones de las venas, la retina, características del iris o el tamaño y forma de la mano. Estos dispositivos comparan todas estas características contra un patrón que ha sido grabado durante un proceso de inscripción. (NEO TEC S.A, 2007)

Los dispositivos que utilizan las aplicaciones SAIME para la identificación de un individuo son: bloque óptico para captar las huellas dactilares, una cámara digital Canon para la foto de identidad y una tabla (tablet) de firma para tomar la firma de la persona.

1.2.4 Estándares asociados a tecnologías biométricas

Los estándares más importantes son: Estándar ANSI X.9.84 creado en 2001 por la ANSI (American National Standards Institute) y actualizado en 2003, define las condiciones de los sistemas biométricos para la industria de servicios financieros haciendo referencia a la transmisión y almacenamiento seguro de información biométrica, y a la seguridad del hardware asociado.

Estándar ANSI / INCITS 358 creado en 2002 por ANSI y BioApi Consortium, que presenta una interfaz de programación de aplicación que garantiza que los productos y sistemas que cumplen este estándar sean inter-operables entre sí.

Estándar NISTIR 6529 conocido como CBEFF (Common Biometric Exchange File Format) es un estándar creado en 1999 por NIST y Biometrics Consortium que propone un formato estandarizado (estructura lógica de archivos de datos) para el intercambio de información biométrica.

Por otra parte la Norma Biométrica ISO ISO/IEC JTC1/SC37 "Biometrics", determina que su campo de actividad sean las tecnologías biométricas genéricas, correspondientes a seres humanos, para aportar interoperabilidad e intercambio de datos entre aplicaciones y sistemas. Las normas sobre identificación biométrica humana genérica incluye: entornos de ficheros comunes; interfaces de programación de aplicaciones biométricas; formato de intercambio de datos biométricos; perfiles relacionados con la biometría; aplicación de criterios de evaluación para las tecnologías biométricas; metodologías para la verificación y emisión de informes sobre el rendimiento de los sistemas, y los aspectos sociales y jurisdiccionales. (Argentina, 2007)

Otro de los estándares es WSQ (Wavelet Scalar Quantization). Es un estándar definido por el FBI (Federal Bureau of Investigation) y desarrollado por el NIST, en los Estados Unidos de Norteamérica, la especificación WSQ se ha convertido en el estándar de la industria para almacenar y transportar impresiones dactilares por medios electrónicos.

La especificación WSQ fue diseñada específicamente para lograr altos grados de compresión de las imágenes de huellas dactilares, con una calidad superior.

El principal beneficio de este formato es el alto grado de compresión que se puede alcanzar. Por ejemplo, una imagen de 800 x 750 píxeles, en escala de grises de 8 bits, ocupa en disco aproximadamente 585 kilo bytes. Con compresión JPEG el tamaño de la imagen ocuparía 88 kilobytes. En cambio con WSQ, el tamaño en disco de la imagen sería de 33 kilobytes. En otras palabras, con JPEG se alcanza una compresión de 1 a 6, mientras que con WSQ la relación es usualmente 1 a 15.

Además la calidad de la imagen recuperada es más alta en formato WSQ, lo que es vital para aplicaciones de identificación y comparación de huellas, en aplicaciones médicas, o cualquier uso donde es importante no perder los detalles de la imagen. (Technologies, 2005)

1.2.5 Procesos de Autenticación e Identificación biométrica

En los procesos de autenticación o verificación los rasgos biométricos se comparan solamente con los de un patrón ya guardados, este proceso se conoce también como comparación uno a uno (1:1). También implica conocer presuntamente la identidad del individuo a autenticar, es decir, dicho individuo ha presentado algún tipo de credencial, que después del proceso de autenticación biométrica será válido o no.

En el proceso de identificación los rasgos biométricos se comparan con los de un conjunto de patrones ya guardados, este proceso se conoce también como comparación uno a muchos (1: N). El mismo implica no conocer la identidad presunta del individuo, la nueva muestra de datos biométricos es tomada del usuario y comparada una a una con los patrones ya existentes en el banco de datos registrados. El resultado de este proceso es la identidad del individuo, mientras que en el proceso de autenticación es un valor verdadero o falso.

El proceso de autenticación o verificación biométrica es más rápido que el de identificación biométrica, sobre todo cuando el número de usuarios (N) es elevado. Esto es debido a que la necesidad de procesamiento y comparaciones es más reducida en el proceso de autenticación. Por esta razón, es habitual usar autenticación cuando se quiere validar la identidad de un individuo desde un sistema con capacidad de procesamiento limitada o se quiere un proceso muy rápido. (Wikipedia, 2007)

1.3 Soluciones existentes sobre identificación biométrica

Teniendo en cuenta que el presente trabajo tiene como condición la utilización de tecnologías libres, se realizó un estudio sobre soluciones existentes de este tipo. Y además se hace referencia a dos sistemas o plataformas propietarias como base de estudio para estudiar las funcionalidades más relacionadas con este subsistema de comunicación biométrica. La plataforma biométrica HOMINI y MegaMatcher SDK.

1.3.1 Plataforma Biométrica HOMINI

Es una plataforma propietaria con la cual se pueden administrar las huellas digitales que son utilizadas en procesos de verificación de identidad, autorizadores de transacciones o reemplazo de la contraseña por la huella digital. Posee componentes que pueden ser integrados a aplicaciones existentes y así lograr integrar la biometría a los programas que en la actualidad utiliza el cliente evitando el desarrollo de nuevas aplicaciones.

La plataforma HOMINI es una excelente opción para controlar procesos a través de huellas digitales por su:

- Flexibilidad
- Escalabilidad
- Seguridad
- Robustez
- Parametrización
- Fácil integración
- Trabaja con componentes que interactúan con aplicaciones existentes.

La plataforma también está compuesta por varios componentes:

- *Consola de administración:* Lugar donde se parametriza y administra la seguridad del sistema a través de perfiles preestablecidos; adicionalmente sirve para monitorear y auditar cualquier transacción hecha con huella digital.
- *Servidor de Sockets:* Componente adecuado para la comunicación TCP/IP cuando existe una zona militarizada o firewall.
- *Servidor de Verificación:* Este componente permite la verificación de identidad en forma centralizada y adicionalmente realizar actualización allí y no en todos los clientes.
- *Componente de enrolamiento:* Este componente permite ingresar o enrolar a todos los usuarios que van a interactuar con los procesos donde se utilizará la biometría como medio autorizador.
- *Componente de Verificación:* Este es el componente que permite realizar verificaciones haciendo el llamado al hardware biométrico, comprimiendo, encriptando y transmitiendo los datos al aplicativo del cliente para luego ser corroborados en el servidor de verificación. (HOMINI, 2004)

1.3.2 MegaMatcher SDK

El SDK MegaMatcher se basa en la tecnología de MegaMatcher que es biométrica múltiple y se puede escalar, que garantiza una gran confiabilidad y una gran velocidad en la identificación biométrica incluso al usar grandes bases de datos. Se logran una gran productividad y eficiencia por medio de un algoritmo combinado que contiene motores de reconocimiento de huellas digitales y rostros. Los integradores pueden usar el algoritmo combinado para lograr mejores resultados de identificación o usar cualquiera de los algoritmos por separado. El software de clusters (grupos) que tolera fallas y se puede escalar permite comparaciones rápidas en paralelo, procesa una gran cantidad de solicitudes de identificación y maneja bases de datos, casi sin limitación en cuanto a tamaño.

Para garantizar la compatibilidad de sistemas con otro software, se incluye la librería WSQ, y los módulos para la conversión entre la plantilla MegaMatcher y los estándares biométricos (/NIST ITL-1-2000, ANSI/INCITS 378-2004 e ISO/IEC 19794-2).

El SDK MegaMatcher 2.0 es útil no solo para el desarrollo de AFIS para el sector privado, sino también para aplicaciones AFIS forenses, ya que incluye un componente de clasificación de huellas digitales y un API (Interfaz de Programación de Aplicaciones) para la edición de plantillas de huellas digitales latentes. Se requiere este tipo de edición para ingresar huellas digitales latentes para la identificación de AFIS (por ejemplo, si se obtiene la huella digital de un sitio de crimen). MegaMatcher también puede cotejar huellas digitales enrolladas y planas. (S.A, 2007)

Las dos soluciones anteriormente analizadas a pesar de poseer las características mencionadas requieren de licencia, lo que las hace software propietario, además no entra en el plano de los servicios de comunicación de un sistema biométrico distribuido, como lo es el subsistema que se quiere realizar.

1.3.3 Biometric SDK

Biometric SDK es un kit en software libre de desarrollo de aplicaciones biométricas. Aunque muy bien organizado, todavía el software está en su etapa alfa y no tiene el nivel de madurez requerido para ser aplicada. Adolece de un servidor biométrico al que integrarse, de comunicaciones a distancia. Solo presenta la parte de extracción y matcheo de plantillas biométricas. Desarrolladas versiones en Java y C#, se espera desarrollarla también en C++. Su propósito es hacer un software de comparación biométrico por huellas digitales competitivo comercialmente.

1.4 EI AFIS

1.4.1 Sistema Automatizado de Identificación de Huellas Dactilares (AFIS)

Es un sistema computarizado que permite la identificación rápida y confiable de personas. Está compuesto de Hardware y Software integrados que permite la captura, consulta y comparación automática de huellas dactilares agrupadas por fichas decadactilares, monodactilares o en forma de rastro, basados en las ciencias biométricas, la matemática, los cálculos de transformadas (Fourier) la coherencia y la correlación. Actualmente las principales policías del mundo están equipadas de sistemas AFIS para la investigación criminal. Además de que el uso de la huella dactilar para controlar la entrega de tarjetas de identidad y pasaportes es un requisito cada día más frecuente por parte de las administraciones con el fin de evitar el fraude. (S.A, 2002)

1.4.2 La evolución del AFIS

Estos sistemas se emplean desde hace varios años, pero al principio eran sólo archivos informatizados que funcionaban con ayuda manual (los datos tenían que buscarse personalmente). Con el tiempo, se han perfeccionado y se han convertido en sistemas integrados, que utilizan tecnología digital: la huella se puede escanear para su búsqueda y cotejo en el sistema o se introduce directamente a través de un "live-scan". (Reisz)

1.4.3 AFIS civil y criminal

Existen dos clase de AFIS, el civil y el criminal. El criminal tiene como objetivo la lucha contra el crimen. Se utiliza para buscar rastros (una huella "latente" encontrada en la escena de un crimen), contra una base de datos AFIS, con el objetivo de identificar a la persona poseedora de dicha huella o comprobar que el dueño de la latente no se encontraba en otra escena de un crimen donde dejó sus huellas.

Un sistema civil se utiliza por ejemplo, para garantizar que una persona no logre, mediante la presentación de documentos falsos, poseer doble o múltiple identidad. (Reisz)

El AFIS Criminal tiene huellas roladas, y generalmente solo de la población penal. El AFIS Civil por su parte tiene generalmente huellas planas de toda la población. Las huellas roladas pueden compararse contra huellas latentes pues guardan mayor superficie que las planas pero requieren del

AFIS un mayor procesamiento, por lo que el AFIS Civil usa huellas planas. Frecuentemente un AFIS Criminal puede preguntarle a un AFIS Civil en caso de no encontrar la identidad buscada.

1.4.4 Arquitectura de los sistemas AFIS

La arquitectura de los sistemas automáticos de reconocimiento de patrones de huella dactilar consta, básicamente, de tres partes:

- Etapa de adquisición de datos: captura mediante escáner la imagen de la huella dactilar.
- Etapa de extracción de características: a partir de la imagen adquirida se extraen automáticamente las características de la huella que conforman el patrón biométrico.
- Etapa de comparación de patrones: en esta etapa, el patrón biométrico de la huella que accede al sistema es comparado con el patrón registrado en la base de datos. El grado de similitud resultante de la comparación de los dos patrones es cuantificado dentro de un rango de valores. (Reisz)

El AFIS proporciona estas dos últimas partes, a diferencia de la primera que lo hace un sistema externo. En el presente trabajo se presenta la comunicación de estos dos sistemas.

1.5 Servicios de comunicación con el AFIS Civil de SAIME

En la República Bolivariana de Venezuela, dada su extensión territorial, la distribución geográfica de su población y la voluntad del ministerio de llevar los beneficios de SAIME a todos los ciudadanos, este se encuentra distribuido en las Oficinas Regionales (OR) que se encuentran en todos los estados de la nación. Generalmente estas oficinas están muy distantes del AFIS y frecuentemente con recursos de conexión limitados, por lo que se ha dado la necesidad de comunicarlos, generando un conjunto de componentes que gestionan la adquisición y transmisión de datos biométricos de un ciudadano desde cualquier oficina hasta los sistemas centrales.

Esta comunicación la realizan los servicios (local y central) a partir de correos electrónicos que portan solicitudes de identificación, autenticación o recuperación de datos, cuyos datos viajan como un fichero adjunto de formato ANSI/NIST. El AFIS Civil elabora respuestas de la misma forma. (Cervantes, 2007)

Estos servicios sirven como base para diseñar e implementar los nuevos servicios que permitirán la comunicación de sistemas clientes con el AFIS Civil. En los siguientes puntos se hace una breve explicación de lo que hace cada uno.

1.5.1 Servicio Local AFIS (SLA)

La comunicación de la Oficina Regional con el sistema AFIS Civil, la realiza el “Servicio Local AFIS” que corre en el servidor de la oficina, cuyo objetivo es constituir una interfaz de comunicación asincrónica, por medio de correos electrónicos, entre las aplicaciones que usan la BD Local de la Oficina Regional y los componentes biométricos remotos que participan en el proceso de verificación biométrica.

O sea, el SLA revisa constantemente aquellas transacciones que están listas para enviar y en caso de haber alguna las envía por correo hacia su destino, que suele ser directamente las cuentas de correo internas del AFIS Civil. A medida que este las va procesando envía las respuestas a un buzón que tiene cada oficina y que a su vez está siendo revisado constantemente por el SLA para incluirlas en la BD. Ver figura 1.1. (Cervantes, 2007)

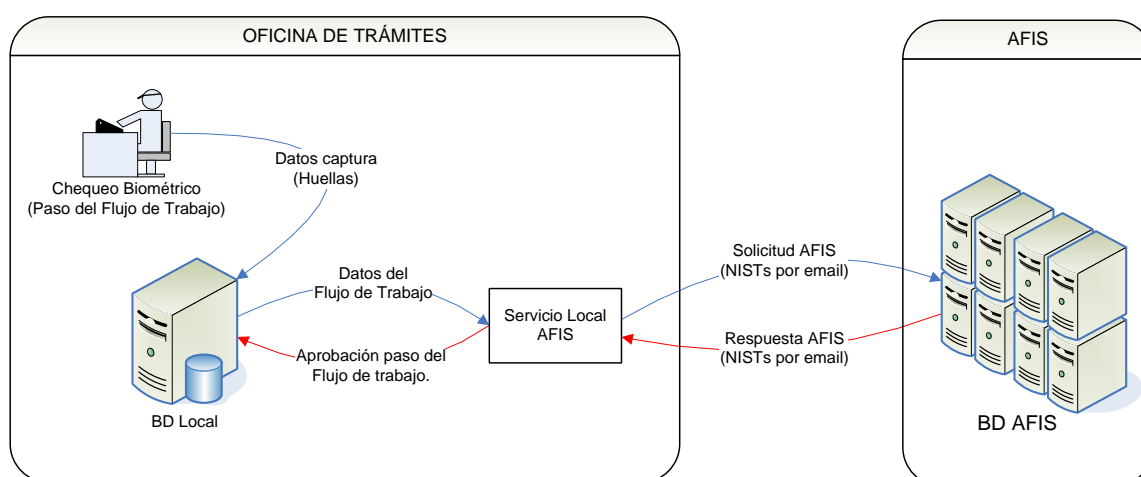


Figura 1.1: Interacción simplificada de una Oficina Regional con el AFIS Civil

1.5.2 Servicio Central AFIS (SCA)

El proceso de personalización de documentos de identidad se realiza a nivel central, obteniendo los datos del individuo desde el centro de datos de SAIME, incluyendo sus huellas. Por lo que se hace necesario entonces un mecanismo que permita transportar las huellas desde las oficinas hasta el centro de datos y con el segundo objetivo de tener una especie de registro de todas las transacciones realizadas en las distintas oficinas.

Aquí es donde participa el Servicio Central AFIS (SCA) en el cual debe persistir en la BD Central una copia de los datos de las transacciones enviadas desde las aplicaciones SAIME de las OR hacia el

AFIS Civil, y de las respuestas correspondientes. Dentro de estos datos deben estar las huellas correspondientes a las solicitudes. (Cervantes, 2007)

Ahora la comunicación entre el Centro de Datos de SAIME, La Oficina de Regional de SAIME y el AFIS Civil quedaría como se especifica en la figura 1.2.

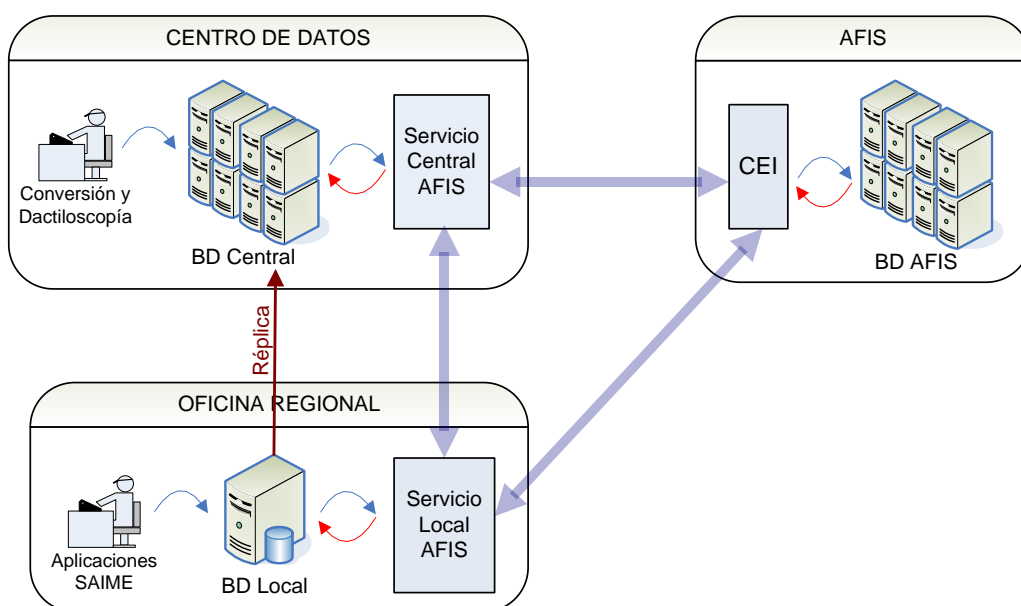


Figura 1.2 Forma general del proceso de verificación biométrica

Debe notarse que en este modelo de comunicación el AFIS Civil usado en las aplicaciones de SAIME se duplican las transacciones enviadas al AFIS, lo que aumenta la complejidad del estado de las verificaciones. Esto sucede por utilizar los mecanismos de transporte de correos para subir las huellas al centro de datos, y así aliviar de esta carga al servicio de réplica. Este modelo requiere una perfecta sincronización de esta comunicación tripartita, y si falla alguna de las tres comunicaciones el estado de la verificación del trámite es un poco más difícil de determinar y resolver. En la solución propuesta, no existen estos requerimientos especiales por lo que este problema se evita enviando todas las solicitudes al servicio de distribución central sin copias y solo desde allí se envían al AFIS, a su vez las respuestas del AFIS se concentran en un solo lugar para ser distribuidas a su destino final. Esto hace posible que los datos de una transacción estén en un solo lugar a la vez y que se tenga mayor control sobre el estado de la verificación.

El hecho de que las transacciones se puedan enviar desde entidades remotas directamente al AFIS, hace también difícil un mecanismo de control de flujo unificado. Es decir, a las entidades remotas se le puede pedir que envíen solicitudes a un ritmo, pero no hacer una programación coordinada de ese

envío al AFIS. En la solución propuesta se concentran las solicitudes y solo desde ahí se envían al AFIS.

Además el hecho de concentrar las transacciones ofrecería la posibilidad de cuantificar de manera activa las transacciones a las entidades remotas, permitiendo establecer cuotas y permisos. Por ejemplo la entidad "A" puede recibir solo transacciones AUT y una cantidad de 2000 diarias por enviar todas las transacciones a nivel central y comunicarse con el AFIS solo desde un punto controlado (el SDA) y no desde todas las oficinas directamente.

1.5.3 Arquitectura interna e Interacción entre los servicios de comunicación

Al realizar un envío, el Servicio de Comunicación AFIS (sea Local o Central) procesa la información en varias fases para lograr conformar el correo electrónico con los datos de la transacción adjuntos y en el formato adecuado. Cada una de las fases utiliza una capa distinta de componentes que le da el formato adecuado al mensaje como se muestra en la figura 1.3.

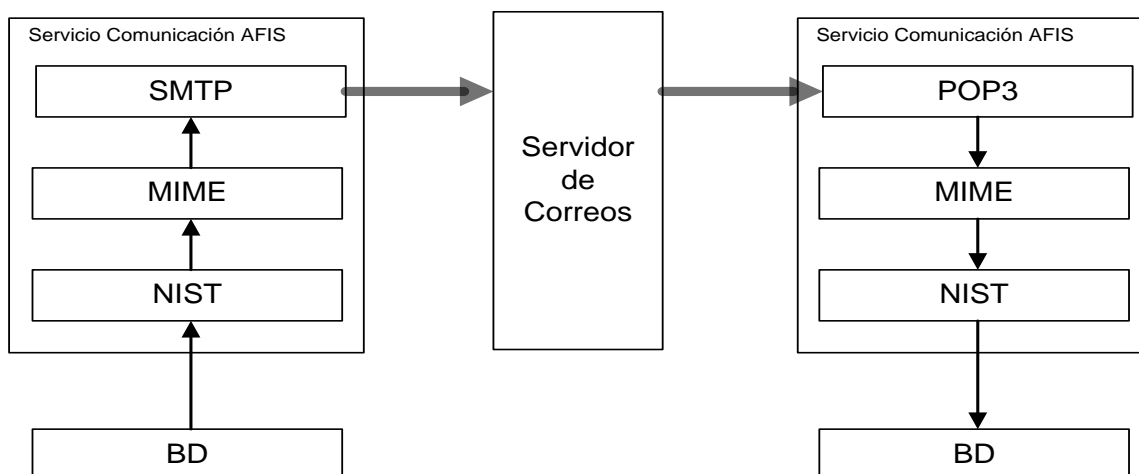


Figura 1.3: Modelo de interacción entre dos servicios de comunicación AFIS

El servicio de comunicación AFIS está compuesto por un conjunto de componentes que le permiten realizar estas tareas:

1. Librería de Acceso a la BD asociada al servicio que permite obtener los datos de las transacciones a transmitir o escribir los datos de las transacciones recibidas.
2. Librería NIST que permite crear este tipo de ficheros con los datos biométricos de la persona para ser enviados al AFIS.
3. Librería MIME que permite elaborar mensajes de correo electrónico que incluyan adjuntos.

4. Librerías de Correos que permite establecer sesiones con servidores SMTP para enviar los mensajes de correo y/o con el servidor POP3 para recibirlos.

Para transmitir cada transacción, el servicio debe realizar las siguientes operaciones, en este orden:

1. Obtener los datos desde la BD asociada al servicio.
2. Elaborar el paquete NIST con esos datos.
3. Codificar un mensaje electrónico MIME con el fichero NIST adjunto.
4. Establecer una sesión SMTP con el servidor de correos y enviarle los comandos necesarios para transferirle el mensaje.

Para recibir esa transacción, el servicio de destino debe:

1. Establecer una sesión POP3 con el servidor de correos y enviarle los comandos necesarios para recibir el mensaje.
2. Decodificar el mensaje electrónico MIME y extraer el contenido equivalente al fichero adjunto.
3. Interpretar ese fichero obtenido como un formato NIST y extraerle los datos de la solicitud.
4. Escribir en la BD de destino los datos correspondientes. (Cervantes, 2007)

1.6 Servicios Web

Conocer qué son y para qué sirven los servicios web es un punto de suma importancia en la presente investigación. Dos de los objetivos específicos descritos anteriormente reflejan la necesidad de implementar dos servicios web, uno que se conectará con las aplicaciones cliente, y el otro para el beneficio del administrador.

1.6.1 ¿Qué son los Servicios Web?

Existen múltiples definiciones sobre lo que son los Servicios Web, lo que muestra su complejidad a la hora de dar una adecuada definición que englobe todo lo que son e implican. Una posible sería hablar de ellos como distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, soportando así la interoperabilidad máquina - máquina (interoperabilidad sintáctica) y el intercambio de datos entre redes de ordenadores. Esta interoperabilidad se consigue mediante la adopción de estándares abiertos. (Webmaster, 2008)

Otra manera de definirlos sería como; una colección de protocolos y estándares que sirve para comunicar aplicaciones e intercambiar datos entre ellas. Es un componente de software que se comunica con otras aplicaciones codificando los mensajes en XML y enviando estos mensajes a través de protocolos estándares de Internet tales como el Hypertext Transfer Protocol (HTTP). Intuitivamente, es similar a un sitio web pero no cuenta con una interfaz de usuario ya que brinda servicios a aplicaciones y no a las personas, pues en vez de obtener solicitudes desde el navegador y retornar páginas web como respuesta, lo que hace es recibir solicitudes a través de un mensaje SOAP (formato XML diseñado para esta comunicación) desde una aplicación, realiza una tarea y devuelve un mensaje de respuesta también formateado en XML. (W3C, 2008)

1.6.2 ¿Para qué sirven?

Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario, es decir, exponen funcionalidad útil a los usuarios web mediante un protocolo web estándar. En la mayoría de los casos, el protocolo utilizado es SOAP.

Para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferente y ejecutada sobre cualquier plataforma pueden utilizar estos servicios para intercambiar datos en redes de ordenadores como Internet. (W3C, 2008)

1.6.3 Estándares que realizan el intercambio de mensajes en los Servicios Web

1.6.3.1 XML (Extensible Markup Language)

Es un subconjunto simplificado del SGML el cual fue diseñado principalmente para documentos web. Permite a los diseñadores crear sus propias "etiquetas" o "tags" (Ejemplo: <libro>), habilitando la definición, transmisión, validación, y la interpretación de datos entre aplicaciones y entre organizaciones. El HTML y el XML tienen funciones diferentes. El HTML tiene por objeto mostrar información, mientras que el XML se ocupa de la información propiamente dicha (el contenido). Constituye el lenguaje base de los servicios web; aunque por la utilidad que proporcionan pueden ser utilizados con otros fines y por otros servicios, por ejemplo los servicios o demonios, de los cuales se abordará más adelante. (Alvarez)

1.6.3.2 SOAP (Simple Object Access Protocol)

SOAP es el protocolo diseñado para acceder remotamente a los servicios web, pero se puede utilizar para realizar invocaciones remotas de diferentes tipos. Sus características son las siguientes:

- No está restringido a un lenguaje ni a una plataforma. SOAP no especifica una API, lo que permite al programador abstraer el lenguaje sobre el que pueda estar implementado el objeto invocado. Esto facilita la reusabilidad de código, y las actualizaciones de la aplicación.
- No está restringido a un protocolo de transporte. La especificación SOAP describe su transporte en HTTP pero puede ser transmitido en cualquier protocolo de transporte de texto.
- No está restringido a un Middleware. Está basado en estándares ya existentes. De hecho, incluso el sistema de tipos que utiliza es el de XML.
- Hace posible la comunicación entre entornos muy heterogéneos, ya que cualquier aplicación capaz de escribir y leer XML sobre HTTP puede utilizar SOAP. (Benedí)

1.6.3.3 WSDL (Web Services Description Language)

WSDL describe la interfaz pública a los servicios web. Es un protocolo basado en XML y describe los accesos al servicio web, la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.

Así, WSDL se usa a menudo en combinación con SOAP Y XML Schema. Un programa cliente que se conecta a un servicio web puede leer el WSDL para determinar qué funciones están disponibles en el servidor. Los tipos de datos especiales se incluyen en el archivo WSDL en forma de XML Schema. El cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL.

1.6.3.4 NuSOAP

Es una reescritura de SOAPx4, provista por NuSphere y Dietrich Ayala. Es un conjunto de clases (no se requiere ninguna extensión de PHP) que permiten a los desarrolladores crear y consumir servicios Web basados en SOAP 1.1, WSDL 1.1 y HTTP 1.0/1.1.

NuSOAP es un kit de herramientas (ToolKit) para desarrollar servicios web bajo el lenguaje PHP. Está compuesto por una serie de clases que hacen mucho más fácil el desarrollo de servicios web. Provee soporte para el desarrollo de clientes (aquellos que consumen los web services) y de servidores (aquellos que los proveen). NuSOAP está basado en SOAP 1.1, WSDL 1.1 y HTTP 1.0/1.1. (Webmaster)

1.7 Servicios o demonios

Un demonio o daemon (de sus siglas en inglés Disk And Execution Monitor), es un tipo especial de proceso informático que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario (es un proceso no interactivo). Este tipo de programas se ejecutan de forma continua (infinita), vale decir, que aunque se intente cerrar o matar el proceso, este continuará en ejecución o se reiniciará automáticamente. Todo esto sin intervención de terceros y sin dependencia de consola alguna.

Los programas demonios reciben este nombre en los sistemas UNIX. En otros sistemas existen procesos similares como los programas residentes de MS-DOS o los servicios de Windows.

1.7.1 Características

Los demonios suelen tener las siguientes características:

- No disponen de una interfaz directa con el usuario, ya sea gráfica o textual.
- No hacen uso de las entradas y salidas estándar para comunicar errores o registrar su funcionamiento, sino que usan archivos del sistema en zonas especiales (`/var/log/` en los UNIX más modernos) o utilizan otros demonios especializados en dicho registro como el `syslogd`. (Wikipedia, 2008)

Para el cumplimiento del segundo y tercer objetivo específico que ya se planteó es necesario desarrollar dos servicios que tengan estas características. Estos son los encargados de comunicarse con el AFIS civil a través de correos electrónicos.

1.8 Estándares y protocolos de Internet

En el epígrafe 1.5 se mencionan varios protocolos de Internet que están presentes en la arquitectura interna e interacción entre los servicios de comunicación. A continuación se hará una breve descripción de algunos de ellos.

1.8.1 MIME (Multipurpose Internet Mail Extensions)

MIME, en español Extensiones de Correo Internet Multipropósito. Son una serie de convenciones o especificaciones dirigidas a que se puedan intercambiar a través de Internet todo tipo de archivos (texto, audio, vídeo, etc.) de forma transparente para el usuario. Una parte importante del MIME está dedicada a mejorar las posibilidades de transferencia de texto en distintos idiomas y alfabetos.

En sentido general las extensiones de MIME van encaminadas a soportar:

- Texto en conjuntos de caracteres distintos de US-ASCII.
- Adjuntos que no son de tipo texto.
- Cuerpos de mensajes con múltiples partes (multi-part).
- Información de encabezados con conjuntos de caracteres distintos de ASCII.

MIME además define mecanismos para enviar otros tipos de información en un correo (e-mail), incluyendo texto en lenguajes distintos del inglés utilizando otros conjuntos de caracteres además de ASCII, así como contenido binario representado con 8 bits como archivos que contienen imágenes, sonidos, películas, y programas de computadora. MIME es adicionalmente un componente fundamental en protocolos de comunicación como HTTP, el que requiere que los datos sean transmitidos como un e-mail aunque los datos pueden no ser un e-mail propiamente dicho. Los clientes de correo y los servidores de correo convierten automáticamente desde y a formato MIME cuando envían o reciben (SMTP/MIME) e-mails.

En la actualidad ningún programa de correo electrónico o navegador de Internet puede considerarse completo si no acepta MIME en sus diferentes facetas (texto y formatos de archivo). (Wikipedia, 2007)

1.8.2 IMAP (Internet Message Access Protocol)

IMAP es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor. Mediante él se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a Internet. IMAP tiene varias ventajas sobre POP, que es el otro protocolo empleado para

obtener correo desde un servidor. Por ejemplo, es posible especificar en IMAP carpetas del lado servidor. Por otro lado, es más complejo que POP.

IMAP es utilizado frecuentemente en redes grandes y les permite a los usuarios acceder a los nuevos mensajes instantáneamente en sus computadoras, ya que el correo está almacenado en la red. Con POP3 los usuarios tendrían que descargar el email a sus computadoras o accederlo vía web. Ambos métodos toman más tiempo de lo que le tomaría a IMAP, y se tiene que descargar el correo electrónico nuevo o refrescar la página para ver los nuevos mensajes.

Los clientes de correo electrónico que utilizan IMAP dejan por lo general los mensajes en el servidor hasta que el usuario los elimina explícitamente. Esto y otros factores hacen que la operación de IMAP permita a múltiples clientes acceder al mismo buzón de correo. La mayoría de los clientes de correo electrónico soportan POP3 ó IMAP; sin embargo, solo unos cuantos proveedores de Internet ofrecen IMAP como un valor agregado a sus servicios. (Wikipedia, 2007)

1.8.3 SMTP (Simple Mail Transfer Protocol)

Protocolo Simple de Transferencia de Correo que controla la forma en que el correo electrónico se transporta a través de Internet y se entrega en el servidor de destino. Se basa en texto y es utilizado para el intercambio de mensajes de correo electrónico entre computadoras o distintos dispositivos (PDA's, teléfonos móviles, etc.) y es un estándar oficial de Internet.

SMTP se basa en el modelo cliente-servidor, donde un cliente envía un mensaje a uno o varios receptores. La comunicación entre el cliente y el servidor consiste enteramente en líneas de texto compuestas por caracteres ASCII.

Las respuestas del servidor constan de un código numérico de tres dígitos, seguido de un texto explicativo. El número va dirigido a un procesado automático de la respuesta por autómata, mientras que el texto permite que un humano interprete la respuesta. (Wikipedia, 2007)

1.9 Tendencias y tecnologías actuales

Durante los cursos de la carrera se estudiaron algunos lenguajes de programación, de modelado, y metodologías de desarrollo que sirvieron en la confección de proyectos de cursos. Hoy se escogen para modelar los servicios el lenguaje de modelado UML, y junto a RUP se seguirá el camino de sus flujos de trabajo para el análisis y documentación del sistema.

Desarrollar la implementación en dos lenguajes de programación, PHP y Python. PHP para el desarrollo de los servicios web y Python para el desarrollo de los servicios o demonios. Un acercamiento a estas tendencias junto a otras no menos importantes sugieren los puntos siguientes.

1.9.1 RUP como metodología de desarrollo de software

El Proceso Unificado Racional (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

Se caracteriza por ser iterativa e incremental, centrada en la arquitectura y guiada por los casos de uso. Divide el proceso de desarrollo en ciclos de vida, obteniendo un producto al final de cada ciclo, los cuales se dividen en fases que deben de terminar con un hito, dentro de estas fases se encuentran: Inicio, Elaboración, Construcción y Transición. Unifica los mejores elementos de metodologías anteriores, está preparada para desarrollar grandes y complejos proyectos y utiliza el UML como lenguaje de representación visual para preparar los esquemas de un sistema software.

Las principales características de RUP son:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software. (Wikipedia, 2008)

1.9.2 UML como lenguaje de modelado de sistemas de software

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando no es un estándar oficial, está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como

procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional) -pero no especifica en sí mismo qué metodología o proceso usar. (Wikipedia, 2008)

1.9.3 Herramienta Visual Paradigm

Es una herramienta CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) que da soporte al modelado visual con UML.

Visual Paradigm ofrece:

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDEs.
- Disponibilidad en múltiples plataformas. (Caballero)

1.9.4 Lenguaje de programación para los Servicios Web: PHP

PHP es el acrónimo de Hypertext Preprocessor (Preprocesador de Hipertexto). Se trata de un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Está muy orientado al desarrollo de aplicaciones web y permite insertar contenidos dinámicos en las páginas.

Para la implementación del servicio web se emplea como lenguaje de programación PHP ya que es uno de los más extendidos en la red de redes y ha sido aceptado precisamente por la simplicidad y potencia que lo caracteriza, ofrece gran variedad de funciones para la explotación de bases de datos sin grandes complicaciones. Es un lenguaje multiplataforma completamente gratuito que puede ser

ejecutado en la mayoría de los sistemas operativos tales como UNIX, Windows y Mac OS X, y puede interactuar con los servidores de web más populares, pues existe en versión CGI, módulo para Apache, e ISAPI.

Principales Ventajas:

- Muy sencillo de aprender.
- Similar en sintaxis a C y a PERL.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- El análisis léxico para recoger las variables que se pasan en la dirección lo hace PHP de forma automática. Librándose el usuario de tener que separar las variables y sus valores.
- Se puede incrustar código PHP con etiquetas HTML.
- Excelente soporte de acceso a base de datos.
- La comprobación de que los parámetros son válidos se hace en el servidor y no en el cliente (como se hace con javascript) de forma que se puede evitar chequear que no se reciban solicitudes adulteradas. Además PHP viene equipado con un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos.
- Se puede hacer de todo lo que se pueda transmitir por vía HTTP.

1.9.5 Lenguaje de programación Python

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- La cantidad de plataformas en las que podemos desarrollar, como Unix, Windows, OS/2, Mac, Amiga y otros.
- Además, Python es gratuito, incluso para propósitos empresariales. (DesarrolloWeb, 2007)

1.9.6 Plataforma y entorno de desarrollo. Ubuntu y Eclipse

La plataforma de desarrollo es el entorno común en el cual se desenvuelve la programación de un grupo definido de aplicaciones. Comúnmente se encuentra relacionada directamente a un sistema operativo. El desarrollo de este trabajo será sobre la plataforma de Linux, específicamente con la versión Ubuntu 7.10.

Linux es la denominación de un sistema operativo tipo Unix (también conocido como GNU/Linux) y el nombre de un núcleo. Es uno de los ejemplos más prominentes del software libre y del desarrollo del código abierto, cuyo código fuente está disponible públicamente, para que cualquier persona pueda libremente usarlo, estudiarlo, redistribuirlo, comercializarlo y, con los conocimientos informáticos adecuados, modificarlo.

Eclipse es un entorno integrado de desarrollo libre, del inglés *Integrated Development Environment* (IDE), disponible en Linux. Eclipse tiene la posibilidad de añadir nuevas funcionalidades al editor, a través de nuevos módulos (*plugins*), para programar en otros lenguajes de programación además de Java como C/C++, PHP, Python, Ruby, Cobol, y más.

A continuación se enumeran sus principales características:

- Multiplataforma (GNU/Linux, Solaris, Mac OSX).
- Estructura de *plug-in* que hace sencillo añadir nuevas características y funcionalidades.
- Resaltado de sintaxis, autocompletado, tabulador de un bloque de código seleccionado, multitud de utilidades de edición que ayudan enormemente al programador. (Campus Libre y Abierto, 2006)

1.9.7 Servidor web Apache

Para realizar páginas web dinámicas con PHP en Ubuntu es necesario la utilización de PHP5 y Apache2, para ello se estudia el servidor apache, ya que PHP es un lenguaje de programación ejecutado en un servidor.

Apache está diseñado para ser un servidor web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios web elegir qué características van a ser incluidas en el servidor seleccionando, qué módulos se van a cargar, ya sea al compilar o al ejecutar el servidor.

¿Cómo funciona este servicio?

Un servidor es una computadora que entrega a otras computadoras (los clientes), una información que ellos requieren bajo un lenguaje común, denominado protocolo. Por lo tanto, al ver una página web es porque el servidor les entrega una página HTML vía protocolo HTTP (HyperText Transport Protocol) o protocolo para la transmisión de hipertexto, a través de una conexión TCP/IP por el puerto 80. Es por eso que en el Servidor Web es donde se almacena la información estática accedida y/o las aplicaciones que la generan.

1.9.8 Servidores de correos

¿Qué es un Servidor de Correo?

Linux utiliza los programas Sendmail, QMail y PostFix como servidor de correo. Estos programas son utilizados por muchos proveedores de servicio. Entre otros, utiliza los protocolos SMTP, POP3 e IMAP. El SendMail es el servidor de correos más usado a nivel mundial, pero no recomendado para un servidor con más de 500 cuentas. El Postfix es un servidor de correos mucho más estable seguro y de capacidad ilimitada de cuentas de correo, utilizado en las principales compañías proveedoras de servicios de Internet en el mundo.

Para ambos servidores existen módulos adicionales para soporte de Listas de Correo, Scan Antivirus Anti Spam, Interface Webmail y otros.

El MTA (Mail Transportation Agent) Postfix pretende ser rápido, fácil de administrar, seguro y a la vez suficientemente compatible con Sendmail como para que los usuarios existentes no se asusten. Por lo tanto, externamente mantiene el estilo de Sendmail, mientras que internamente es completamente diferente.

A diferencia de Sendmail, Postfix no es un programa monolítico, sino una combinación de pequeños programas, cada uno de los cuales lleva a cabo una función especializada. (Jaume Sabater, 2008)

Postfix es un MTA libre, y escrito por Vietse Venema, el autor de programas tan conocidos como TCP Wrappers, The Coroner Toolkit, o SATAN, todos ellos orientados a la seguridad. De ello podemos deducir que Postfix está desde un principio orientado a la seguridad, una de las principales ventajas frente a su predecesor Sendmail. Puede utilizarse perfectamente como reemplazo de Sendmail, ya que Postfix posee varias capas de compatibilidad con el mismo.

Algunas características de Postfix son las siguientes:

- Seguridad: Tanto en el diseño como en el soporte de SASL, TLS, etc
- Diseño modular

- Soporte de distintas bases de datos: LDAP, MySQL, PostgreSQL, etc
- Estricto cumplimiento de los estándares de correo electrónico
- Soporte para dominios y usuarios virtuales (que no están realmente en el sistema)
- Mucha documentación. (Webmaster , 2008)

1.9.9 PostgreSQL como gestor de Base de Datos

El PostgreSQL es un poderoso sistema manejador de bases de datos, es decir, un sistema diseñado para administrar grandes cantidades de datos, que tiene la fama de ser la base de datos de código abierto (Open Source) más avanzada del mundo. El PostgreSQL se ejecuta en la mayoría de los Sistemas Operativos más utilizados en el mundo incluyendo, Linux, varias versiones de UNIX y por supuesto Windows.

PostgreSQL se ha preocupado por ser una solución real a los complejos problemas del mundo empresarial y a la vez mantener la eficiencia al consultar los datos. Con ese fin, se han desarrollado y añadido al PostgreSQL las más interesantes y útiles características que antes sólo podían hallarse en sistemas manejadores de bases de datos comerciales como *Oracle*, *DB2* o *Sybase*; lo cual lo coloca, como su lema indica, como "el manejador de bases de datos de código abierto más avanzado del mundo". (http Perú, 2007)

1.10 Conclusiones

El análisis de los temas descritos en el presente capítulo conllevó a las siguientes reflexiones:

La biometría es considerada en la actualidad como el método ideal de identificación humana, pues utiliza aquellas características únicas del hombre, como por ejemplo la huella dactilar. Se describe el Reconocimiento de Huellas Dactilares como una de las técnicas biométricas más maduras y confiables.

Una mirada al funcionamiento del AFIS, permitió conocer más acerca de sus procesos. Toda la interacción entre las aplicaciones de SAIME y las del AFIS Civil se realiza a partir de correos electrónicos.

Se estudiaron algunos sistemas similares que utilizan la huella dactilar para la identificación y reconocimiento de las personas, lo que permitió conocer la importancia que estos tienen que se puede fundamentar ya que en los últimos años, la demanda de sistemas biométricos de gran escala se ha intensificado, muchos países ya están incluyendo información biométrica en pasaportes, tarjetas de identidad, visas y otros documentos.

Capítulo 1: Fundamentación Teórica

Y para el desarrollo del trabajo se usarán un conjunto de tecnologías actuales compatibles con Software Libre, que servirán de apoyo durante el análisis e implementación de los servicios.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En este capítulo se describen los procesos que forman parte del negocio del presente trabajo. El análisis del negocio permite conocer las actividades que conforman el flujo actual de los procesos involucrados y quiénes lo realizan. El modelado de estos serán el punto de partida para la comprensión de lo que se hará como solución del sistema.

Según la situación planteada y la información que se tiene al respecto, tratada en el capítulo anterior, se resuelve construir un modelo de dominio en el que se representan los conceptos más importantes y sus relaciones, conociéndose así los procesos del negocio, sus implicados y la conexión entre ellos.

Los conceptos del modelo de dominio se explican para una mejor comprensión del lector, en el glosario de términos especificado posteriormente.

La visión general que se obtiene con el análisis del negocio hasta aquí, es el punto de partida para adentrarse en las funcionalidades del futuro sistema, para ello se utilizan artefactos generados en el flujo de trabajo Requerimientos, de RUP. Levantamiento de requisitos, diagramas de casos de usos y descripción textual de dichos casos de usos.

2.2 Descripción de los procesos del negocio

2.2.1 Breve Descripción Del Negocio

El modelo de comunicación SAIME-AFIS representado en la figura 1.1 del capítulo anterior, está simplificado al análisis de las solicitudes y las respuestas de manera genérica. Hay varios tipos de solicitudes y respuestas que pueden combinarse para cumplimentar el chequeo biométrico de una persona en el sistema SAIME utilizando el AFIS.

Desde las Oficinas Regionales se le realiza la captura biométrica al ciudadano. Los pasos respectivos al subsistema de captura biométrica y comunicación con el AFIS son: Captura de Imágenes, Aprobación de Imágenes, Chequeo Dactiloscópico.

Una vez que el usuario se ha capturado sus huellas, se hace una representación en BD de la solicitud a enviar al AFIS, con todos sus datos. Además las solicitudes en BD tienen un atributo que indica el estado de esa solicitud concerniente a la comunicación con el sistema AFIS. La comunicación

de la Oficina Regional con el sistema AFIS Civil, la realiza un servicio que corre en el servidor de la oficina llamado “Servicio Local AFIS” (SLA), el cual se encarga de enviar por correo las transacciones listas para enviar al centro de datos o al AFIS según el tipo de solicitud.

2.2.2 Problemas fundamentales detectados

1. No hay control estadístico de la cantidad de solicitudes enviadas por cada cliente.
2. No hay asignación de cuotas de servicio por cliente.
3. El sistema de transición de estados de las transacciones para la comunicación es muy complejo por lo que se vuelve más difícil la resolución de problemas.
4. El sistema de comunicación con el AFIS Civil es muy complejo porque incluye otros procesos como dactiloscopia (o verificación manual), se utiliza con otros objetivos (como el transporte de huellas) y presenta transacciones de inserción al AFIS (lo cual genera una lógica mas complicada para resolución de conflictos de identidades)
5. Los sistemas clientes de AFIS pertenecientes a SAIME son muy dependiente de las tecnologías usadas en este. Por lo que no permite adaptar fácilmente a otras aplicaciones aunque tiene una arquitectura bastante flexible.

2.3 Modelo de Dominio

La comprensión de la situación actual, tras un análisis profundo, conllevó a la confección de un modelo de dominio que expresa los conceptos más significativos relacionados entre sí. Conceptos de sistemas y servicios, que vienen a jugar un papel relevante en el futuro sistema.

Un modelo del dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema. Su diagrama muestra claramente el funcionamiento del entorno en el que se encuentra el negocio. Cuyos procesos no fueron delimitados con precisión hasta obtener un diagrama de casos de usos correspondiente.

El presente modelo especifica las relaciones que existen entre los principales conceptos de los procesos que se llevan a cabo para la transmisión de los datos biométricos desde las Oficinas Regionales de SAIME, donde están las aplicaciones de SAIME y hasta el AFIS Civil, enviando distintos tipos de solicitudes y recibiendo los distintos tipos de respuestas AFIS, a través de un servicio local y otro central.

2.4 Diagrama de Clases del Modelo de Dominio

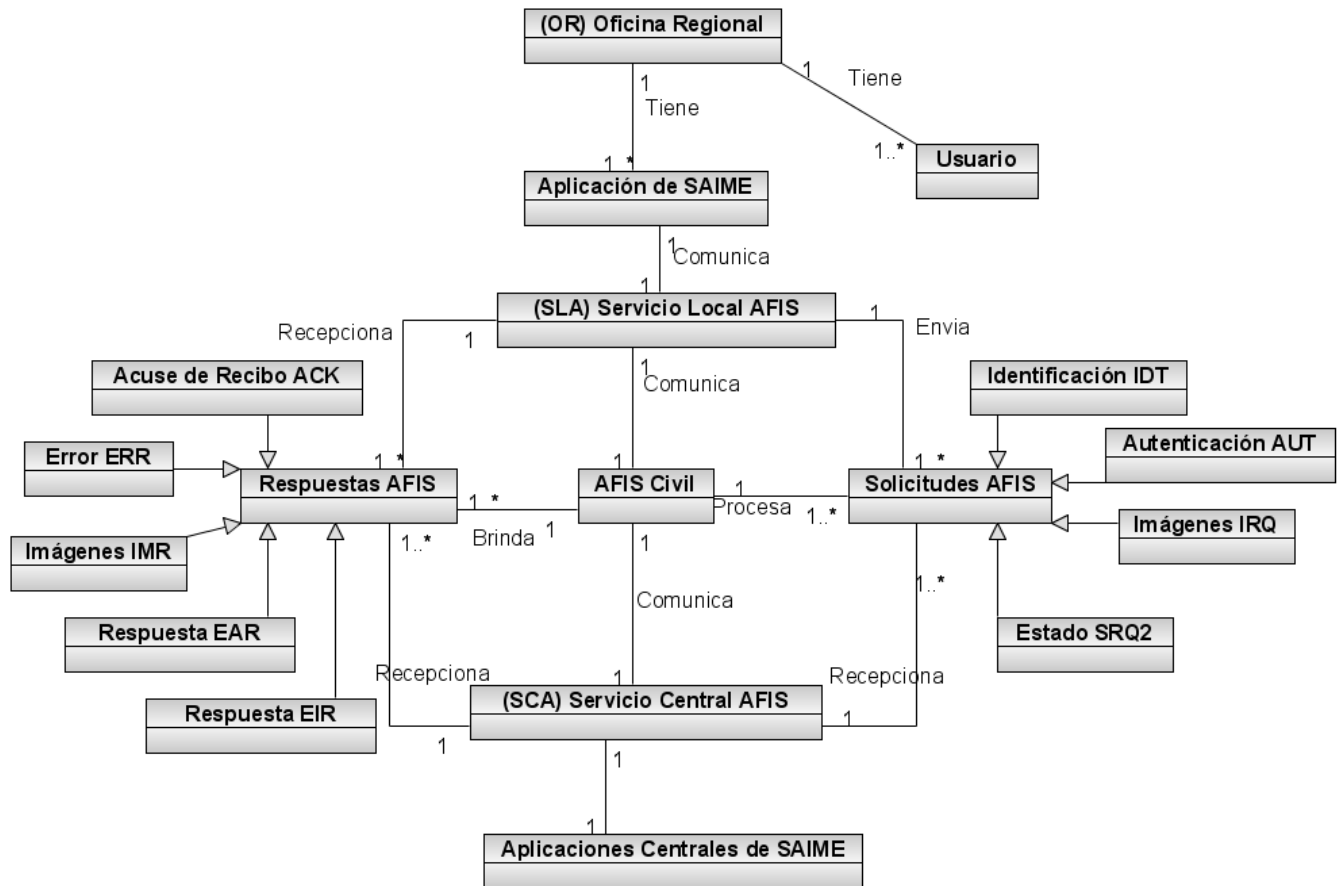


Figura 2.1: Modelo de Dominio

2.5 Glosario de Términos del Modelo de Dominio

- ✓ **AFIS Civil:** Es un sistema automatizado de identificación de huellas dactilares, sus huellas son planas y son capturadas en vivo para hacer la verificación.
- ✓ **Aplicación de SAIME:** Es el programa que usan los usuarios del sistema para los distintos procesos que se realizan, se encuentra instalada en un puesto de trabajo y posee un estado de activación.
- ✓ **Oficina Regional (OR):** Es un concepto del sistema que se especializa en brindar un conjunto de servicios, a través de los funcionarios que operan en los puestos de trabajo que esta contiene. Es la unidad estructural del sistema y posee los datos de su ubicación física y el estado de activación en que se encuentra.

Capítulo 2: Características Del Sistema

- ✓ **Usuario:** Se refiere al individuo que tendrá una estrecha interacción con el sistema. Podrá efectuar determinadas operaciones en cualquier puesto de trabajo de la oficina, posee datos de la persona vinculada.
- ✓ **(SLA) Servicio Local AFIS:** Servicio que corre en el servidor de la oficina. Su objetivo es constituir una interfaz de comunicación asincrónica, por medio de correos electrónicos, entre las aplicaciones que usan la BD Local de la oficina regional y los componentes biométricos centrales que participan en el proceso de verificación biométrica.
- ✓ **(SCA) Servicio Central AFIS:** Servicio que tiene como objetivo constituir una interfaz de comunicación asincrónica entre las aplicaciones que usan la BD Central de la OR y los componentes biométricos remotos que participan en el proceso de verificación biométrica, como los SLA y el AFIS Civil.
- ✓ **Respuestas del AFIS:** Se refiere a los mensajes que se reciben del AFIS Civil como respuesta a una funcionalidad solicitada por el cliente mediante una Solicitud AFIS.
- ✓ **Acuse de Recibo ACK:** Tipo de respuesta AFIS, que le informa a la oficina que la solicitud llegó al AFIS y este la está procesando.
- ✓ **Error ERR:** Tipo de respuesta AFIS. Consiste en un mensaje de error.
- ✓ **Imágenes IMR:** Tipo de respuesta AFIS respecto a las imágenes.
- ✓ **Respuesta EAR:** Tipo de respuesta AFIS referente a la Autenticación.
- ✓ **Respuesta EIR:** Tipo de respuesta AFIS referente a la Identificación.
- ✓ **Solicitudes del AFIS:** Se refiere a los mensajes que se le puede enviar al AFIS Civil para solicitarle una determinada funcionalidad.
- ✓ **Identificación IDT:** Tipo de solicitud AFIS. Se refiere a la identificación. El objetivo principal de este procedimiento es identificar a una persona.
- ✓ **Autenticación AUT:** Tipo de solicitud AFIS. Se refiere a la autenticación. El objetivo principal de este procedimiento es autenticar a una persona: validar que corresponde a la persona que tiene el mismo identificador.
- ✓ **Imágenes IRQ:** Tipo de solicitud AFIS. Se refiere a la solicitud de Imágenes.
- ✓ **Estado SRQ:** Tipo de solicitud AFIS. Se refiere a la solicitud de estado.

2.6 Sistema

2.6.1 Propuesta del sistema

Se propone una infraestructura de servicios de comunicación que permita a las aplicaciones desarrolladas por entidades externas a ONIDEX, utilizar de manera remota los servicios del AFIS Civil de SAIME de manera transparente, a través de medios de comunicación susceptibles a inestabilidades, mediante mensajería asincrónica.

Como una de las principales ventajas aportadas por la infraestructura de servicios propuesta está el hecho de que implementa un modelo de centralización de las solicitudes lo que es una base importante para futuros trabajos de control de flujo inteligente y unificado, estadística y control centralizado de los servicios biométricos prestados a entidades externas, y una simplificación del modelo de trabajo que permite verificaciones menos complicadas tecnológicamente. El control centralizado brindado por este esquema, está muy a tono con lo que necesita SAIME para la posible generación de ingresos a partir de los servicios ofrecidos a entidades externas.

Además se propone un conjunto de servicios de monitoreo y de interfaz con el cliente, como punto de partida para obtener la funcionalidad que necesite del AFIS Civil, brindándole cierto control sobre las transacciones que está generando. También incluye un conjunto de servicios centrales para el monitoreo y control por parte del proveedor de servicio biométrico AFIS a sus clientes.

En la solución propuesta los servicios ubicados en el entorno del cliente enviarán las transacciones hacia un servicio central que las guardará en la BD Central y las reenviará guiándose por el flujo permitido hacia el sistema AFIS. Del mismo modo gestionará las respuestas de AFIS para enviarlas al sistema cliente.

El sistema propuesto envía solo transacciones de identificación y autenticación, excluyendo procesos que no son necesarios para clientes externos a SAIME como dactiloscopia o excluyendo la funcionalidad de usar las huellas de transacciones para el transporte de huellas a otros procesos de sede central, haciéndolo más sencillo.

La figura 2.2 representa el sistema que se quiere implementar.

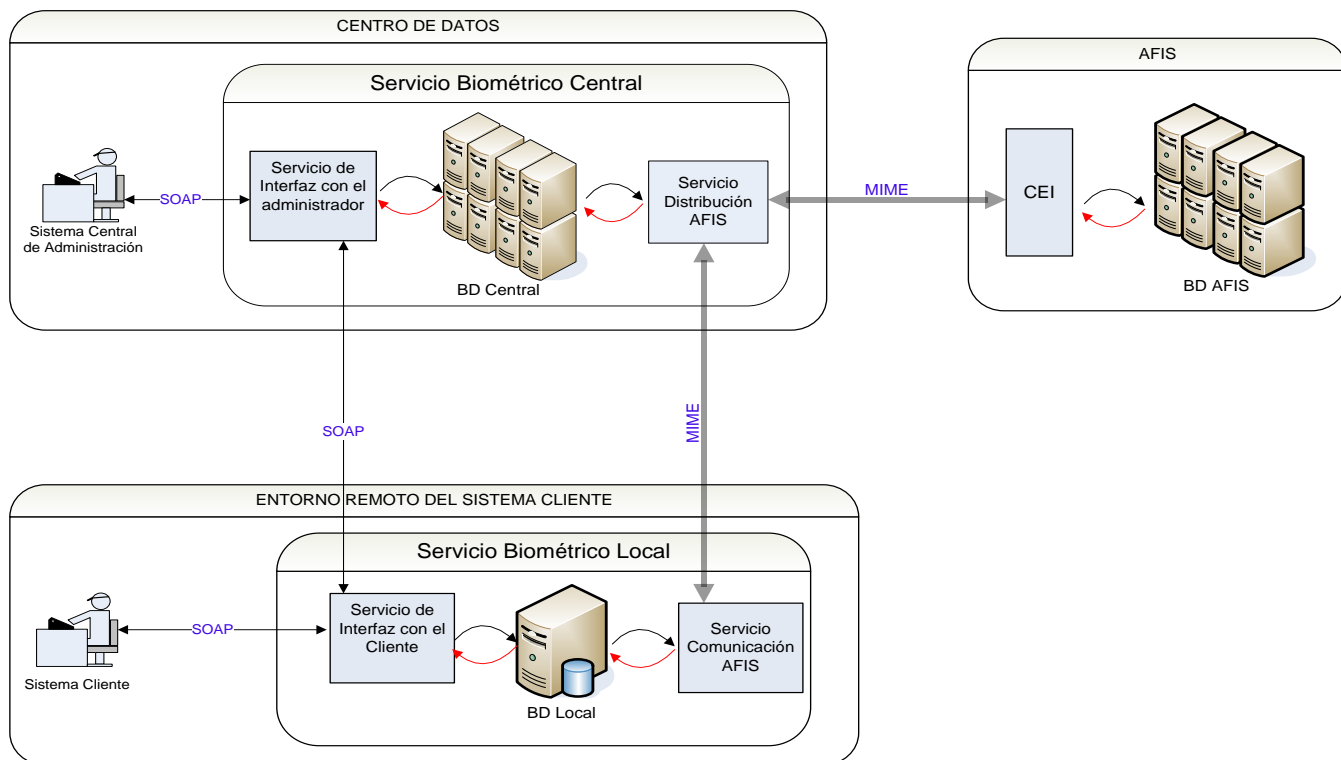


Figura 2.2: Propuesta del Sistema

2.7 Especificación de requisitos

2.7.1 Requerimientos Funcionales

Servicio de Distribución AFIS (central)

RF - 1. Recibir transacciones. Estas pueden ser Respuestas AFIS y Solicitudes enviadas por el servicio de comunicación AFIS (SCA).

RF - 1.1. Establecer una sesión IMAP con el servidor de correos y enviarle los comandos necesarios para recibir el mensaje.

RF - 1.2. Decodificar el mensaje electrónico MIME y extraer el contenido equivalente al fichero adjunto.

RF - 1.3. Interpretar ese fichero obtenido como un formato XML y extraerle los datos de la solicitud.

Capítulo 2: Características Del Sistema

RF - 1.4. Escribir en la BD asociada los datos correspondientes a la transacción recibida.

RF - 1.4.1. Como dirección de origen de la transacción en BD tomar la dirección de origen del correo recibido.

RF - 1.4.2. Como dirección de destino de la transacción en BD:

RF - 1.4.2.1. Si la transacción es una solicitud que proviene del SCA de un cliente remoto tomar la dirección de correo del AFIS.

RF - 1.4.2.2. Si la transacción es una respuesta que proviene del AFIS tomar la dirección de correos del origen de la solicitud correspondiente. (la del SCA que originó la solicitud)

RF - 1.4.2.3. Establecer la transacción como lista para enviar.

RF - 2. Enviar transacciones. Estas pueden ser solicitud al AFIS o respuesta al servicio de comunicación.

RF - 2.1. Consultar desde la BD el identificador de la próxima transacción pendiente a enviar respetando las prioridades que estas poseen.

RF - 2.2. Obtener los datos desde la BD asociada al servicio.

RF - 2.3. Elaborar el paquete XML con estos datos.

RF - 2.4. Codificar un mensaje electrónico MIME con el fichero XML adjunto.

RF - 2.5. Establecer una sesión SMTP con el servidor de correos y enviarle los comandos necesarios para transferirle el mensaje.

Servicio de Comunicación AFIS (local)

RF - 3. Enviar solicitud al Servicio de Distribución AFIS.

RF - 3.1. Verificar que existe una nueva solicitud cada determinado período de tiempo.

RF - 3.2. Consultar desde la BD el identificador de la próxima transacción pendiente a enviar respetando las prioridades que estas poseen.

RF - 3.3. Obtener los datos desde la BD asociada al servicio.

RF - 3.4. Elaborar el paquete XML con estos datos.

RF - 3.5. Codificar un mensaje electrónico MIME con el fichero XML adjunto.

RF - 3.6. Establecer una sesión SMTP con el servidor de correos y enviarle los comandos necesarios para transferirle el mensaje.

RF - 4. Recibir respuestas AFIS.

RF - 4.1. Establecer una sesión IMAP con el servidor de correos y enviarle los comandos necesarios para recibir el mensaje.

RF - 4.2. Decodificar el mensaje electrónico MIME y extraer el contenido equivalente al fichero adjunto.

RF - 4.3. Interpretar ese fichero obtenido como un formato XML y extraerle los datos de la solicitud.

RF - 4.4. Escribir en la BD asociada los datos correspondientes.

Servicio Interfaz con el Cliente (Servicio Web sobre PHP)

RF - 5. Crear verificaciones.

RF - 5.1. Obtener los datos de la verificación desde el cliente, incluyendo el identificador biométrico (BIOID).

RF - 5.2. Obtener los datos necesarios para realizar una verificación al AFIS.

RF - 5.3. Escribir en Base de Datos los datos correspondientes a la solicitud que se quiere enviar.

RF - 6. Generar listado de verificaciones.

RF - 6.1. Obtener los datos correspondientes en la BD Local de las verificaciones creadas.

RF - 6.2. Mostrar listado de verificaciones.

Servicio Interfaz con el Cliente /Servicio de Administración

RF - 7. Generar listado de transacciones. Solicitudes y respuestas AFIS.

RF - 7.1. Mostrar listado de solicitudes hechas y de respuestas recibidas, permitiendo supervisar el estado de las solicitudes y consultar el resultado de la verificación.

RF - 7.1.1. Las solicitudes pueden tener diferentes estados:

RF - 7.1.1.1. Pendiente. Pendiente de envío.

RF - 7.1.1.2. Enviada. Enviada por el servicio, sin recibir respuestas aún.

RF - 7.1.1.3. En proceso. Recibida exitosamente por el AFIS.

RF - 7.1.1.4. Error. Respuesta de error del AFIS.

RF - 7.1.1.5. Completada. Completada exitosamente.

RF - 7.1.1.6. Relanzada.

RF - 7.2. Permitir reenviar las solicitudes que tengan un estado de error, estado terminal no exitoso.

RF - 8. Permitir realizar las solicitudes de estado (SRQ) al AFIS para las solicitudes que hayan demorado mucho.

Servicio de Administración / Configuración / Monitoreo

RF - 9. Generar reportes de solicitudes procesadas en el Servicio Central.

RF - 9.1. Controlar la cantidad de solicitudes que se procesan correctamente de las aplicaciones externas.

RF - 9.2. Controlar cantidad de solicitudes con error por cliente.

RF - 9.3. Controlar cantidad de solicitudes por tipo y por cliente.

RF - 9.4. Controlar tiempo de demora promedio por solicitud.

2.7.2 Requisitos No Funcionales

1. Software:

- ✓ Los servicios web están desarrollados sobre el lenguaje de programación PHP 5.0.
- ✓ Los servicios web utilizarán como base de datos PostgreSQL.
- ✓ Se utilizará como servidor Web Apache 2.0.

2. Seguridad y Privacidad:

- ✓ Se establecerá una llave privada entre el cliente y el proveedor del servicio web, para evitar acceso no autorizado a los servicios.

3. Portabilidad:

- ✓ La portabilidad se observa en los servicios Web, los cuales podrán ser accedidos desde cualquier protocolo de transporte que sea capaz de transportar texto, como por ejemplo (http,

https, SMTP, etc.), a través del puerto 80, pues este puerto no es bloqueado ya que es el que utilizan los navegadores.

- ✓ El diseño del servicio web permite que el mismo pueda funcionar tanto en sistemas operativos libres como propietarios.

4. Restricciones en el diseño y la implementación:

- ✓ Se usa el lenguaje de programación PHP tanto para los servicios como para la aplicación que los consume.
- ✓ Se usa la librería NuSoap para la creación de los servicios Web.
- ✓ Se usa la librería PGSQL para la abstracción a datos.

5. Usabilidad:

- ✓ El uso de servicios web permite que la aplicación pueda consumir estos aunque esté escrita en una plataforma de desarrollo diferente a ellos, pues su protocolo de comunicación está basado en XML y es compatible con todas las plataformas.
- ✓ El uso de los servicios web es muy práctico ya que pueden aportar gran independencia entre la aplicación que usa el servicio web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro.

6. Legales:

- ✓ La aplicación y toda la documentación generada pertenecen a la Universidad de las Ciencias Informáticas.

7. Disponibilidad:

- ✓ La aplicación debe estar disponible las 24 horas de forma tal que se pueda acceder a todos sus servicios, y así aprovechar todas sus funcionalidades.

8. Importación y exportación de datos:

- ✓ El sistema deberá almacenar todos los datos en una base de datos PostgreSQL donde puedan ser accedidos por los servicios.

9. Hardware:

- ✓ El servidor local donde estarán montados los servicios y la base de datos requiere como mínimo de RAM 1 GB y 160 GB de disco duro.
- ✓ El servidor central donde estarán montados los servicios y la base de datos requiere como mínimo de RAM 2 GB y 300 GB de disco duro.

2.8 Diagrama de Casos de Usos de Sistema

Los requisitos funcionales definen las funcionalidades del sistema. Apoyados en ellos se obtienen los casos de uso del sistema, artefacto del flujo de trabajo Requerimientos que describe el comportamiento del sistema desde el punto de vista del usuario. Los siguientes diagramas de casos de uso del sistema se encuentran divididos por paquetes (ver este diagrama en Anexos II) según las responsabilidades que atiende. Su representación gráfica permite comprender fácilmente qué hace el sistema.

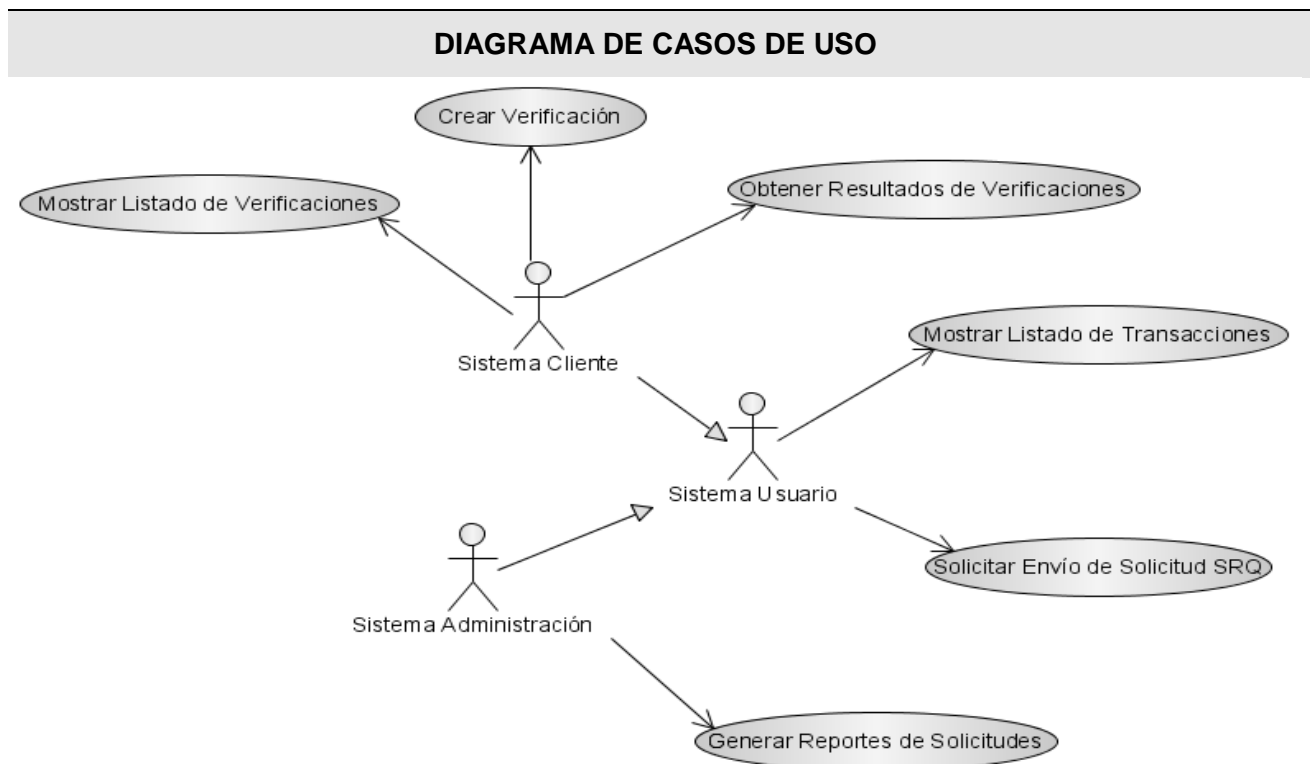


Figura 2.3: Diagrama de CUS para el paquete Servicio de Interfaz con el Cliente y Administrador (Servicio Web)

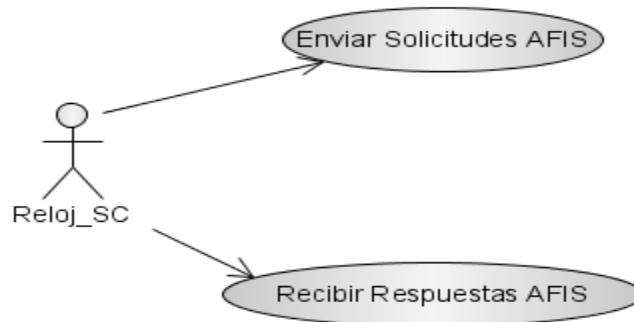


Figura 2.4: Diagrama de CUS del paquete Servicio de Comunicación AFIS (SCA)

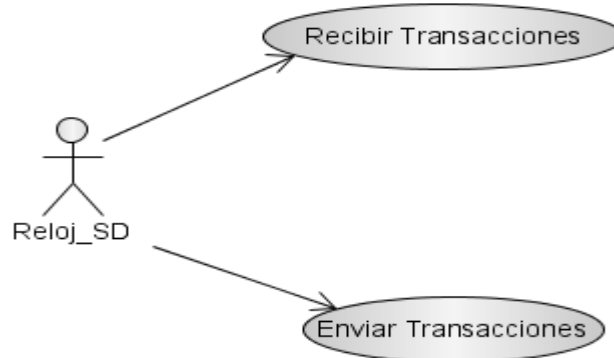


Figura 2.5: Diagrama de CUS del paquete Servicio de Distribución AFIS (SDA)

2.9 Definición de los casos de usos del sistema

2.9.1 Actores del sistema

Tabla 1 Descripción de los actores del sistema

Actores:	Justificación
Sistema Cliente	Actor que inicializa las acciones del servicio de interfaz con el cliente. Se refiere a la aplicación cliente que tiene relación directa con el servicio web a través de una conexión SOAP.
Sistema Administración	Actor que inicializa las acciones del servicio de interfaz con el administrador. Se refiere a la aplicación del administrador que tiene relación directa con el servicio web a través de una conexión SOAP.
Sistema Usuario	Generalización de los actores sistema cliente y sistema administración. Inicializa acciones que pueden ser iniciadas por cualquiera de los dos actores especializados.
Reloj_SC	Actor ficticio que inicializa y controla cada un determinado tiempo la existencia de una nueva verificación hecha y

Capítulo 2: Características Del Sistema

	registrada en la base de datos local, además de la llegada de un nuevo mensaje de respuesta AFIS enviada por el servicio de distribución.
Reloj_SD	Actor ficticio que inicializa y controla cada determinado tiempo la recepción y envío de transacciones.

2.10 Descripción Textual de CUS

La siguiente tabla especifica la descripción del caso de uso principal correspondiente a las funcionalidades del Servicio de Interfaz con el Cliente (Servicio Web).

Tabla 2: Descripción del CUS "Crear Verificación"

Caso de Uso:	Crear Verificación	
Actores:	Sistema Cliente	
Propósito:	Crear una verificación.	
Resumen:	El sistema cliente obtiene los datos necesarios de la solicitud de verificación hecha por el cliente, entre los datos están las huellas dactilares y número de cédula. Procede a crear una verificación para ser enviada al AFIS, escribe dicha verificación en la BD local, la misma queda en estado pendiente (para ser enviada al AFIS a través del servicio de comunicación).	
Precondiciones:	El sistema cliente verifica que la persona tenga BIOID en la base de datos central.	
Referencias	RF 5 (5.1, 5.2, 5.3)	
Prioridad	Crítica	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El sistema cliente obtiene los datos introducidos y se conecta al servicio web. Los datos son: 1.1. Letra y número de cédula. 1.2. Tipo de solicitud. 1.3. Si es juvenil o no.		
	2. El servicio web se conecta a la base de datos y obtiene el BIOID correspondiente a la	

Capítulo 2: Características Del Sistema

	cédula introducida; comprobando así su existencia. Si no existe el BIOID ver flujos alternos.
3. El sistema cliente capta las huellas.	
	4. El servicio obtiene y guarda en la base de datos local las huellas captadas por el sistema cliente.
	5. El servicio escribe en la BD local la nueva verificación y devuelve mensaje informándolo al sistema cliente.
Flujos Alternos	
Acción del actor	Respuesta del Sistema
	2.1 El servicio web devuelve un mensaje de error si no existe el BIOID.
	4.1. El servicio web informa a través de un mensaje de error que la verificación no fue creada satisfactoriamente.
Poscondiciones	Se creó la verificación.

A continuación se muestran las tablas para la descripción de los casos de usos que atienden a las funcionalidades de los Servicios de Comunicación.

Tabla 3: Descripción del CUS “Enviar Solicitudes”

Caso de Uso:	Enviar Solicitudes.
Actores:	Reloj_SC
Propósito:	Enviar solicitudes al servicio de distribución AFIS.
Resumen:	Periódicamente, el servicio de comunicación AFIS realiza una revisión en la BD Local hasta detectar una solicitud con estado pendiente. Luego se conforma un mensaje de correo con los datos de la solicitud detectada y se envían al servicio de distribución AFIS.
Precondiciones:	El servicio de interfaz con el cliente ha creado una solicitud y la ha escrito en la BD Local.
Referencias	RF 3 (3.1, 3.2, 3.3, 3.4, 3.5. 3.6)
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

Capítulo 2: Características Del Sistema

1. El reloj_SC chequea y detecta la existencia de una nueva verificación en la BD local respetando su prioridad.	
	2. El servicio obtiene los datos correspondientes al tipo de solicitud encontrada, de la BD. 2.1. Tipo y estado de la solicitud. 2.2. Imágenes de huellas captadas. 2.3. TCN (Identificador de una transacción) 2.4. Origen y destino de la transacción.
	3. El servicio comunicación conforma el fichero adjunto teniendo en cuenta estos datos.
	4. El servicio codifica un email con el fichero adjunto conformado.
	5. El servicio establece una sesión SMTP con el servidor de correos.
	6. El servicio envía el mensaje de correo al servicio de distribución.
Poscondiciones:	Se envió la solicitud.

Tabla 4: Descripción del CUS “Recibir Transacciones”

Caso de Uso:	Recibir Transacciones.
Actores:	Reloj_SD
Propósito:	Recibir solicitudes y respuestas AFIS
Resumen:	El reloj_SD realiza una revisión cada un determinado tiempo para detectar la existencia de transacciones enviadas al servicio de distribución, tanto por el servicio de comunicación y el sistema AFIS, las cuales recibe. Estas transacciones se escriben en BD Central posibilitando su monitoreo por parte del administrador.
Precondiciones:	El servicio de comunicación envió una solicitud. El AFIS envió una respuesta de una solicitud.
Referencias	RF 1 (1.1, 1.2, 1.3, 1.4, (1.41, 1.4.2, (1.4.2.1, 1.4.2.3)))
Prioridad	Crítico
Flujo Normal de Eventos	

Capítulo 2: Características Del Sistema

Acción del Actor	Respuesta del Sistema
1. El reloj_SD chequea y detecta la existencia de una transacción enviada.	
	2. El servicio de comunicación establece una sesión IMAP con el servidor de correos con los comandos necesarios para recibir el mensaje.
	3. El servicio recibe el mensaje y lo decodifica. 3.1. Extrae el fichero adjunto. 3.2. Del fichero con formato XML obtenido se extraen los datos de la transacción.
	4. Escribe en BD Central los datos correspondientes a la transacción recibida teniendo en cuenta: 4.1. Dirección de origen: la dirección de origen del correo recibido 4.2. Dirección de destino: 4.2.1. Si la transacción es una solicitud que proviene del SCA de un cliente remoto tomar la dirección de correo del AFIS. 4.2.2. Si la transacción es una respuesta que proviene del AFIS tomar la dirección de correos del origen de la solicitud correspondiente (la del SCA que originó la solicitud).
	5. El servicio establece la transacción como lista para enviar.
Poscondiciones	Se escribió en base de datos central la transacción recibida y queda lista para ser enviada.

Ver descripción de los demás casos de usos en Anexos II.

2.11 Conclusiones

El capítulo 2 permitió conocer el ambiente de la situación actual del futuro sistema. Se analizó y se trazó el negocio, en un diagrama con la interrelación de los conceptos principales que se destacaron, obteniéndose así un modelo de dominio.

Los diagramas de casos de usos del sistema vienen a resolver las dificultades detectadas. Para su confección se hizo un análisis exhaustivo y detallado de los distintos escenarios y sus beneficiados, obteniéndose finalmente 10 casos de usos, 4 actores y una generalización para dos de ellos.

Como resultado final del análisis de este apartado se obtiene una vista externa del sistema, es decir se describe lo que se espera de él a través de los diagramas de casos de usos del sistema. El siguiente paso estará estrechamente ligado a los casos de usos, ya que el propósito es detallarlos hasta obtener una vista interna del sistema descrita con el lenguaje de los programadores.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

“Análisis y diseño del sistema”, recoge esencialmente la descripción de la solución del software a través de diagramas de clases del análisis y del diseño. El análisis pretende traducir los requisitos especificados en el capítulo anterior a una especificación que describe cómo implementar el sistema. Es decir consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales.

El diseño por su parte, es un refinamiento del análisis hecho y tiene en cuenta los requisitos no funcionales, modelando cómo cumple el sistema sus objetivos.

3.2 Análisis

3.2.1 Modelo de clases del análisis

Se muestran en las siguientes figuras los diagramas de clases del análisis para los CUS principales respecto a los paquetes definidos en el capítulo anterior, que reúnen las funcionalidades para el servicio web, los servicios de comunicación y distribución. Los demás diagramas de clases del análisis se pueden ver en los Anexos III.

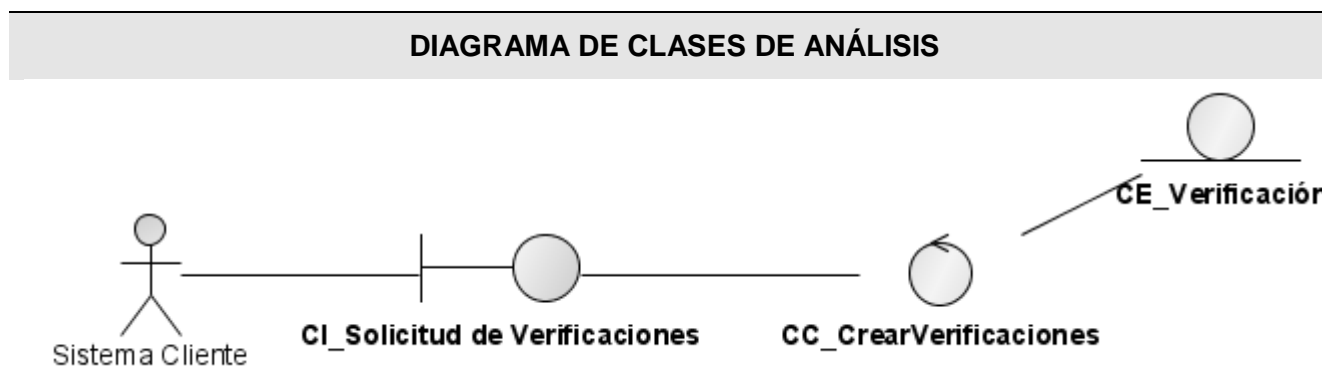


Figura 3.1: Diagramas de clases del análisis para CUS “Crear Verificación” del paquete Servicio de Interfaz con el Cliente y Administrador (Servicio Web)

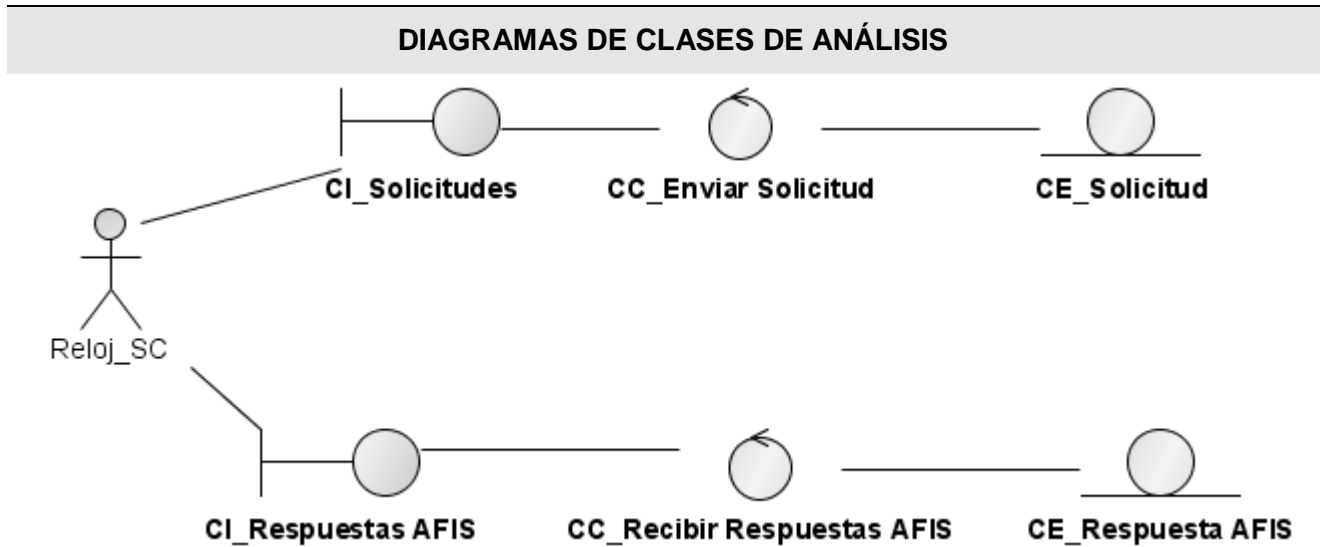


Figura 3.2: Diagramas de clases del análisis del paquete Servicio de Comunicación AFIS

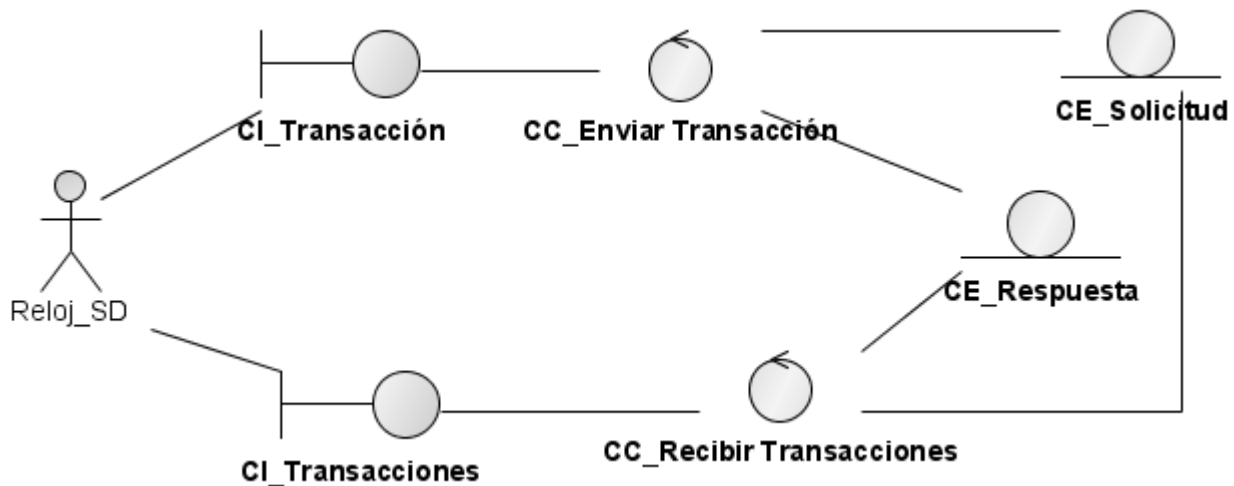


Figura 3.3: Diagramas de clases del análisis del paquete Servicio de Distribución AFIS

3.3 Diseño

El análisis hecho hasta este momento es el punto de partida, y la primera aproximación conceptual para una vez comprendidos los requisitos a este nivel, aumentar el nivel de especificidad en aras de garantizar el cubrimiento de los requisitos funcionales y no funcionales, considerando además el entorno de implementación.

3.3.1. Modelo de clases del diseño

La aplicación del cliente se conectará a través del protocolo de comunicación SOAP, y utilizando mensajes codificados en XML se comunicará con el servicio web. Para lograr una mejor comprensión de los diagramas de clases del diseño del servicio web, (que será el que brinde todas las funcionalidades al sistema cliente), se han separado los mismos en cuatro paquetes rigiéndose por su funcionalidad, como se muestra en la siguiente figura.

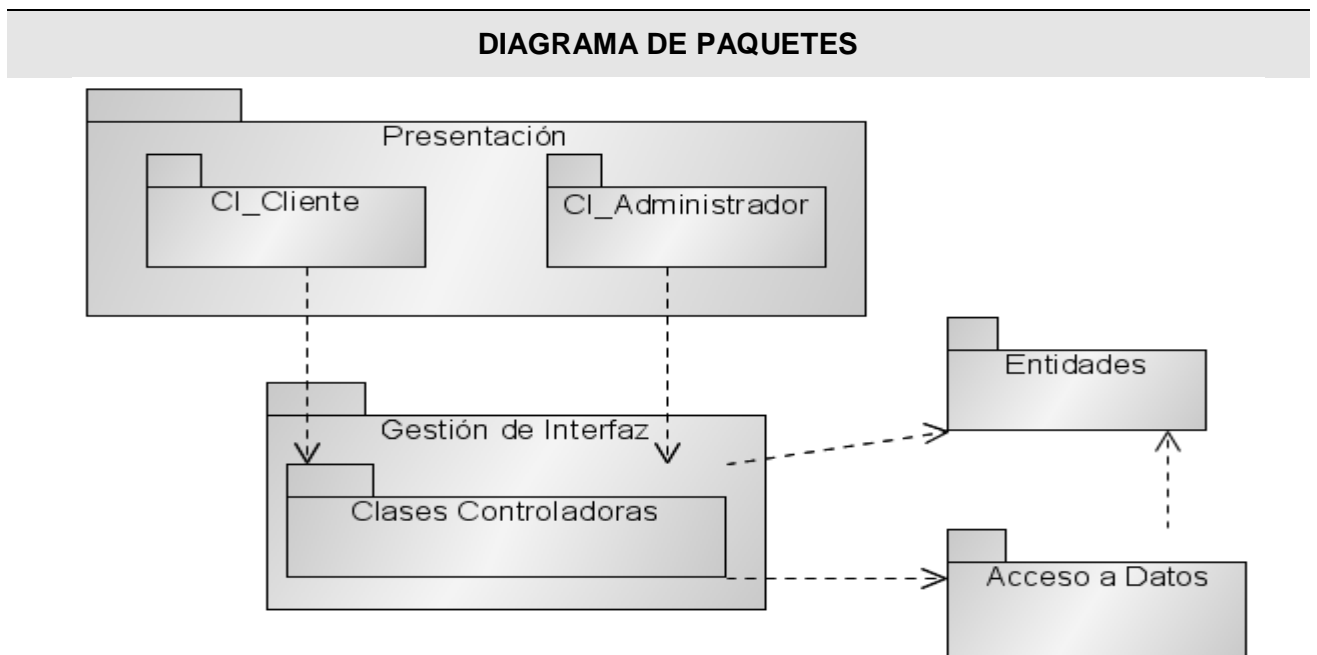


Figura 3.4: Diagrama de paquetes que agrupa las clases del diseño para el Servicio Web

El diagrama expuesto en la figura anterior se explica como sigue:

El paquete de presentación, contiene las clases interfaz del servicio para el sistema cliente y el administrador, donde están registrados todos los servicios que se exponen en el WSDL (Lenguaje de Descripción del Servicio Web).

El paquete de gestión de interfaz se encarga de la lógica del negocio, de la gestión de interfaz del cliente y el administrador a través de las clases controladoras de las entidades relacionadas en el negocio, estas clases permiten diversas funciones entre las que se puede mencionar la validación de los datos necesarios para llevar a cabo las funcionalidades del servicio web, ejemplo; crear una verificación.

El paquete *acceso_datos* contiene las clases para hacer posible la persistencia y recuperación de objetos, son las encargadas de acceder a la base de datos para manipular la persistencia de las entidades. El paquete *acceso_datos* en general permite a la aplicación abstraerse del origen de los datos y de la lógica de su persistencia, logrando un bajo acoplamiento entre sus componentes.

El paquete *entidades* contiene clases que no tienen comportamiento, sólo propiedades y son representaciones de entidades reales del dominio, la mayoría son clases persistentes que son accedidas por las clases de los paquetes *Gestión de Interfaz* y *Acceso Datos*.

A continuación se muestran los diagramas de clases del diseño para el servicio web.

3.3.1.1. Diagramas de clases del diseño por paquetes del Servicio Web

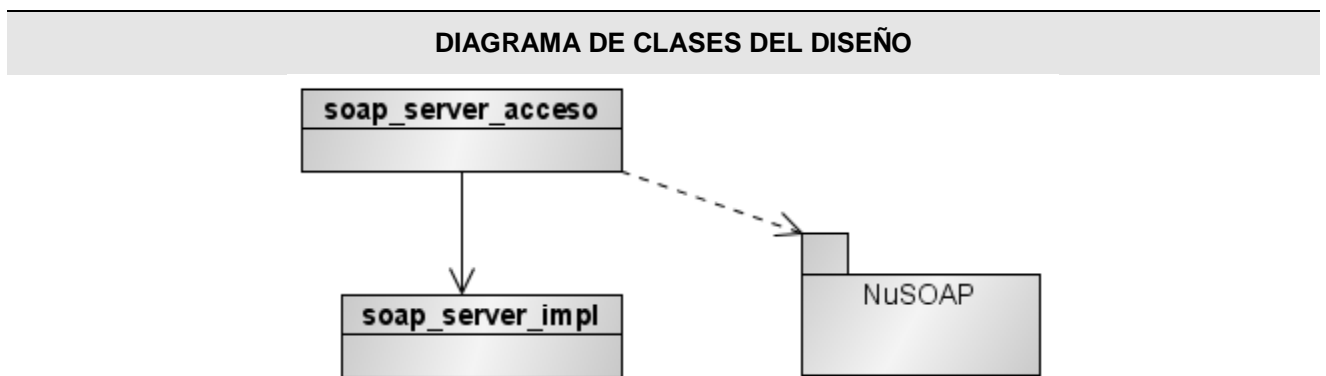


Figura 3.5: Diagrama de clases del diseño paquete de Presentación, CI_Cliente

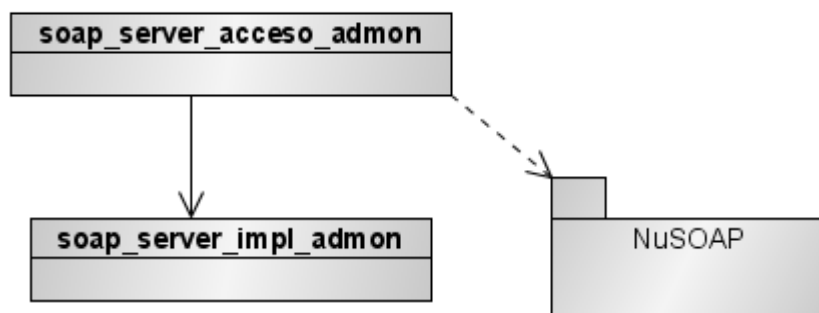


Figura 3.6: Diagrama de clases del diseño paquete de Presentación, CI_Administrador

DIAGRAMA DE CLASES DEL DISEÑO

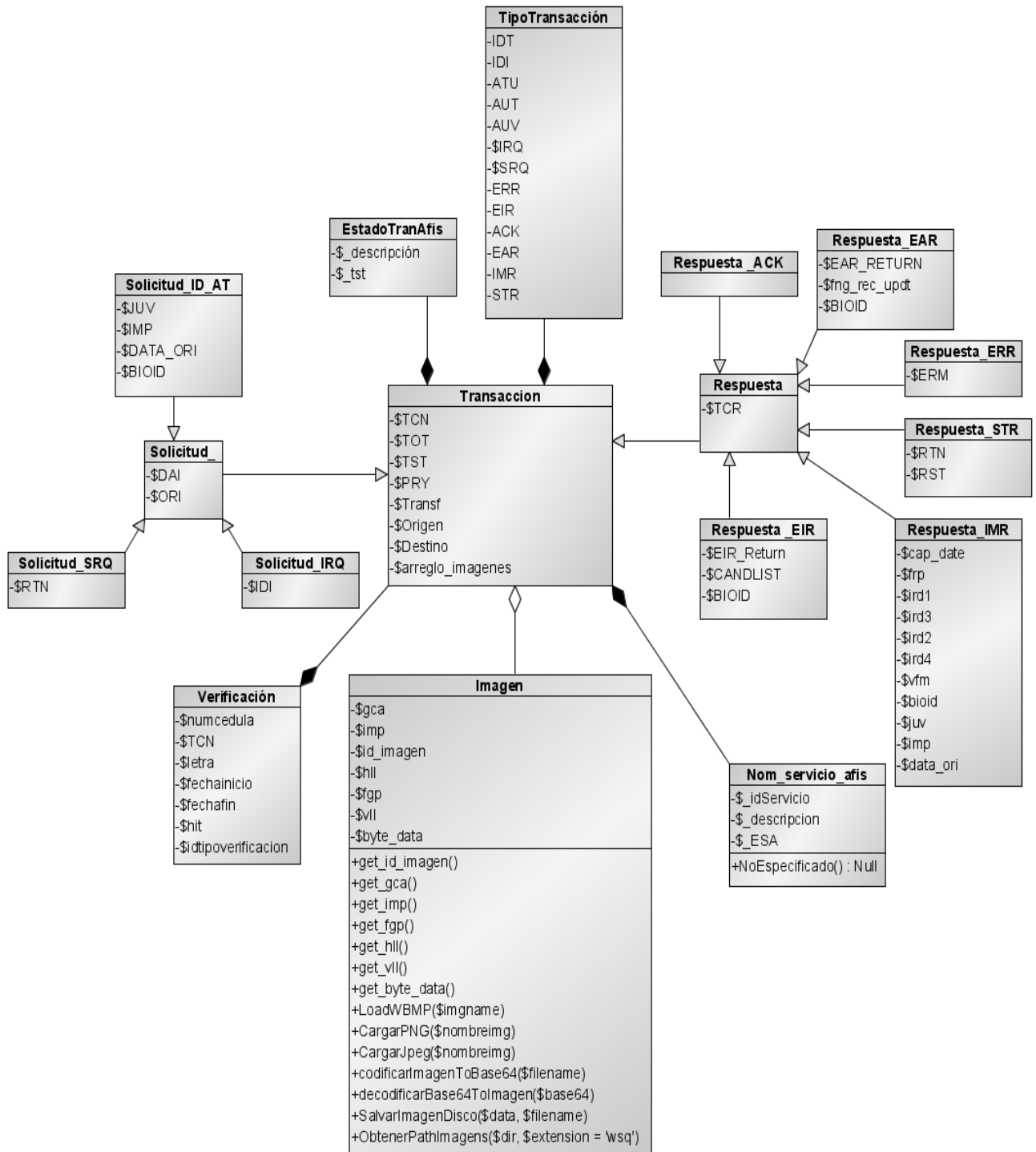


Figura 3.7: Diagrama de clases del diseño paquete de Entidades

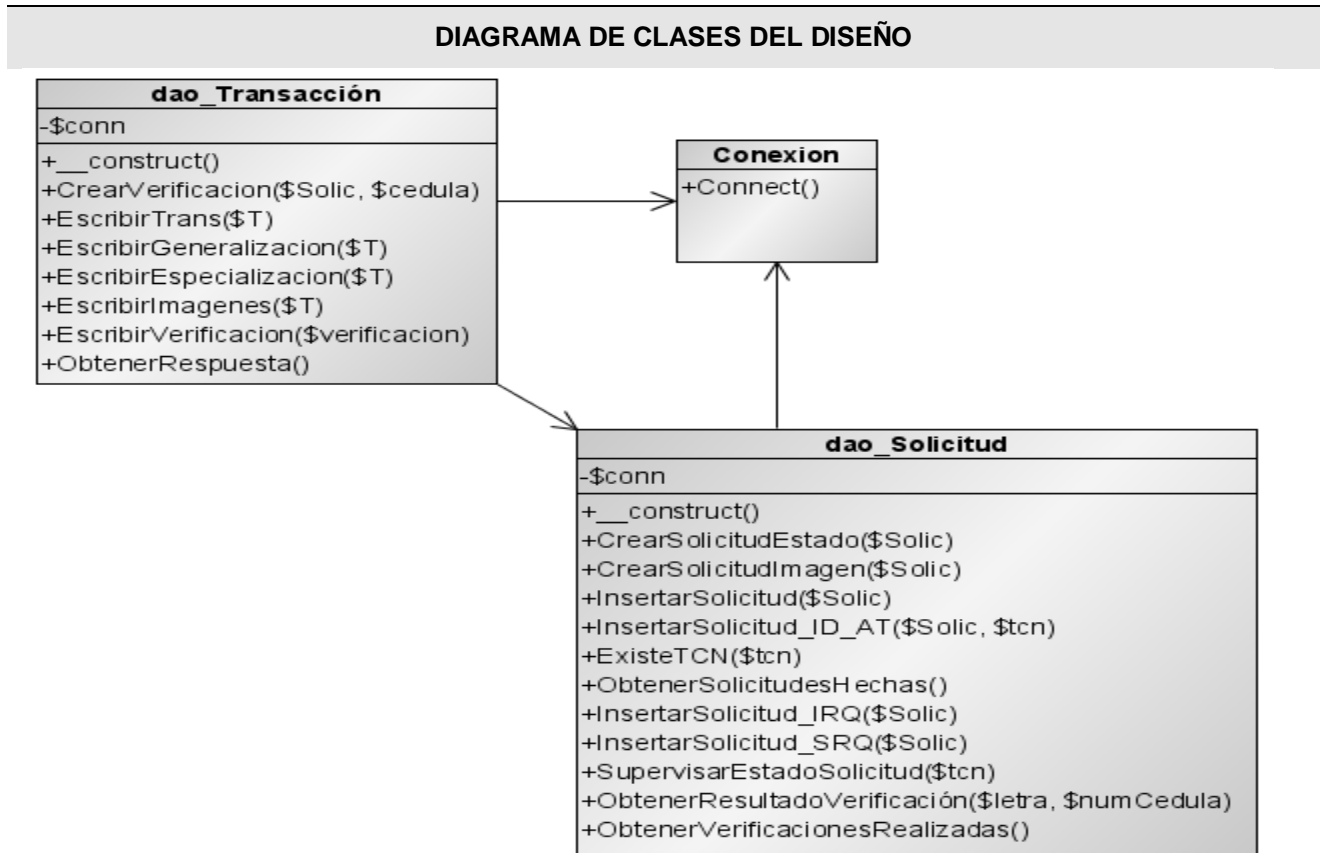


Figura 3.8: Diagrama de clases del diseño paquete de Acceso a Datos

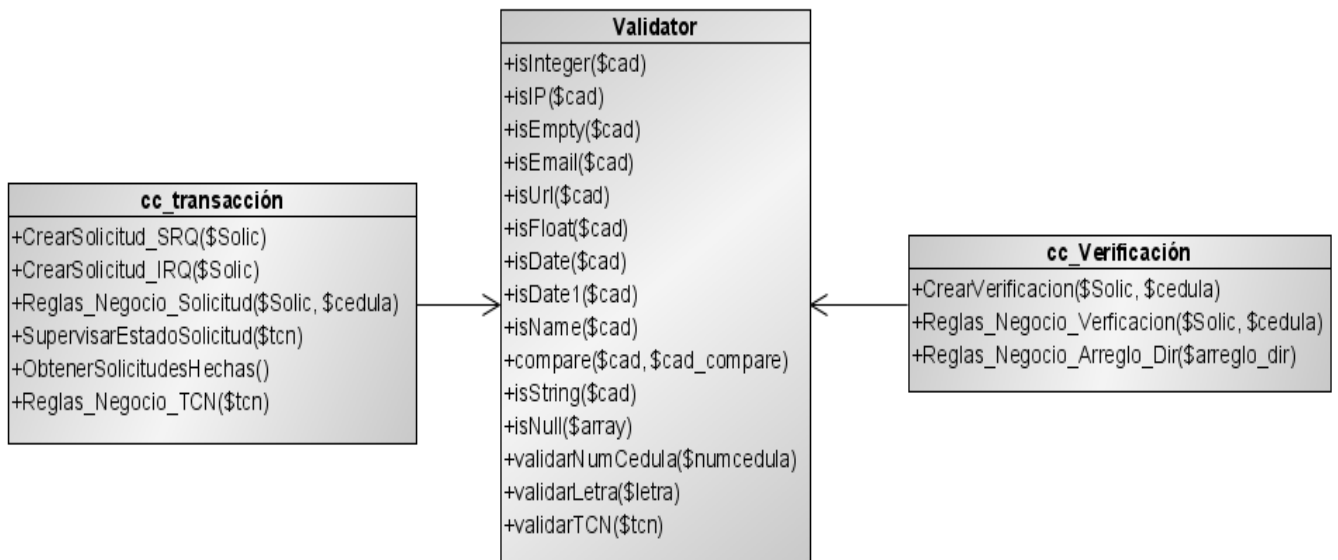


Figura 3.9: Diagrama de clases del diseño paquete de Gestión de Interfaz, clases controladora

3.3.1.1. Descripción de las clases del diseño del Servicio Web

Tabla 5: Descripción de Clase interfaz. CI_Cliente

Nombre: CI_Cliente	
Tipo de clase: Interfaz	
Atributo	Tipo
--	--
Para cada responsabilidad:	
Nombre:	CrearVerificacion(\$numcedula, \$letra, \$tot, \$arreglo_imagenes, \$juv, \$key)
Descripción:	Función que crea la verificación dado los datos enviados: número de cédula y letra, el tipo de solicitud y la especificación de si es juvenil o no, además de una clave para poder tener acceso a esta función.
Nombre:	ObtenerEstadoSolicitud(\$tcn)
Descripción:	Función encargada de dado los datos enviados obtener el estado de la solicitud correspondiente.
Nombre:	CrearSolicitud_SRQ(\$transf, \$tot, \$stt, \$pry, \$origen, \$destino, \$RTN, \$key)
Descripción:	Función encargada de crear una solicitud de estado dado los datos enviados además de una clave para el acceso a esta función.
Nombre:	CrearSolicitud_IRQ(\$transf, \$tot, \$stt, \$pry, \$origen, \$destino, \$IDI, \$key)
Descripción:	Función encargada de crear una solicitud de imágenes dado los datos enviados.
Nombre:	ObtenerSolicitudesHechas()
Descripción:	Función encargada de devolver los datos de todas las solicitudes realizadas.
Nombre:	ObtenerBioidDeUnaPersona(\$letra, \$nuncedula)
Descripción:	Función encargada de devolver el BIOID de una persona dado los datos enviados: letra y número de cédula.
Nombre:	ObtenerTodasVerificacionesRealizadas()
Descripción:	Función encargada de obtener todas las verificaciones realizadas.
Nombre:	ObtenerRespuestasRecibidas ()
Descripción:	Función encargada de obtener todas las respuestas recibidas desde el AFIS.
Nombre:	ObtenerResultadoVerificacion(\$letra, \$numcedula)
Descripción:	Función encargada de obtener el resultado de una verificación dado el número de cédula.

Tabla 6: Descripción de Clase interfaz. CI_Administrador

Nombre: CI_Administrador	
Tipo de clase: Interfaz	
Atributo	Tipo
--	--
Para cada responsabilidad:	
Nombre:	ObtenerBioidDeUnaPersona(\$letra, \$nuncedula)
Descripción:	Obtiene el identificador biométrico (BIOID) de una persona dado su letra y número de cédula.
Nombre:	ObtenerReporteSolicitudesCentral()
Descripción:	Obtiene un reporte de todas las solicitudes a nivel central.
Nombre:	ObtenerEstadoSolicitud(\$tcn)
Descripción:	Función encargada de dado los datos enviados obtener el estado de la solicitud correspondiente.
Nombre:	CrearSolicitud_SRQ(\$transf, \$tot, \$stst, \$pry, \$origen, \$destino, \$RTN, \$key)
Descripción:	Función encargada de crear una solicitud de estado dado los datos enviados además de una clave para el acceso a esta función.
Nombre:	CrearSolicitud_IRQ(\$transf, \$tot, \$stst, \$pry, \$origen, \$destino, \$IDI, \$key)
Descripción:	Función encargada de crear una solicitud de imágenes dado los datos enviados.
Nombre:	ObtenerSolicitudesHechas()
Descripción:	Función encargada de devolver los datos de todas las solicitudes realizadas.
Nombre:	ObtenerTodasVerificacionesRealizadas()
Descripción:	Función encargada de obtener todas las verificaciones realizadas.
Nombre:	ObtenerRespuestasRecibidas ()
Descripción:	Función encargada de obtener todas las respuestas recibidas desde el AFIS.

Tabla 7: Descripción de Clase controladora. CC_Transacción

Nombre: CC_Transacción	
Tipo de clase: Controladora	
Atributo	Tipo
--	--
Para cada responsabilidad:	
Nombre:	CrearSolicitud_SRQ(\$Solic)
Descripción:	Función responsable de crear una solicitud de estado si el dato enviado (solicitud) es correcto.

Nombre:	CrearSolicitud_IRQ(\$Solic)
Descripción:	Función responsable de crear una solicitud de imagen si el dato enviado (solicitud) es correcto.
Nombre:	Reglas_Negocio_Solicitud(\$Solic, \$cedula)
Descripción:	Función que tiene la responsabilidad de validar los datos enviados de una solicitud y la cédula.
Nombre:	SupervisarEstadoSolicitud(\$tcn)
Descripción:	Función responsable de obtener el estado de una solicitud dado su identificador.
Nombre:	ObtenerSolicitudesHechas()
Descripción:	Función responsable de devolver todas las solicitudes creadas.
Nombre:	Reglas_Negocio_TCN(\$tcn)
Descripción:	Función responsable de validar que el TCN sea correcto.

Tabla 8: Descripción de Clase controladora. CC_Verificación

Nombre: CC_Verificación	
Tipo de clase: Controladora	
Atributo	Tipo
--	--
Para cada responsabilidad:	
Nombre:	CrearVerificacion(\$Solic, \$numcedula, \$letra, \$idtipoverficacion)
Descripción:	Función responsable de crear una verificación si los datos enviados (solicitud, letra y número de cédula e identificador de tipo de verificación) son correctos.
Nombre:	Reglas_Negocio_Verficacion(\$Solic, \$numcedula, \$letra)
Descripción:	Función responsable de validar que los datos enviados de la verificación (solicitud, letra y número de cédula) sean correctos.
Nombre:	Reglas_Negocio_Arreglo_Dir(\$arreglo_dir)
Descripción:	Función responsable de validar un arreglo de direcciones.

Tabla 9: Descripción de Clase controladora. Conexión

Nombre: Conexión	
Tipo de clase: Entidad	
Atributo	Tipo
--	--
Para cada responsabilidad:	
Nombre:	Connect()

Descripción:	Responsable de realizar la conexión a la base de datos a través de la librería pgsqL.
--------------	---

Tabla 10: Descripción de Clase controladora. Dao_Solicitud

Nombre: Dao_Solicitud	
Tipo de clase: Entidad	
Atributo	Tipo
conn	Var
Para cada responsabilidad:	
Nombre:	CrearSolicitudEstado(\$Solic)
Descripción:	Responsable de crear una solicitud de estado dado una solicitud.
Nombre:	CrearSolicitudImagen(\$Solic)
Descripción:	Responsable de crear una solicitud de imagen.
Nombre:	InsertarSolicitud(\$Solic)
Descripción:	Responsable de insertar una solicitud en la base de datos.
Nombre:	InsertarSolicitud_ID_AT(\$Solic, \$tcn)
Descripción:	Responsable de insertar una solicitud de identificación o autenticación.
Nombre:	InsertarSolicitud_IRQ(\$Solic)
Descripción:	Responsable de escribir una solicitud de imágenes.
Nombre:	InsertarSolicitud_SRQ(\$Solic)
Descripción:	Responsable de insertar en la base de datos una solicitud de estado
Nombre:	ExisteTCN(\$tcn)
Descripción:	Devuelve verdadero si existe una solicitud dado el identificador TCN.
Nombre:	ObtenerSolicitudesHechas()
Descripción:	Responsable de obtener el listado de solicitudes hechas.
Nombre:	SupervisarEstadoSolicitud(\$tcn)
Descripción:	Obtiene el estado de una solicitud dado el identificador TCN.
Nombre:	ObtenerResultadoVerificación(\$letra, \$numCedula)
Descripción:	Obtiene el resultado de una verificación dado letra y número de cédula.
Nombre:	ObtenerVerificacionesRealizadas()
Descripción:	Obtener todas las verificaciones realizadas.

Tabla 11: Descripción de Clase controladora. Dao_Transaccion

Nombre: Dao_Transaccion	
Tipo de clase: Entidad	
Atributo	Tipo

conn	Var
Para cada responsabilidad:	
Nombre:	CrearVerificacion(\$Solic, \$numcedula, \$letra, \$idtipoverificacion)
Descripción:	Responsable de crear una verificación dado los datos enviados.
Nombre:	EscribirTrans(\$T)
Descripción:	Escribe una transacción en la base de datos ya sea una solicitud o una respuesta.
Nombre:	EscribirGeneralizacion(\$T)
Descripción:	Responsable de insertar la generalización de la transacción en la base de datos.
Nombre:	EscribirEspecializacion(\$T)
Descripción:	Responsable de insertar la especialización de la transacción en la base de datos.
Nombre:	EscribirImágenes(\$T)
Descripción:	Responsable de insertar las imágenes correspondientes de una transacción en la base de datos.
Nombre:	EscribirVerificacion(\$verificacion)
Descripción:	Inserta los datos correspondientes de una verificación en la base de datos.
Nombre:	ObtenerRespuesta()
	Obtener todas las respuestas recibidas por el AFIS.

Tabla 12: Descripción de Clase entidad. Transacción

Nombre: Transaccion	
Tipo de clase: Entidad	
Atributo	Tipo
\$TCN	Var
\$TOT	Var
\$TST	Var
PRY	Var
transf	Var
origen	Var
destino	Var
arreglo_imagenes	Var
Para cada responsabilidad:	
Nombre:	get_TCN()
Descripción:	Devuelve el TCN de una solicitud.
Nombre:	set_TCN(\$tcn)
Descripción:	Permite cambiar el valor del TCN.

Nombre:	get_TOT()
Descripción:	Devuelve tipo de transacción; TOT.
Nombre:	get_TST()
Descripción:	Devuelve el estado de una transacción; TST.
Nombre:	get_PRY()
Descripción:	Devuelve la prioridad de una transacción; PRY.
Nombre:	get_Transf()
Descripción:	Devuelve 0 ó 1 para conocer de donde es transferida la transacción.
Nombre:	get_Origen()
Descripción:	Devuelve el nombre de origen de la transacción.
Nombre:	get_Destino()
Descripción:	Devuelve el nombre de destino de la transacción.
Nombre:	get_imagenes()
Descripción:	Devuelve el arreglo de imágenes de huellas captadas correspondientes a la transacción.

Tabla 13: Descripción de Clase entidad. Solicitud

Nombre: Solicitud	
Tipo de clase: Entidad	
Atributo	Tipo
\$DAI	Var
\$ORI	Var
Para cada responsabilidad:	
Nombre:	--
Descripción:	--

Tabla 14: Descripción de Clase entidad. Respuesta

Nombre: Respuesta	
Tipo de clase: Entidad	
Atributo	Tipo
_TCR	Var
Para cada responsabilidad:	
Nombre:	get_TCR()
Descripción:	Devuelve el número de control de una transacción respuesta, el TCR.

Tabla 15: Descripción de Clase entidad. Verificación

Nombre: Verificacion	
Tipo de clase: Entidad	
Atributo	Tipo
TCN	Var
letra	Var
numcedula	Var
fechainicio	Var
fechafin	Var
Hit	Var
idtipoverificacion	Var
Para cada responsabilidad:	
Nombre:	get_cedula()
Descripción:	Devuelve letra y número de cédula.
Nombre:	get_tcn()
Descripción:	Devuelve el TCN de una verificación. TCN: identificador de una verificación correspondiente a una solicitud.
Nombre:	set_tcn(\$value)
Descripción:	Permite cambiar el TCN.
Nombre:	get_letra()
Descripción:	Devuelve la letra de una cédula (V o E).
Nombre:	get_numcedula()
Descripción:	Devuelve el número de cédula.
Nombre:	get_fecha_inicio()
Descripción:	Devuelve la fecha en que se inicio la verificación.
Nombre:	get_fecha_fin()
Descripción:	Devuelve la fecha de finalizada la verificación.
Nombre:	get_id_tipo_verificacion()
Descripción:	Devuelve e l identificador de la verificación.
Nombre:	get_hit()
Descripción:	Devuelve hit como respuesta del AFIS.

Tabla 16: Descripción de Clase entidad. Imagen

Nombre: Imagen	
Tipo de clase: Entidad	

Atributo		Tipo
id_imagen		Var
gca		Var
imp		Var
fgp		Var
hll		Var
vll		Var
byte_data		Var
Para cada responsabilidad:		
Nombre:	get_id_imagen()	
Descripción:	Devuelve el identificador de la imagen.	
Nombre:	get_gca()	
Descripción:	Devuelve el GCA (Algoritmo de compresión en escala de grises)	
Nombre:	get_imp()	
Descripción:	Devuelve el IMP (Tipo de impresión)	
Nombre:	get_fgp()	
Descripción:	Devuelve el FGP (Posición del dedo)	
Nombre:	get_hll()	
Descripción:	Devuelve el HLL (Longitud de línea horizontal de la imagen)	
Nombre:	get_vll()	
Descripción:	Devuelve VLL (Longitud de línea vertical de la imagen)	
Nombre:	get_byte_data()	
Descripción:	Devuelve un arreglo de byte de la imagen de la huella.	

Las clases descritas anteriormente corresponden a la clase interfaz del servicio web con el cliente y el administrador, en la cual están expuestos todos los servicios que brindan. Las clases de tipo controladoras poseen las responsabilidades encargadas del funcionamiento del servicio web. Y solo se especificaron en este apartado cinco de las del tipo entidades, pues de ellas se derivan las demás, de las cuales se puede ver su descripción en los Anexos III.

3.3.1.2. Diagrama de secuencia del diseño

DIAGRAMA DE INTERACCIÓN

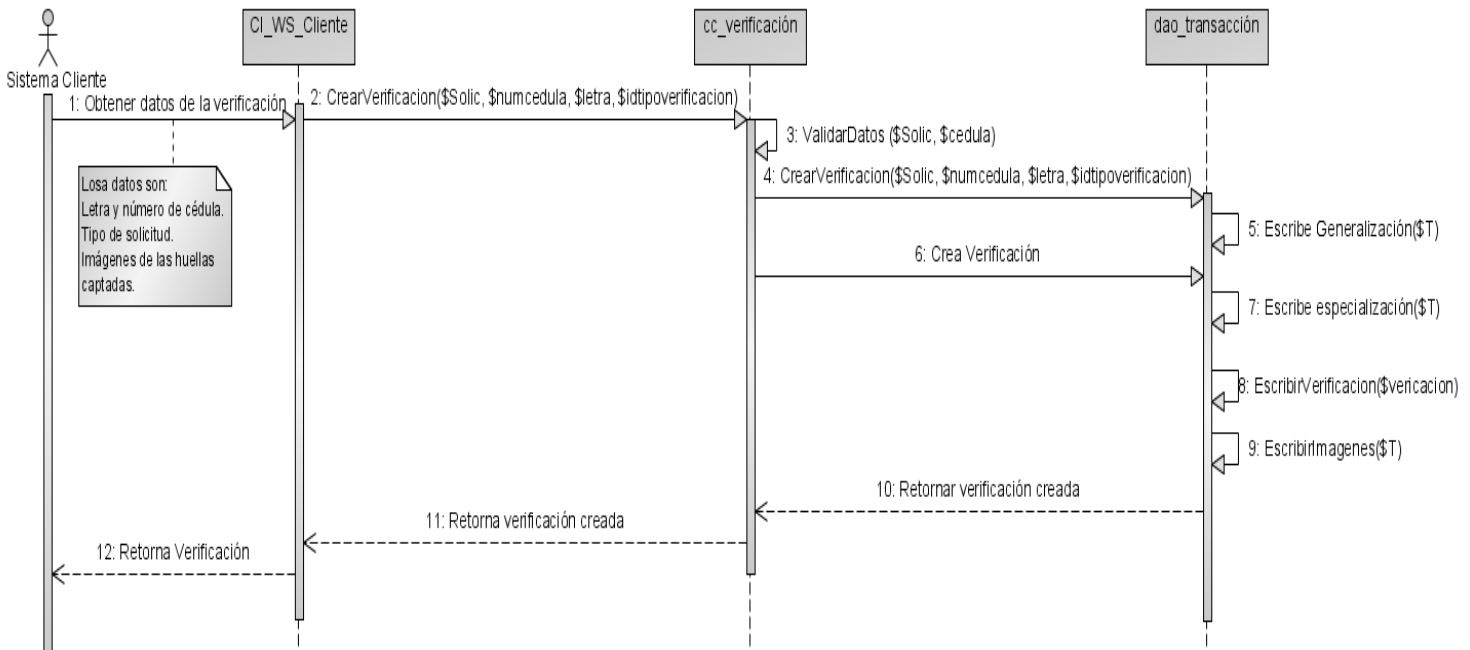


Figura 3.10: Diagrama de secuencia del diseño del caso de uso Crear Verificación

Se especifica el diagrama de secuencia del diseño para el caso de uso principal Crear Verificación del servicio web para el cliente, los demás diagramas de secuencias para los servicios web se pueden ver en los Anexos III.

3.3.2. Descripción de la arquitectura de los Servicios de Comunicación

Para realizar el envío de la solicitud escrita en BD, tanto el servicio de comunicación como el de distribución AFIS, procesan la información de la solicitud en el transcurso de varias fases que le permitirá conformar el mensaje electrónico con los datos correspondientes.

Primeramente se lee de la base de datos de las transacciones, se conforma el archivo XML, se codifica el mensaje MIME y a través del protocolo simple de transferencia de correos SMTP es enviado el mensaje conformado al servidor de correo Postfix.

Del servidor de correo, el servicio de comunicación lee los mensajes utilizando el protocolo de red de acceso a mensajes electrónicos almacenados en un servidor, IMAP, luego decodifica el mensaje MIME y conforma el XML que luego escribe en la base de datos.

Este proceso de envío y recepción entre los servicios sucede en dos direcciones de igual forma, la primera, cuando el servicio de comunicación envía una solicitud al servicio de distribución y este la recibe, y viceversa. La segunda cuando el servicio de distribución envía transacciones al AFIS y recibe otras de él.

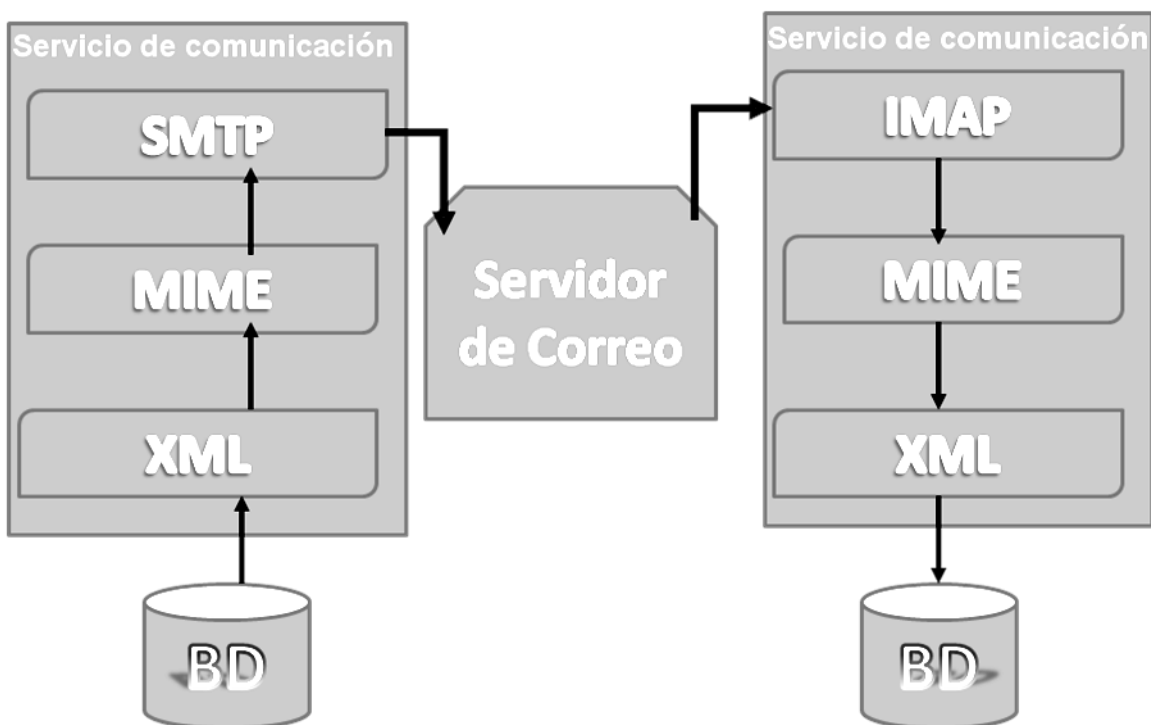


Figura 3.11 Arquitectura de comunicación interna entre los servicios de comunicación

3.3.2.1. Modelo de diseño de los Servicios de Comunicación

Los servicios web escribirán en la base de datos local las verificaciones creadas. El servicio de comunicación estará chequeando cada determinado tiempo la existencia de una nueva solicitud, lista para ser enviada al servicio de distribución. El diseño de los servicios que proveen este funcionamiento se especifica en los diagramas de clases del diseño.

A continuación se muestra el diseño empleado para la implementación de los servicios de comunicación. Los cuales se agruparon en tres paquetes atendiendo a sus responsabilidades; los mismos son: Persistencia, Entidades y Servicios.

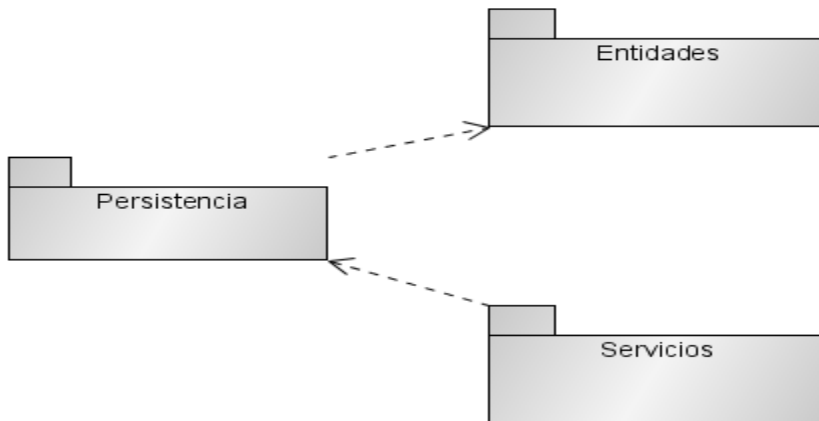


Figura 3.12 Diagrama de Paquetes de los Servicios de Comunicación

El diagrama de paquetes de los servicios de comunicación expuesto anteriormente se explica como sigue:

Entidades: Agrupa las clases del diseño pertenecientes a las transacciones y otras entidades pertenecientes al negocio de los servicios.

Persistencia: Agrupa las clases del diseño que responden a las funciones de persistencia correspondiente a los datos de las entidades.

Servicios: Agrupa las clases del diseño para la implementación de los servicios o demonios. En estas clases se unen todas las clases y funcionalidades descritas en las clases del paquete de persistencia logrando así el envío y recepción de las transacciones.

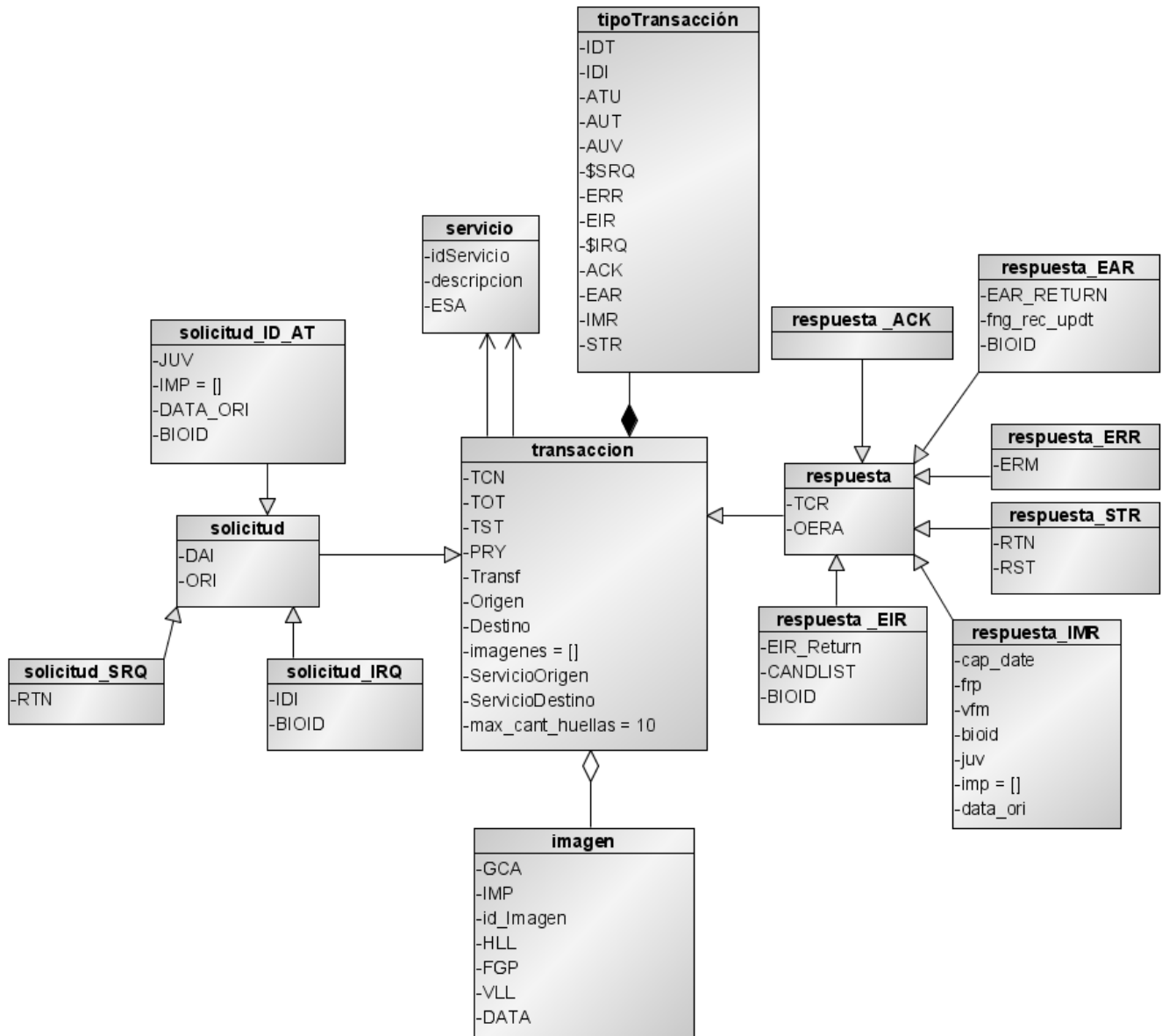


Figura 3.13 Diagrama de clases del diseño del paquete Entidades de los Servicios de Comunicación

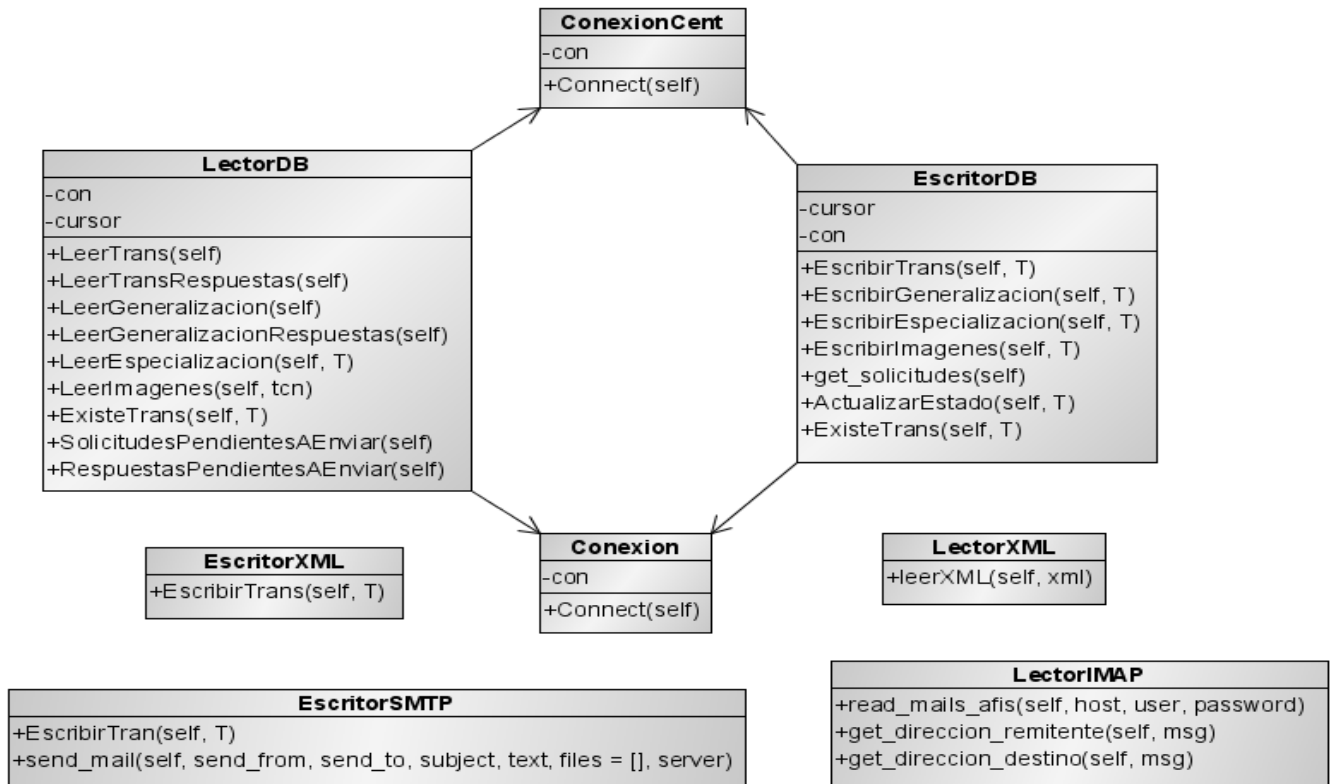


Figura 3.14 Diagrama de clases del diseño del paquete Persistencia de los Servicios de Comunicación

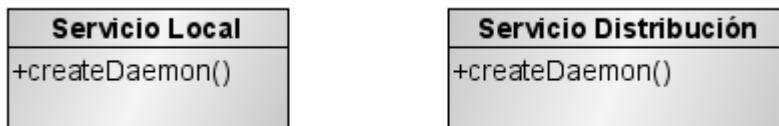


Figura 3.15 Diagrama de clases del diseño del paquete Servicios de los Servicios de Comunicación

A continuación se describen las clases del diseño para los servicios de comunicación de los paquetes Persistencia y Servicios, las del paquete Entidades presentan una descripción muy similar a las del paquete entidades de los servicios web.

Tabla 17: Descripción de la clase del diseño del paquete de Persistencia. ConexionCent

Nombre: ConexionCent	
Tipo de clase: Entidad	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Conect()

Descripción:	Permite la conexión con la base de datos a nivel central, retornando dicha conexión.
--------------	--

Tabla 18: Descripción de la clase del diseño del paquete de Persistencia. Conexión

Nombre: Conexion	
Tipo de clase: Entidad	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	Conect ()
Descripción:	Permite la conexión con la base de datos local, retornando dicha conexión.

Tabla 19: Descripción de la clase del diseño del paquete de Persistencia. LectorDB

Nombre: LectorDB	
Tipo de clase: Entidad	
Atributo	Tipo
con	Var
cursor	Var
Para cada responsabilidad:	
Nombre:	LeerTrans()
Descripción:	Lee de la base de datos los datos correspondientes a una transacción y retorna la transacción.
Nombre:	LeerTransRespuestas()
Descripción:	Lee de la base de datos los datos correspondientes a una transacción de tipo respuesta y retorna la transacción.
Nombre:	LeerGeneralizacion()
Descripción:	Lee de la base de datos los datos correspondientes a la tabla de la generalización una solicitud.
Nombre:	LeerGeneralizacionRespuestas()
Descripción:	Lee de la base de datos los datos correspondientes a la tabla de la generalización una respuesta.
Nombre:	LeerEspecializacion ()
Descripción:	Lee de la base de datos los datos correspondientes a la tabla de la especialización una transacción.
Nombre:	LeerImagenes()
Descripción:	Lee de la base de datos los datos correspondientes a la tabla de imágenes correspondientes a una transacción.

Nombre:	ExisteTrans()
Descripción:	Devuelve verdadero si existe una transacción en la base de datos
Nombre:	SolicitudesPendientesAEnviar()
Descripción:	Devuelve verdadero o falso si existen solicitudes en la base de datos pendientes por enviar.
Nombre:	RespuestasPendientesAEnviar()
Descripción:	Devuelve verdadero o falso si existen respuestas en la base de datos pendientes por enviar.

Tabla 20: Descripción de la clase del diseño del paquete de Persistencia. EscritorDB

Nombre: EscritorDB	
Tipo de clase: Entidad	
Atributo	Tipo
con	Var
cursor	Var
Para cada responsabilidad:	
Nombre:	EscribirTrans()
Descripción:	Escribe en la base de datos los datos correspondientes a una transacción.
Nombre:	EscribirGeneralizacion()
Descripción:	Escribe en la base de datos los datos correspondientes a la tabla de la generalización una solicitud.
Nombre:	LeerGeneralizacion()
Descripción:	Lee de la base de datos los datos correspondientes a la tabla de la generalización una solicitud.
Nombre:	EscribirEspecializacion()
Descripción:	Escribe en la base de datos los datos correspondientes a la tabla de la especialización una transacción.
Nombre:	EscribirImágenes()
Descripción:	Escribe en la base de datos los datos correspondientes a la tabla de imágenes correspondientes a una transacción.
Nombre:	LeerImágenes()
Descripción:	Lee de la base de datos los datos correspondientes a la tabla de imágenes correspondientes a una transacción.
Nombre:	ActualizarEstado()
Descripción:	Actualiza el estado de una transacción a estado enviada.
Nombre:	ExisteTrans()

Descripción:	Devuelve verdadero o falso si existe una transacción en la base de datos.
--------------	---

Tabla 21: Descripción de la clase del diseño del paquete de Persistencia. LectorXML

Nombre: LectorXML	
Tipo de clase: Entidad	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	leerXML()
Descripción:	Lee un XML con los datos de una transacción y retorna una transacción.

Tabla 22: Descripción de la clase del diseño del paquete de Persistencia. EscritorXML

Nombre: EscritorXML	
Tipo de clase: Entidad	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	EscribirTrans()
Descripción:	Escribe en un XML con los datos de una transacción y retorna el XML.

Tabla 23: Descripción de la clase del diseño del paquete de Persistencia. LectorIMAP

Nombre: LectorIMAP	
Tipo de clase: Entidad	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	read_mails_afis()
Descripción:	Se conecta al servidor de correos, mandando los mensajes correspondientes para obtener los correos del servidor.
Nombre:	get_direccion_remitente ()
Descripción:	Devuelve la dirección de correo electrónico del remitente.
Nombre:	get_direccion_destino ()
Descripción:	Devuelve la dirección de correo electrónico de destino.

Tabla 24: Descripción de la clase del diseño del paquete de Persistencia. EscritorSMTP

Nombre: EscritorSMTP	
Tipo de clase: Entidad	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	EscribirTran()
Descripción:	Escribe y envía los datos de una transacción hacia un servidor de correos.
Nombre:	send_mail()
Descripción:	Envía un correo hacia un servidor de correos.

Tabla 25: Descripción de la clase del diseño del paquete de Persistencia. ServicioLocal

Nombre: ServicioLocal	
Tipo de clase: Controladora	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	createDaemon()
Descripción:	Crea un servicio o demonio que va a estar corriendo como un proceso.
Nombre:	Hilo1()
Descripción:	Lee las solicitudes que tienen estado 1 (Pendiente) y las envía al centro de datos.
Nombre:	Hilo4()
Descripción:	Lee por IMAP las respuestas que han sido enviados al sistema local y las escribe en la BD.

Tabla 26: Descripción de la clase del diseño del paquete de Persistencia. ServicioCentral

Nombre: ServicioCentral	
Tipo de clase: Controladora	
Atributo	Tipo
-	-
Para cada responsabilidad:	
Nombre:	createDaemon()
Descripción:	Crea un servicio o demonio que va a estar corriendo como un proceso.
Nombre:	Hilo2()
Descripción:	Lee por IMAP las transacciones que han sido enviadas al centro de datos y las escribe

	en BD para un posterior envío hacia su destino.
Nombre:	Hilo3()
Descripción:	Lee las solicitudes que tienen estado 1 (Pendiente) y las envía al AFIS.
Nombre:	Hilo4()
Descripción:	Lee las respuestas y las envía al sistema cliente.

Se especifica el diagrama de secuencia del diseño para el caso de uso Enviar Solicitudes AFIS y Recibir Respuestas AFIS del servicio de comunicación.

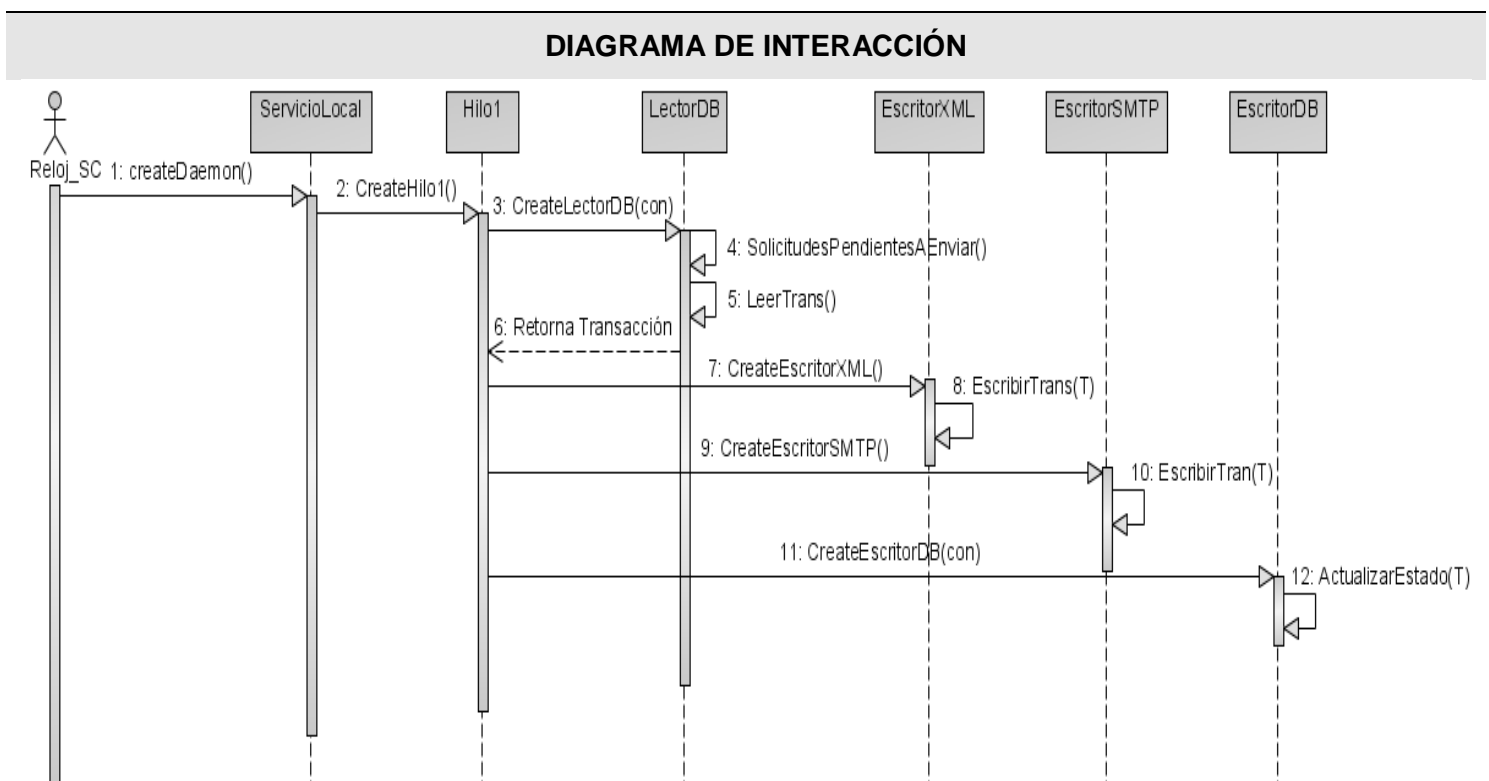


Figura 3.16: Diagrama de secuencia del diseño del caso de uso “Enviar Solicitudes AFIS”

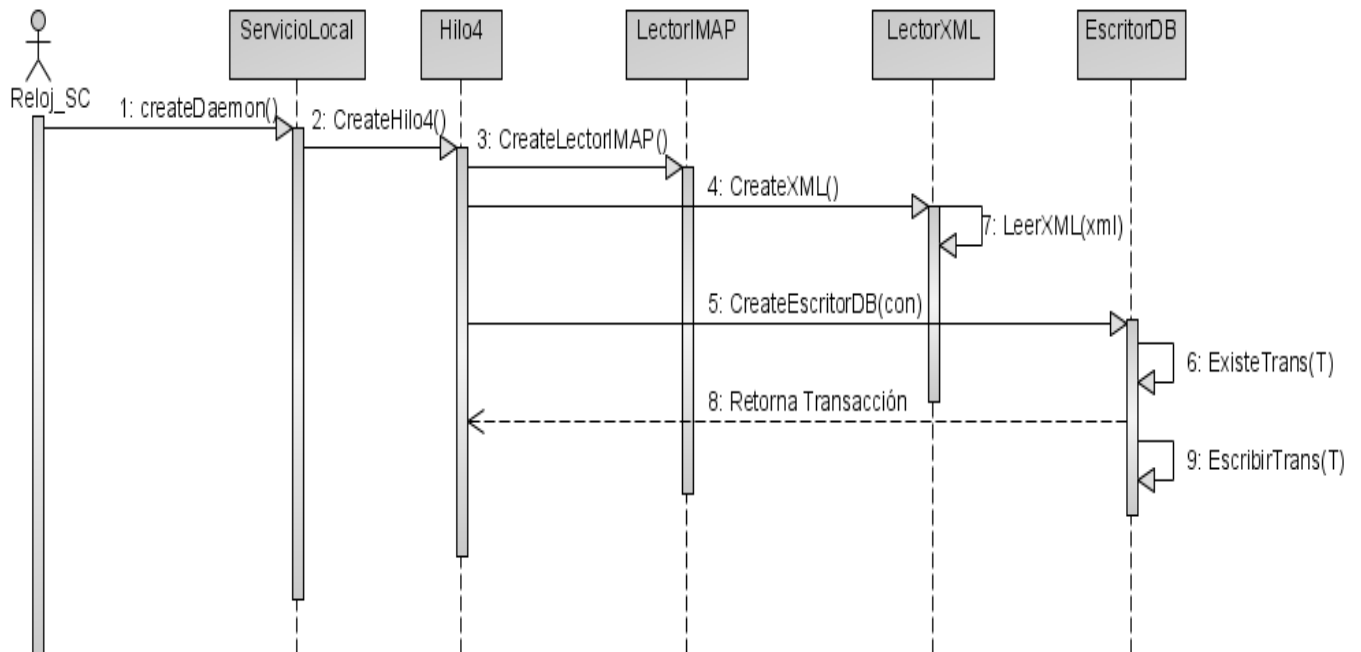


Figura 3.17: Diagrama de secuencia del diseño del caso de uso “Recibir Respuestas AFIS”

3.3.3. Diseño de la Base de Datos

Teniendo en cuenta las clases entidades obtenidas hasta el momento, se obtiene el modelo de base de datos a utilizar. La misma cuenta con 22 tablas donde persiste la información referente a las distintas transacciones que tienen lugar durante las funciones del servicio web, con el sistema cliente y el administrador, así como la de los servicios de comunicación.

Las tablas solicitudes y respuestas son las principales y se especializan debido a la existencia de diferentes tipos de ellas con sus características particulares. En general existe una estrecha relación entre todas las tablas de la base de datos, por la necesidad de obtener datos o escribir de una y otra para realizar cualquier acción.

En los Anexos III se puede ver la figura que visualiza el modelo entidad relación de forma general. A continuación se muestra este modelo dividido en tres partes con el objetivo de obtener una visión más clara de las tablas y sus relaciones. Se describen además las tablas principales del modelo, las demás descripciones verlas en los Anexos III.

DISEÑO DE LA BASE DE DATOS

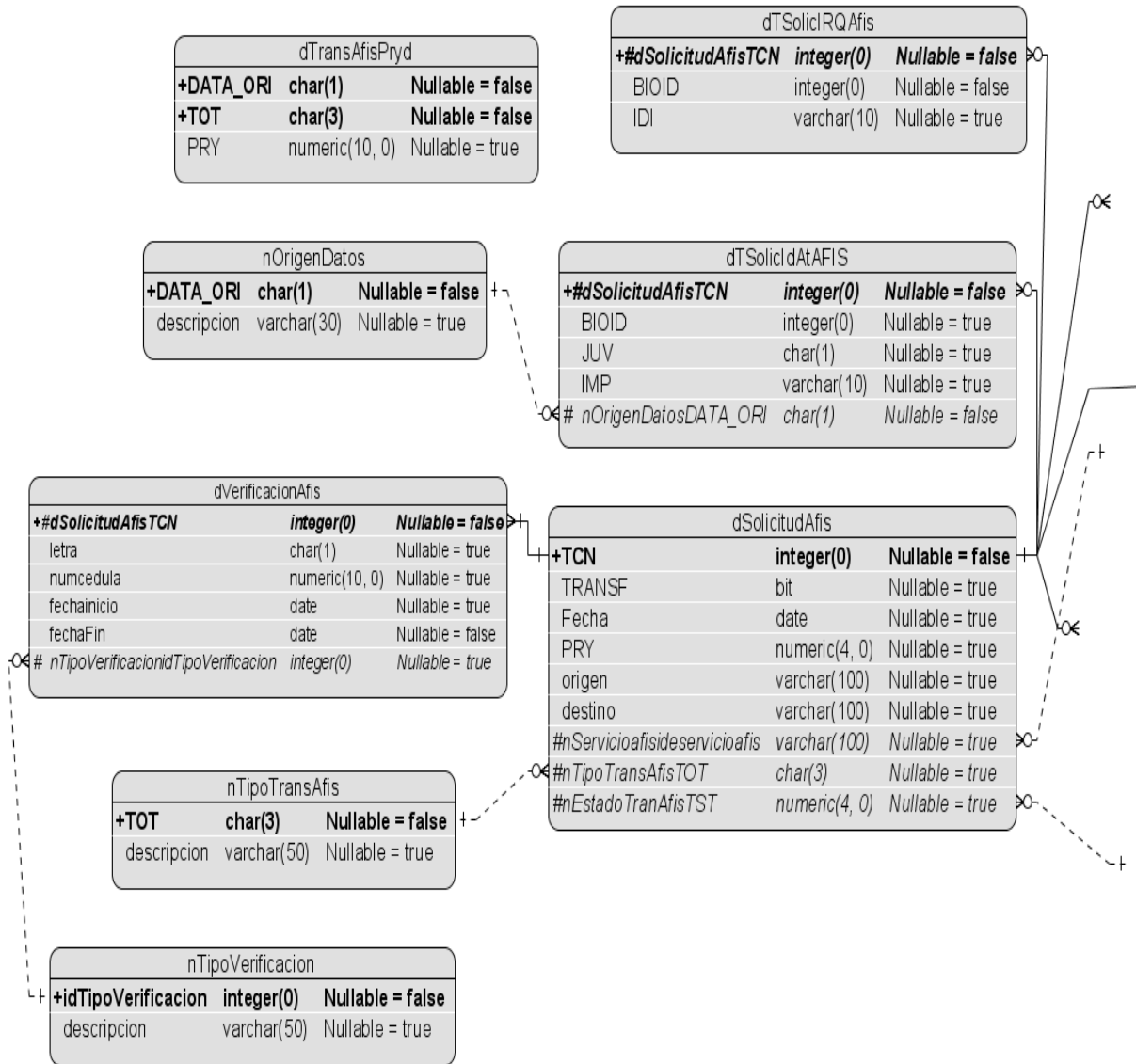


Figura 3.18: Diagrama de modelo de datos, (parte I)

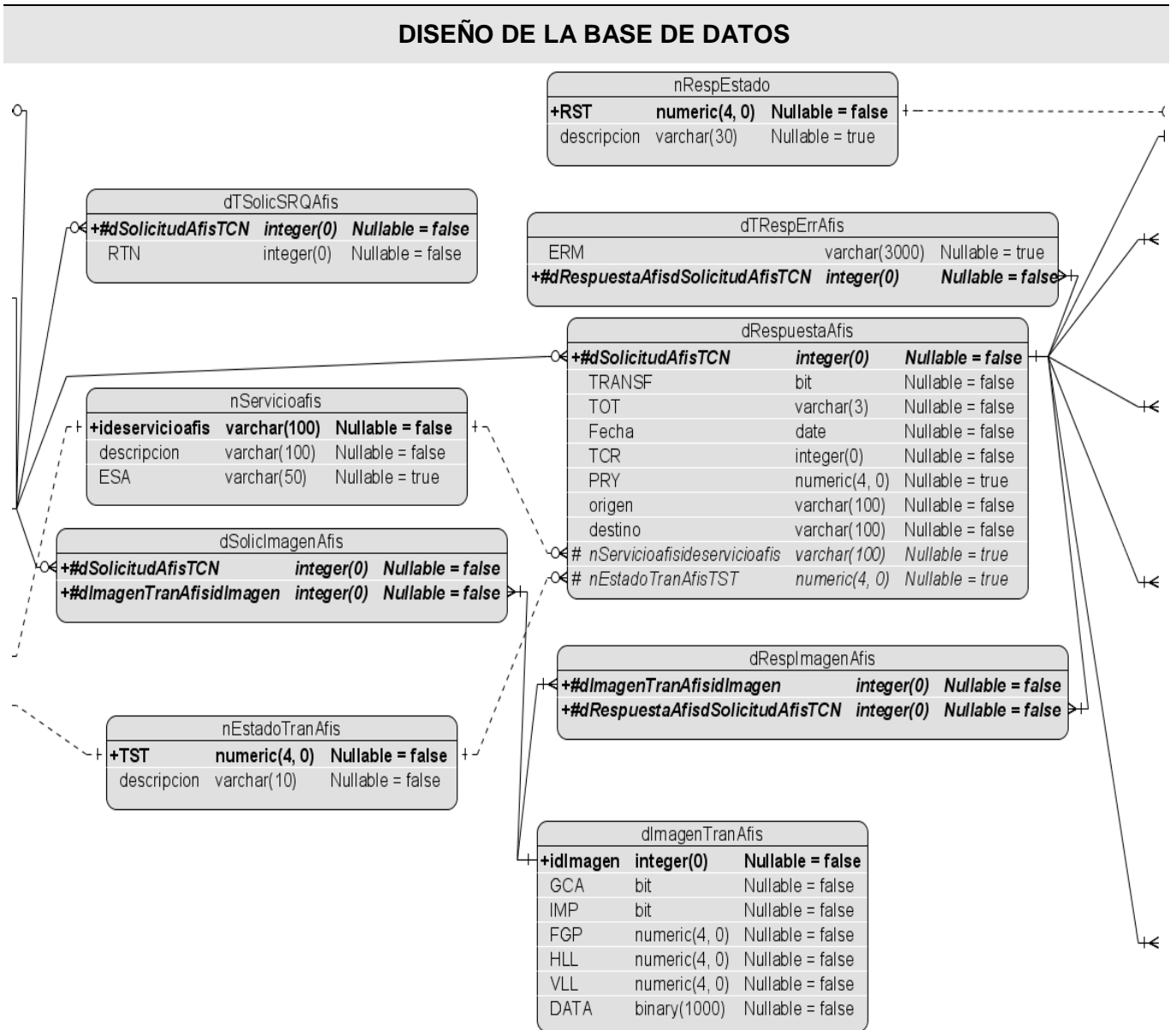


Figura 3.19: Diagrama de modelo de datos, (parte II)

DISEÑO DE LA BASE DE DATOS

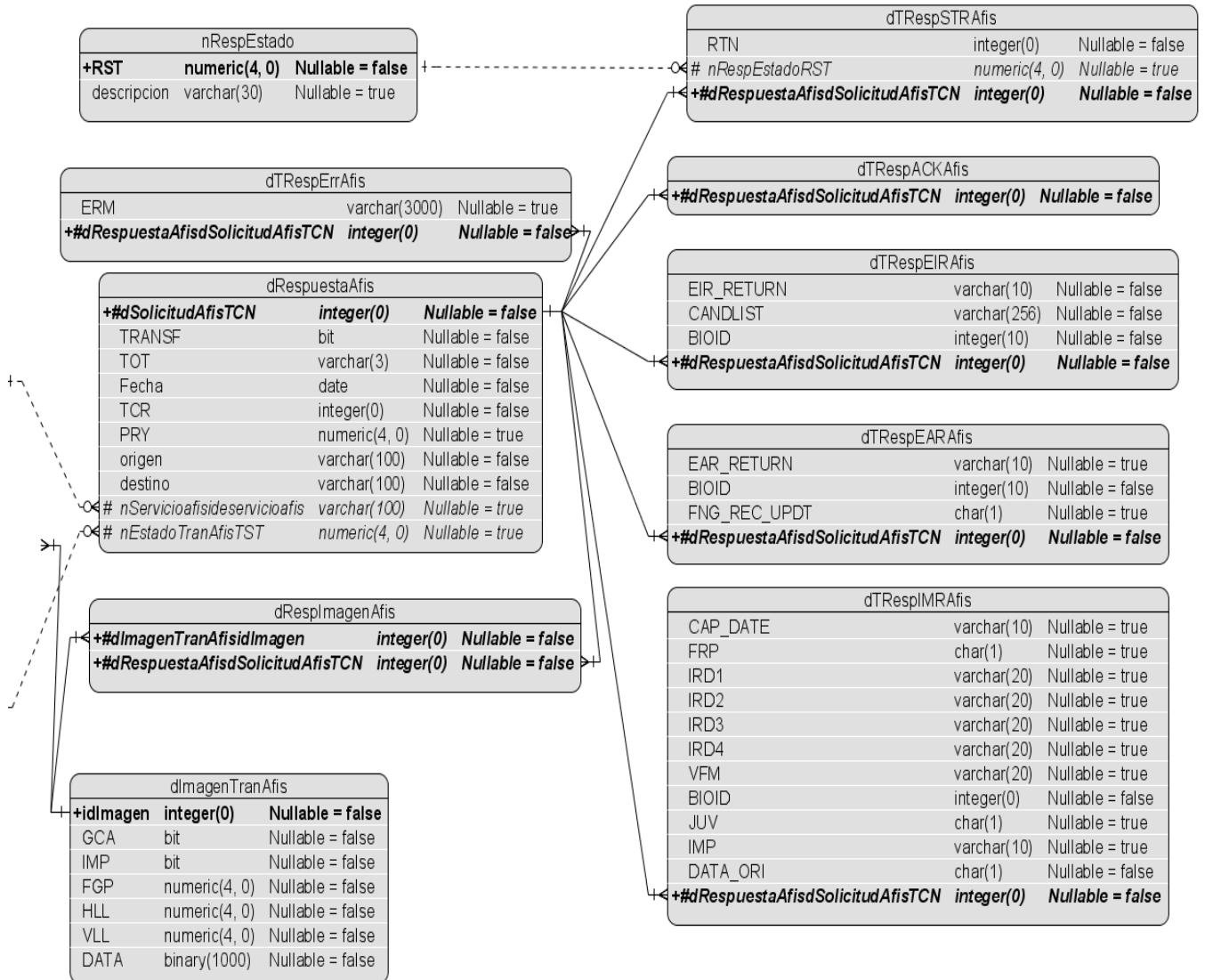


Figura 3.20 Diagrama de modelo de datos, (parte III)

3.3.3.1. Descripción de las tablas del modelo de datos

Tabla 27: Descripción de la tabla de nTipoVerificación del modelo entidad-relación

Nombre: nTipoVerificacion		
Descripción: Tabla que almacena los descripción de los tipos de verificaciones.		
Atributo	Tipo	Descripción
idTipoVerificacion	Integer(0)	Identificador de la tabla.

descripción	Varchar(50)	Identificación o Autenticación
-------------	-------------	--------------------------------

Tabla 28: Descripción de la tabla de dVerificaciónAfis del modelo entidad-relación

Nombre: dVerificacionAfis		
Descripción: Tabla que almacena		
Atributo	Tipo	Descripción
TCN	integer(0)	Identificador de la tabla.
letra	char(1)	Letra correspondiente a la cédula del ciudadano (V o E).
numcedula	numeric(10, 0)	Número correspondiente a la cédula del ciudadano, como máximo debe tener 10 dígitos.
fechainicio	date	Fecha en la que comienza la verificación.
fechaFin	date	Fecha en la que finaliza la verificación.
idTipoVerificacion	integer(0)	Tipo de verificación realizada (Identificación o Autenticación)

Tabla 29: Descripción de la tabla de dSolicitudAfis del modelo entidad-relación

Nombre: dSolicitudAfis		
Descripción: Tabla donde se almacena los datos correspondientes a una solicitud AFIS.		
Atributo	Tipo	Descripción
TCN	integer(0)	Identificador de la tabla.
TRANSF	bit	Especifica de donde fue transferida la transacción.
Fecha	date	Fecha en que se creó la solicitud.
PRY	numeric(4, 0)	Prioridad de la transacción, de 1-9.
origen	varchar(100)	Dirección de origen de donde una transacción fue enviada.
destino	varchar(100)	Dirección de destino a donde una transacción fue enviada.
ideservicioafis	varchar(100)	Identificador de la tabla nServicioAfis donde se almacena la descripción de los servicios.
TOT	char(3)	Identificador de la tabla nTipoTransAfis donde se almacena la descripción de los tipos de las transacciones AFIS.
TST	numeric(4, 0)	Identificador de la tabla nEstadoTransAfis donde

		se almacena la descripción de los estados de las transacciones AFIS.
--	--	--

Tabla 30: Descripción de la tabla de dRespuestaAfis del modelo entidad-relación

Nombre: dRespuestaAfis		
Descripción: Tabla donde se almacenan los datos correspondientes a una respuesta AFIS.		
Atributo	Tipo	Descripción
TCN	integer(0)	Identificador de la tabla.
TRANSF	bit	Especifica de donde fue transferida la transacción.
TOT	varchar(3)	Tipo de transacción.
Fecha	date	Fecha en que se realiza la respuesta.
TCR	integer(0)	Número de control de una transacción de respuesta.
PRY	numeric(4, 0)	Prioridad de la transacción.
origen	varchar(100)	Dirección de origen para la respuesta.
destino	varchar(100)	Dirección de destino a donde será enviada la respuesta.
TST	numeric(4, 0)	Identificador de la tabla nEstadoTranAfis donde se almacena la descripción del estado de la transacción AFIS.

Tabla 31: Descripción de la tabla de dImagenTranAfis del modelo entidad-relación

Nombre: dImagenTranAfis		
Descripción: Tabla que almacena los datos de las imágenes captadas para las transacciones AFIS.		
Atributo	Tipo	Descripción
idImagen	integer(0)	Identificador de la tabla.
GCA	bit	Algoritmo de compresión en escala de grises
IMP	bit	Tipo de impresión.
FGP	numeric(4, 0)	Posición del dedo
HLL	numeric(4, 0)	Longitud de línea horizontal
VLL	numeric(4, 0)	Longitud de línea vertical
DATA	binary(1000)	Información de la huella en base64

Tabla 32: Descripción de la tabla de nOrigenDatos del modelo entidad-relación

Nombre: nOrigenDatos		
Descripción: Tabla que almacena la descripción del origen de los datos.		
Atributo	Tipo	Descripción
DATA_ORI	char(1)	Identificador de la tabla.
descripcion	varchar(30)	Descripción de los datos de origen.

3.4. Conclusiones

Hasta aquí se han detallado más, con los diagramas de clases del diseño, los casos de usos del sistema. Permitiendo una mejor comprensión para el desarrollo de la implementación. Teniendo en cuenta el análisis previo de cada caso de uso.

El diseño de los servicios web quedó bien estructurado por paquetes que reunía clases según las funcionalidades que atendía, definiéndose así su arquitectura. Y se especificó a través de los diagramas de secuencia del diseño el flujo de acciones que se deben seguir en la implementación.

Los servicios de comunicación por su parte fueron diseñados según las cuatro acciones que lo definían, es decir un diagrama de diseño y el correspondiente diagrama de secuencia por cada caso de uso que los implicaba.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.1. Introducción

Corresponde saber ahora la distribución física que tendrá el software desde el punto de vista de los diagramas de despliegue y de componentes. Estos dos artefactos generados en el flujo de trabajo implementación de RUP, conforman un modelo de implementación donde se describen los componentes a construir, su organización y dependencia entre nodos físicos en los que funcionará el sistema. La descripción de estos modelos permitirán una mejor comprensión de cómo está implementado el software y su ubicación.

4.2. Diagrama de Despliegue

La siguiente figura muestra los siete nodos correspondientes a las aplicaciones del cliente y el administrador, los dos servicios web y base de datos respectivos, y un último nodo para la base de datos del AFIS. A continuación se explica cómo están relacionados.

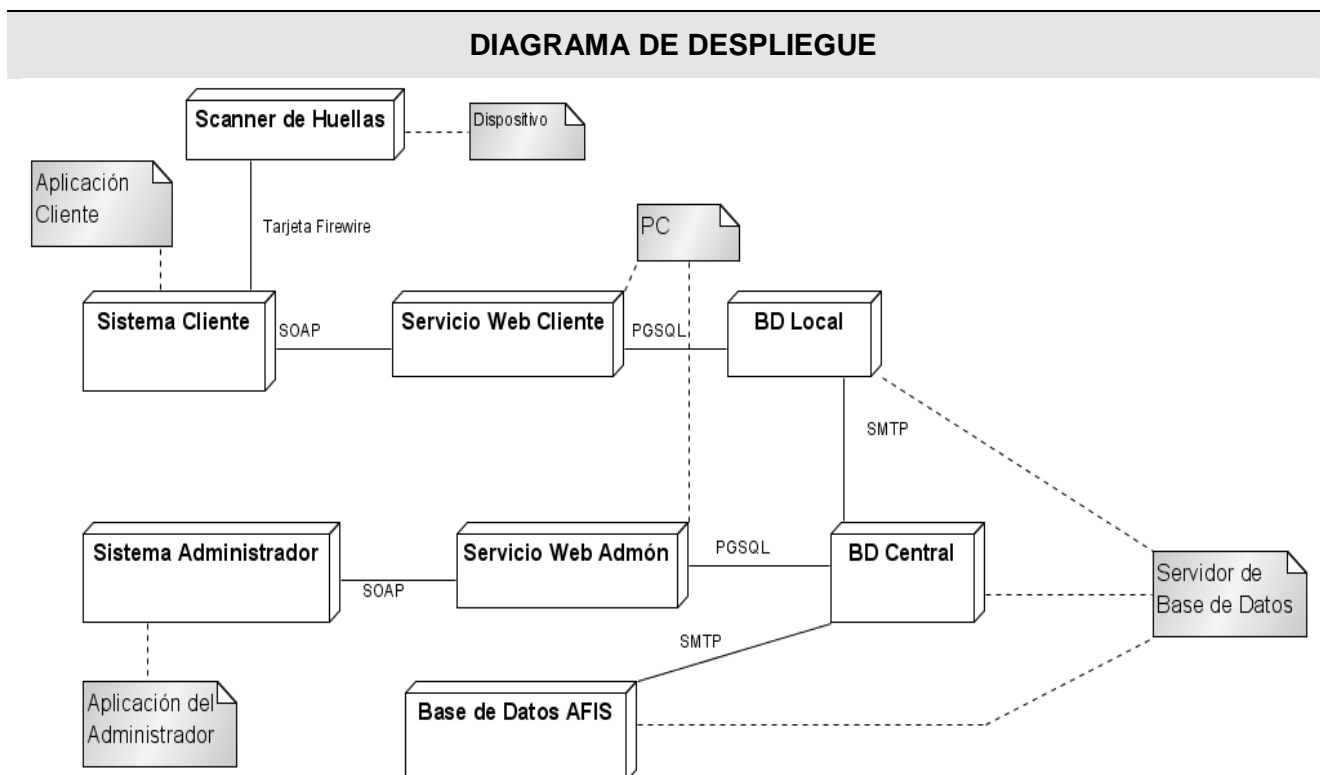


Figura 4.1: Diagrama de despliegue

4.2.1. Descripción del Diagrama de Despliegue

En la figura 4.1 se muestra la distribución física del sistema, obteniendo así el modelo de despliegue donde se representan a través de nodos los objetos que conforman la topología de hardware estrechamente relacionada con el software implementado, así como la relación de funcionamiento que existen entre dichos nodos.

El servicio de interfaz con el cliente será compatible y utilizado por cualquier sistema externo a SAIME y que requiera de las funcionalidades del AFIS, gracias a su publicación. El sistema cliente se conectará al servicio web a través del protocolo SOAP, para consumir las funcionalidades que este brinda. El servicio web utilizando la librería de acceso a datos PGSQL interactúa con el servidor local que se encuentra en el entorno de trabajo del cliente. En este servidor se almacenarán las verificaciones creadas por el servicio web y las respuestas enviadas por el servicio de distribución.

El servidor local se comunica con el centro de datos a través de mensajes de correo utilizando SMTP. De igual forma sucede entre la base de datos del centro de datos y la del AFIS, donde se procesan todas las transacciones.

En la base de datos central se escriben todas las transacciones que llegan al centro de datos. De este se leen los datos almacenados a través de la librería de abstracción a datos PGSQL, por el servicio web del administrador. A este último se conecta la aplicación de administración utilizando el protocolo SOAP.

4.3. Modelo de implementación

Se describen, con el modelo de implementación, cómo se implementan las clases relacionadas en el modelo de diseño. Además define la organización del sistema en términos de componentes y sus relaciones. Vale tener en cuenta que un componente es la parte modular de un sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces y proporciona la realización de los mismos. Un componente típicamente contiene clases y puede ser implementado por unos o más artefactos (ficheros ejecutables, binarios, etc).

4.3.1. Diagrama de Componentes

Los diagramas de componentes son usados para estructurar el modelo de implementación y mostrar las relaciones entre los elementos de implementación. Además representan la organización y

las dependencias lógicas entre un conjunto de componentes software, sean éstos, componentes de código fuente, librerías, binarios o ejecutables.

A continuación se muestra el diagrama de componentes dividido por paquetes del servicio web y los servicios de comunicación, así como los diagramas de componentes correspondientes a cada uno de los paquetes que se especifican.

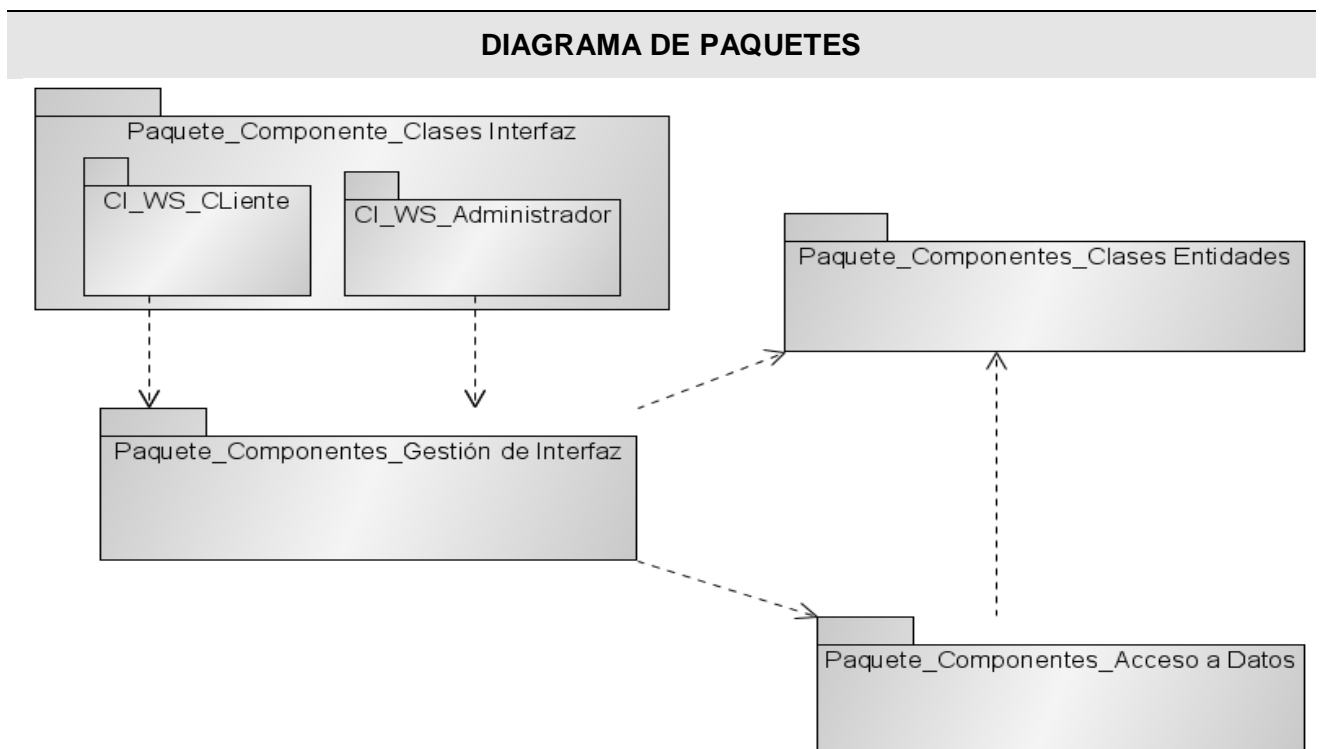


Figura 4.2: Diagrama de Paquetes de Componentes del servicio web

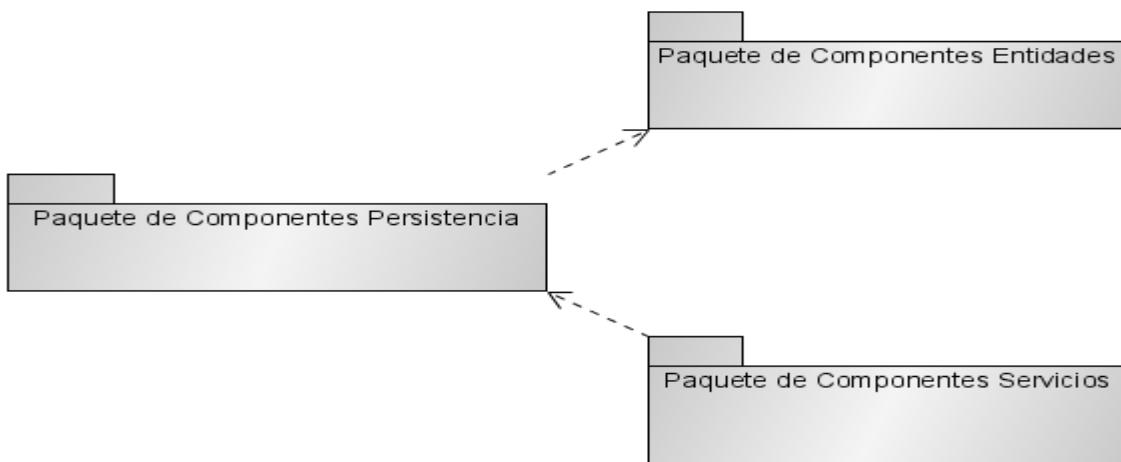


Figura 4.3 Diagrama de paquetes de componentes de los Servicios de Comunicación

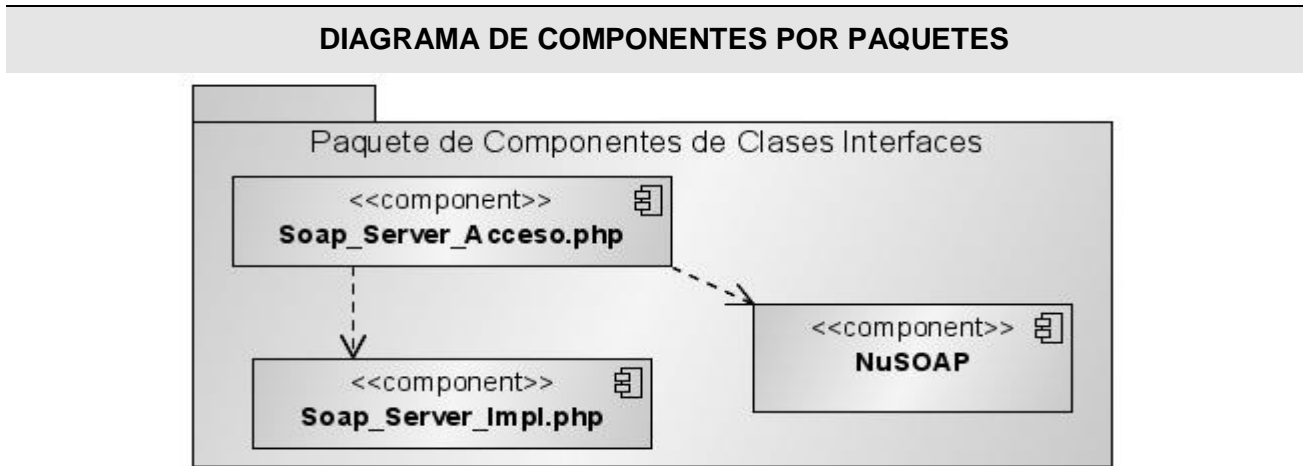


Figura 4.4: Diagrama de componentes para el paquete de clases interfaces de WS_Cliente

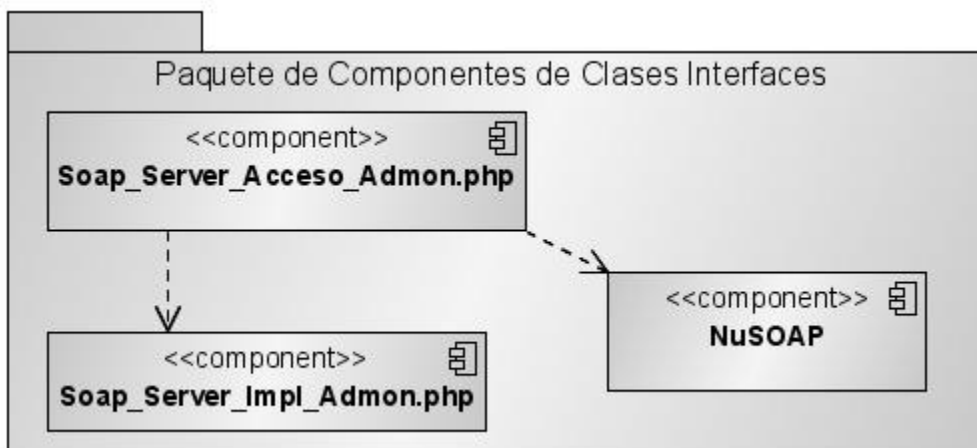


Figura 4.5: Diagrama de componentes para el paquete de clases interfaces de WS_Administrador

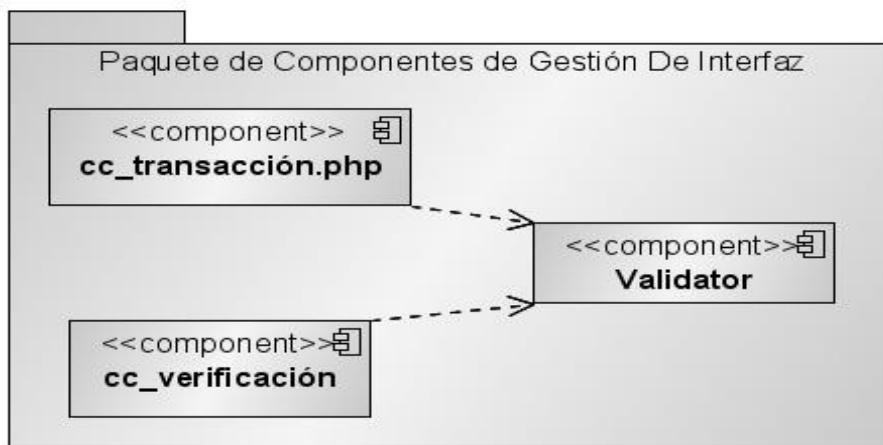


Figura 4.6: Diagrama de componentes para el paquete de gestión de interfaz (Servicio Web)

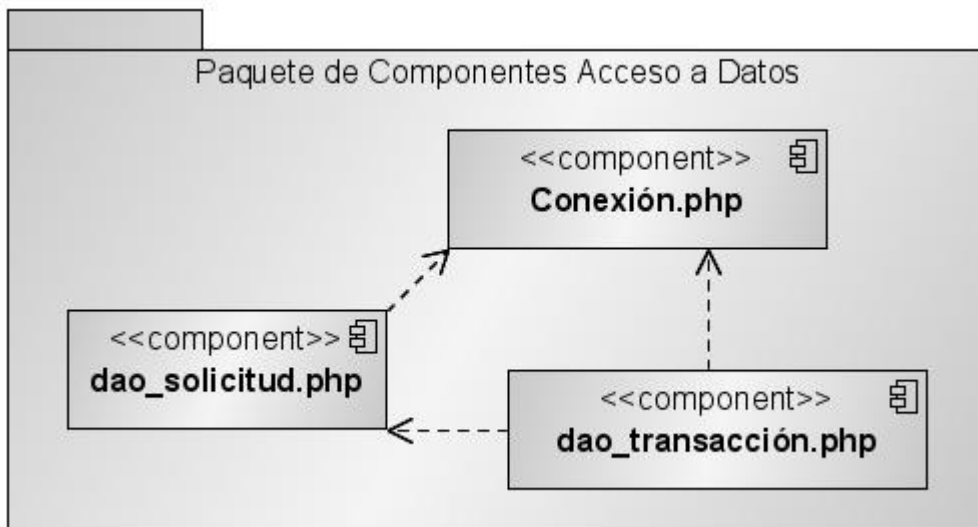


Figura 4.7: Diagrama de componentes para el paquete Acceso a Datos (Servicio Web)

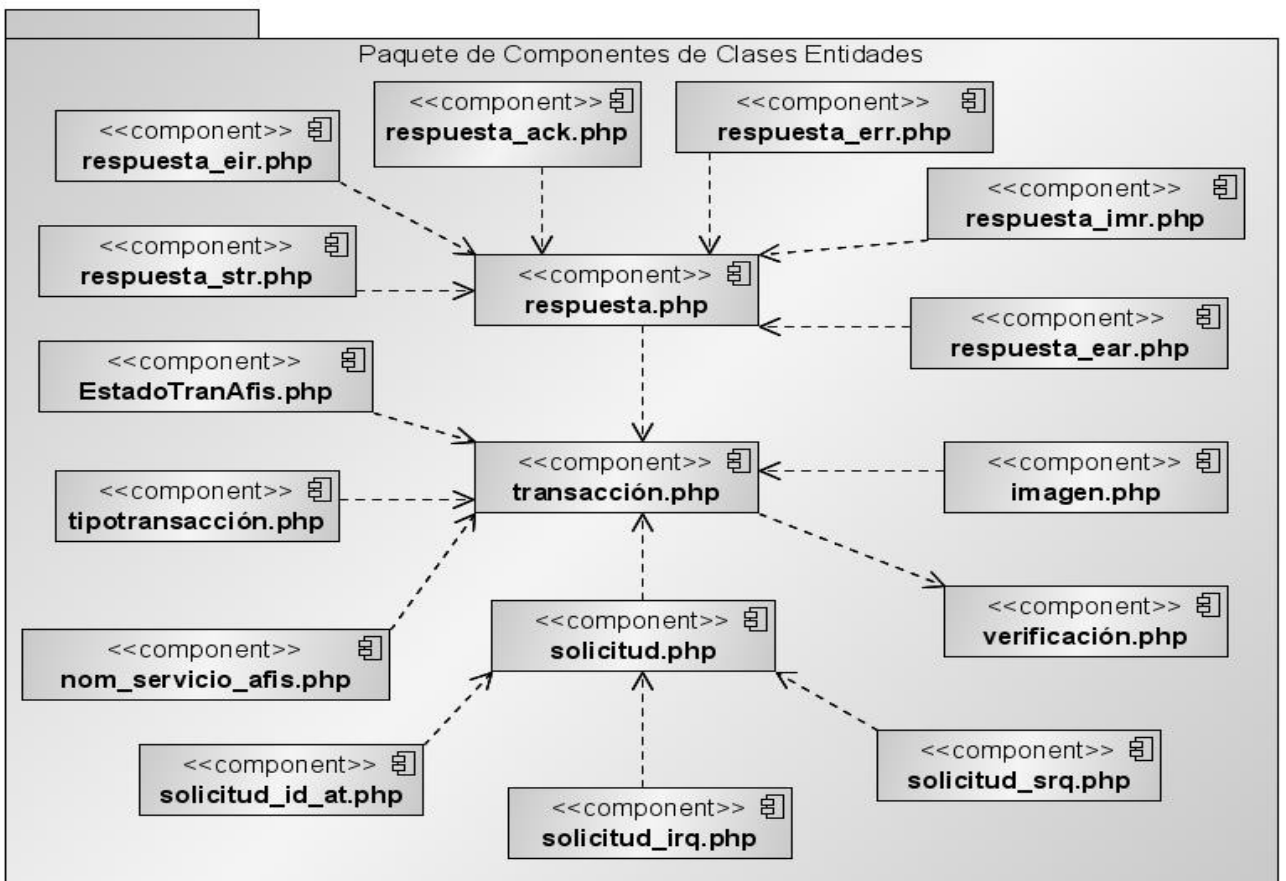


Figura 4.8: Diagrama de componentes para el paquete clases entidades (Servicio Web)

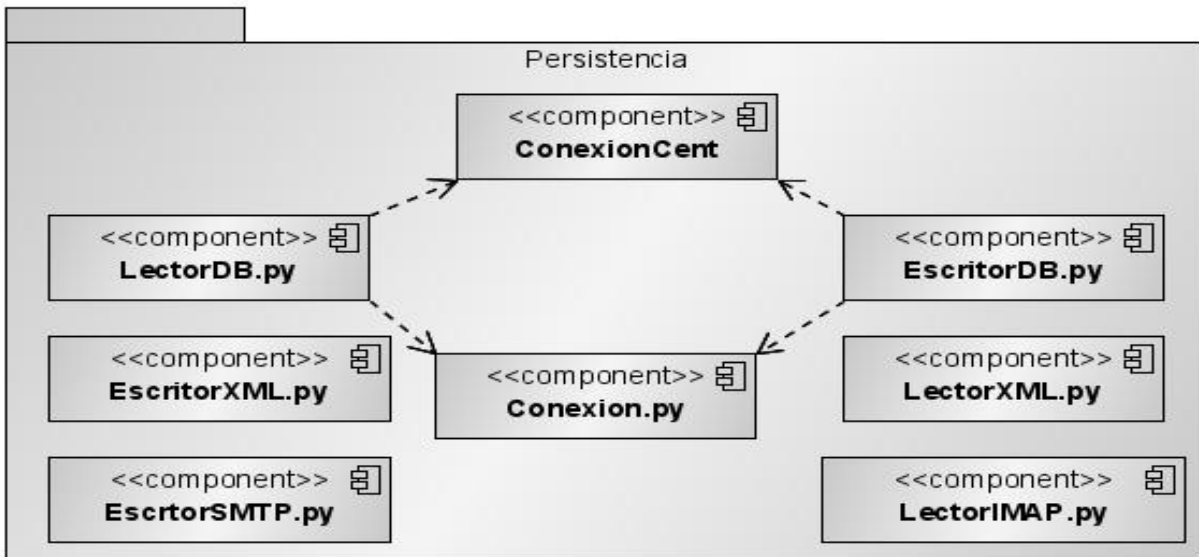


Figura 4.9 Diagrama de componentes para el paquete Persistencia de los Servicios de Comunicación

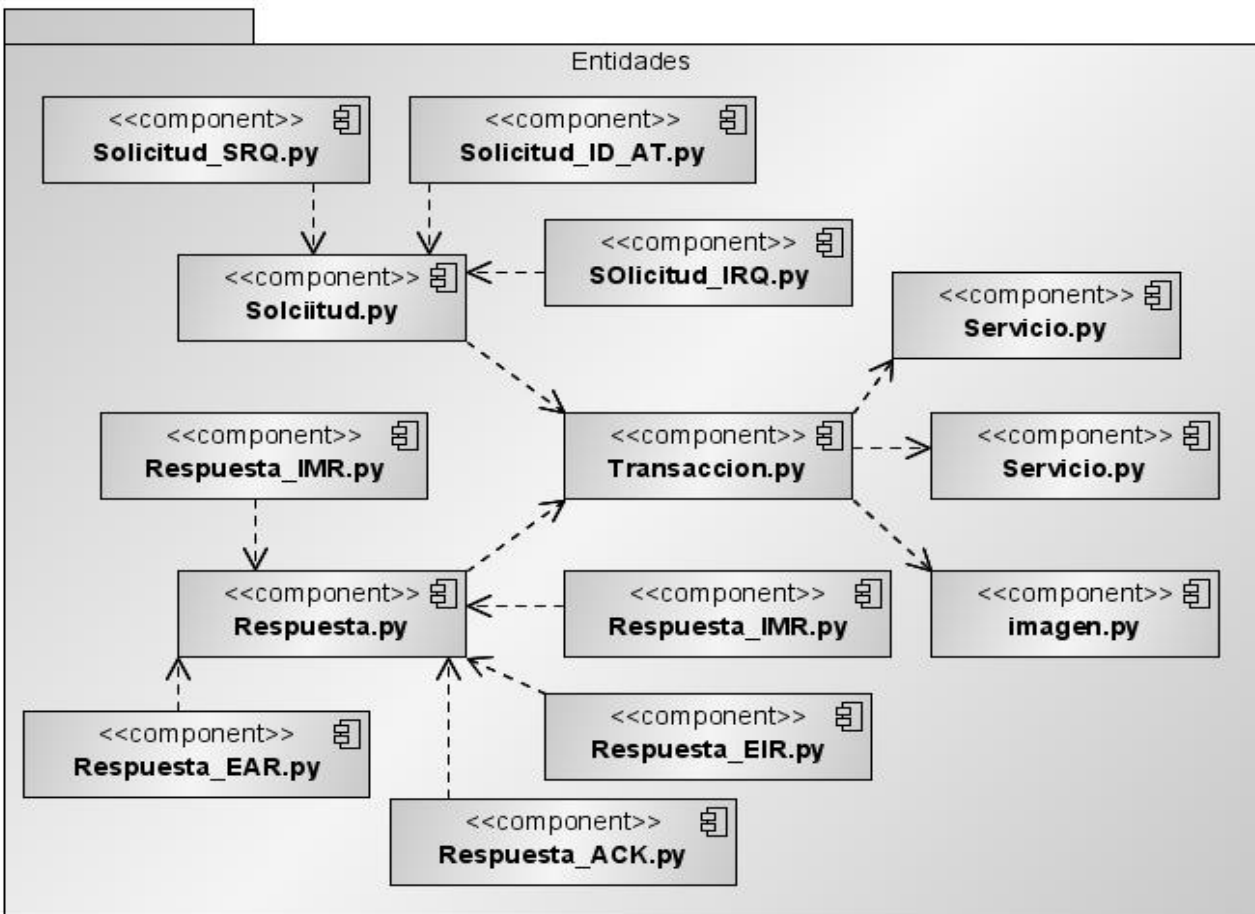


Figura 4.10 Diagrama de componentes para el paquete Transacciones de los Servicios de Comunicación

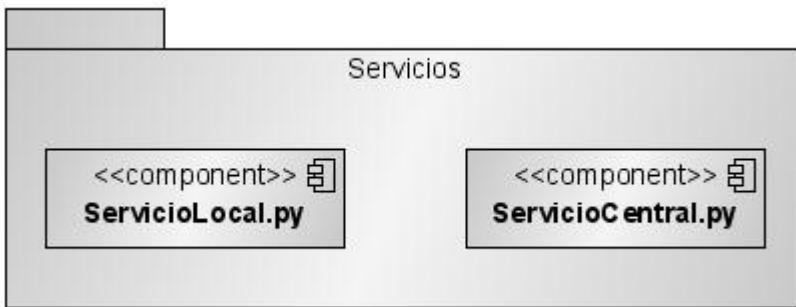


Figura 4.11 Diagrama de componentes para el paquete Servicios de los SC

4.3.2. Descripción de los paquetes de componentes

Paquete Componente Clases Interfaces: En este paquete se encuentran los componentes que contienen las implementaciones de las clases interfaz del servicio web para el cliente y el administrador. Se exponen las funcionalidades del mismo, así como la librería NuSOAP que posibilitó su creación.

Paquete de Componentes de clases Entidades: En este paquete se encuentran los componentes que contienen las implementaciones de las clases que representan las entidades del sistema.

Paquete Componentes de Gestión de Interfaz: En este paquete se encuentran los componentes que contienen las implementaciones de clases controladoras de entidades.

Paquete de Acceso a Datos: En este paquete se encuentran los componentes que contienen las implementaciones de clases que se ocupan de la persistencia en base de datos de las entidades. Así como los componentes que contienen las implementaciones de clases y librerías que permiten conectarnos a diferentes bases de datos.

Paquete de Persistencia: En este paquete se reúnen las componentes que corresponden a las implementaciones de las clases que se encargan del acceso a datos, así como de librerías que permitieron la conexión a la base de datos.

Paquete de Entidades: A este paquete corresponde los componentes que implementan las clases entidades, es decir, las pertenecientes al negocio.

Paquete de Servicios: Paquete donde se reúnen las implementaciones de las clases correspondientes a los servicios de comunicación y distribución.

4.1. Prueba

Para verificar el cumplimiento de los objetivos se realiza las pruebas de caja blanca y caja negra. Lo cual permite validar y darle un nivel de calidad al software. A continuación se especifican las pruebas realizadas de los dos tipos antes mencionados.

4.1.1. Prueba de Caja Negra

El método de pruebas de Caja Negra consiste en realizar pruebas sobre la interfaz del software, sin considerar la estructura interna del sistema. Verifican que las funciones del programa sean operativas y que se obtiene el resultado esperado. Para la realización de este método de prueba se construyó una aplicación web simulando una aplicación con las funcionalidades que el servicio web le daría al cliente y al administrador, y así comprobar la interacción entre el sistema y el servicio web.

Las siguientes descripciones pertenecen a los casos de prueba de integración, de los CU principales de los servicios. Los demás casos de pruebas verlos en los Anexos IV.

Tabla 33: Descripción de caso de prueba para el CU “Crear Verificación”

Entrada	Resultados	Condiciones
El sistema obtiene los datos necesarios para crear la verificación: (letra y número de cédula, tipo de solicitud, si es juvenil o no, y le capta las huellas al ciudadano).	El servicio escribe la verificación en BD Local y crea la solicitud según el tipo que se haya especificado y la pone en estado pendiente, para un posterior envío hacia el centro de datos.	Debe existir el BIOID de la persona en el centro de datos.

Tabla 34: Descripción de caso de prueba para el CU “Enviar Solicitud”

Entrada	Resultados	Condiciones
El servicio de comunicación (local) lee de BD Local las solicitudes listas para enviar. Conformando el mensaje de correo en formato MIME con un adjunto en formato XML que posee la información de la solicitud y la envía al servicio de distribución.	Se envía las solicitudes en forma de correo electrónico al servicio de distribución.	Debe existir alguna solicitud en la BD Local lista para enviar.

Tabla 35: Descripción de caso de prueba para el CU “Recibir Transacciones”

Entrada	Resultados	Condiciones
<p>El servicio distribución (central) establece una sesión IMAP con el servidor de correo para recibir el mensaje. Lo lee y lo decodifica, extrae el fichero adjunto en formato XML. Luego escribe en BD Central los datos de la transacción recibida teniendo en cuenta: (Dirección de origen: la dirección de origen del correo recibido, Dirección de destino: si la transacción es una solicitud que proviene del SCA de un cliente remoto tomar la dirección de correo del AFIS. Si la transacción es una respuesta que proviene del AFIS tomar la dirección de correos del origen de la solicitud correspondiente. (La del SCA que originó la solicitud)).</p>	<p>Se escribe en BD Central las transacciones recibidas.</p>	<p>Debe existir alguna transacción en el servidor de correo.</p>

4.1.2. Prueba de Caja Blanca

La aplicación de este método requiere del conocimiento de la estructura interna del programa y son derivadas a partir de las especificaciones internas del diseño o el código. Se comprueban los caminos lógicos del software dado un código específico.

Casos de prueba

Realización de prueba de caja blanca atendiendo al segmento de código correspondiente al caso de uso Crear Verificación.

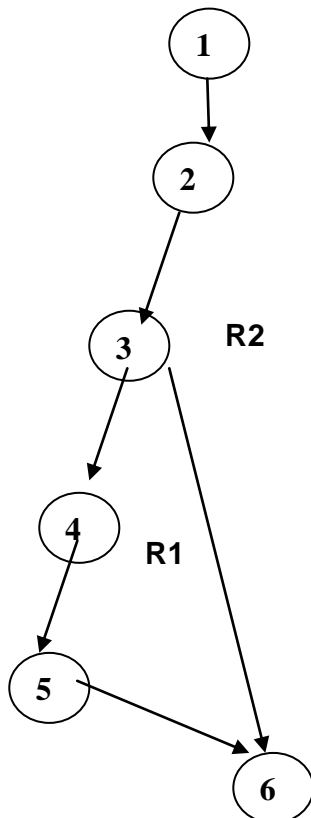
```

public function CrearVerificacion($Solic, $numcedula, $letra, $idtipoverficacion)
{
    $r=false; 1
    $error=$this->Reglas_Negocio_Verficacion($Solic, $numcedula, $letra); 2
    if(!$error[0]) 3
    {
        $dao_transaccion=new dao_transaccion(); 4
    }
}
    
```

```

    $r=$dao_transaccion->CrearVerificacion($Solic, $numcedula, $letra, $idtipoverficacion); 5
}
return $r;
} 6

```



Complejidad Ciclomática:

$V(G)$: Número de regiones del grafo

$$V(G) = A - N + 2$$

$$V(G) = P + 1$$

A : Número de aristas del grafo.

N : Número de nodos.

P : Número de nodos predicados.

$$V(G) = 2$$

Caminos: 1-2-3-4-5-6, 1-2-3-6

Figura 4.12 Código para el CU Crear Verificación del Servicio web

Camino: 1-2-3-6.

Caso de prueba: Crear Verificación para cuando los datos son incorrectos.

Entrada: -

Resultados: No se crea la verificación.

Condiciones: Que los datos de la verificación sean incorrectos.

Camino: 1-2-3-4-5-6

Caso de prueba: Crear Verificación cuando los datos introducidos son correctos.

Entrada: Se introducen los datos de la verificación.

Resultados: Se crea la verificación.

Condiciones: Que los datos introducidos sean correctos.

4.1. Conclusiones

El diagrama de despliegue compuesto por siete nodos, permitió conocer la ubicación física del sistema desde el punto de vista del hardware, teniendo así una mejor perspectiva de cómo será utilizado y el modo de conexión que se empleará.

Por su parte el diagrama de componentes representó el sistema desde el punto de vista de los componentes, dando mayor comprensión de cómo fue desarrollado el sistema. Finalmente las pruebas de caja blanca presentan la verificación y validación de las responsabilidades de cada servicio, dándole nivel de calidad a cada uno de ellos.

CONCLUSIONES

La implementación de servicios de comunicación permitió la transmisión de los datos biométricos dactilares de un ciudadano a través del envío y recepción de transacciones AFIS desde sistemas clientes hacia el AFIS Civil, con el objetivo de identificar o autenticar a una persona.

Los servicios web para el cliente y el administrador presentan las funcionalidades básicas necesarias para crear transacciones, conocer su estado y recibir una respuesta correspondiente, de esta forma los servicios web le permiten al usuario conocer si la solicitudes de identificación y autenticación fueron válidas o no, es decir aprueba si la persona es quién dice ser, o saber quién es la persona.

El servicio de comunicación y el de distribución AFIS en interacción con el servidor de correo postfix realizan las funciones requeridas para la conformación del mensaje de correo y envío del mismo al AFIS, así como para obtener de este una respectiva respuesta.

La utilización de PHP para el desarrollo de los servicios web implementados permite la interacción con cualquier tipo de aplicaciones, de tipo desktop o web desarrollado en cualquier lenguaje.

RECOMENDACIONES

Dada la complejidad propia de estos procesos de comunicación existen muchos trabajos posteriores que es aconsejable acometer para obtener una mejor infraestructura de servicios. Es de suma importancia el hecho de que la solución actual, establece el modelo de organización y desarrollo de la infraestructura básica de comunicación para que estas tareas puedan acometerse posteriormente. Aconsejamos investigar y desarrollar en los siguientes temas relacionados:

- Mejorar el control de flujo. Utilizar un mecanismo dinámico de asignación de cuotas basados en inteligencia artificial para que se prioricen las entidades según el tráfico y la cantidad de solicitudes pendientes.
- Implementar un sistema de permisos por entidad remota, por tipos de solicitudes, por horarios, etc., para el servicio central de distribución
- Implementar un sistema de actualización automático para los servicios y componentes remotos.
- Se recomienda cambiar el fichero XML conformado para el adjunto de correo, por un fichero NIST para la comunicación entre los servicios de comunicación, el servicio de distribución y el AFIS, fundamentalmente. Ya que los archivos NIST poseen como ventaja la capacidad de contener múltiples imágenes y otros tipos de datos sobre la persona, en un solo archivo, lo que simplifica el intercambio de datos biométricos entre las partes participantes.
- Profundizar en el tema del presente trabajo y extender los servicios web agregándole nuevas funcionalidades, según las necesidades que se presenten en el futuro.
- Además que la configuración del tiempo que utilizan los servicios de comunicación para el envío y recepción de las transacciones se realice fuera del código.

BIBLIOGRAFÍA CITADA

W Webmaster . 2008. Productos Linux Soluciones Seguras. [Online] 2008.

<http://www.enetchile.cl/productos/productoslinux/sercorreo.php>.

Alvarez, Miguel Angel. Desarrollo Web. [Online] <http://www.desarrolloweb.com/articulos/1852.php>.

Argentina, Ministerio del Interior de. 2007. Información general sobre biometría. [Online] 2007.

http://www.biometria.gov.ar/referencia/ref_overview_index.php.

Benedí, Jennifer Pérez. Web Services. [Online]

http://www.gvpontis.gva.es/fileadmin/conselleria/images/Documentacion/migracionSwAbierto/Anexos/Anexo_C9.pdf.

Caballero, Ismael. Prácticas Ingeniería del Software 3º Una Herramienta CASE para ADOO: Visual Paradigm. [Online] http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf.

Campus Libre y Abierto. 2006. Cala. [Online] 2006.

<http://campusvirtual.unex.es/cala/cala/mod/forum/discuss.php?d=486>.

Cervantes, Yurdik. 2007. *Especificaciones del subsistema de captura biométrica y comunicación con el AFIS.* 2007.

DesarrolloWeb. 2007. Lenguaje de programación. [Online] 2007.

<http://www.desarrolloweb.com/articulos/1325.php>.

HOMINI S.A. . 2004. Plataforma biométrica HOMINI. [Online] 2004.

http://www.homini.com/new_page_5.htm.

HOMINI, SA. 2004. Plataforma biométrica HOMINI. [Online] 2004.

<http://www.homini.com/pdf/Infor%20tecnica%20plataforma%20biometrica%20Homini.pdf>.

http Perú. 2007. http Perú. [Online] 2007. <http://www.http-peru.com/postgresql.php>.

Jaume Sabater. 2008. Configuración de un completo servidor de correo seguro con Postfix y Cyrus. [Online] 2008. http://bulma.net/body.phtml?nIdNoticia=2163#tec_postfix, Jaume Sabater.

NEO TEC S.A. 2007. NEOTEC. [Online] 2007. <http://www.neotec.com.pa/pdf/bioplanilla.pdf>.

Reisz, Carlos F. Identificación Dactiloscópica. [Online] <http://www.seguridad-la.com/artic/discipl/4112.htm>.

S.A. 2002. IAFIS Argentina. [Online] 2002. <http://www.iafis.net/iafis.htm>.

S.A, BIOMETRIKA. 2007. MegaMatcher SDK. [Online] 2007.

<http://www.biometrika.ec/megamacher.htm>.

Technologies, Biokey Identity. 2005. WSQ (Wavelet Scalar Quantization). [Online] 2005.

<http://www.biokey.cl/BokWsq.bok>.

Webmaster. Desarrollo Web. [Online] <http://www.desarrolloweb.com/articulos/1884.php>.

Webmaster 2008. Guía Breve de Servicios Web. [Online] 2008.

<http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.

Wikipedia. 2007. MIME (Multipurpose Internet Message Extensions). [Online] 2007.

<http://es.wikipedia.org/wiki/MIME>.

Wikipedia. 2008. RUP como metodología de desarrollo. [Online] 2008.

http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational.

Wikipedia. 2007. Biometría. [Online] 2007.

http://es.wikipedia.org/wiki/Biometr%C3%ADa#Tabla_comparativa_de_sistemas_biom.C3.A9tricos.

Wikipedia. 2008. Demonio informático. [Online] 2008.

[http://es.wikipedia.org/wiki/Demonio_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/Demonio_(inform%C3%A1tica)).

W3C. 2008. Guía Breve de Servicios Web. [Online] 2008.

<http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.

BIBLIOGRAFIA CONSULTADA

1. Universidad de las Ciencias Informáticas, Conferencias de Ingeniería de Software I, 2008.
2. Universidad de las Ciencias Informáticas, Conferencias de Ingeniería de Software II, 2008.
3. Rumbaugh, J.; Jacobson, I. y Booch, G.; “El Lenguaje Unificado de Modelado”. Manual de referencia. 2000. Addison-Wesley. Páginas 120-121, 157-162, 305-312.
4. Jain, A.; Bolle, R. y Pankanti, S; “BIOMETRICS Personal Identification in Networked Society”. Páginas 1-10, 43-50.
5. Sagem; “Especificaciones del sistema AFIS Civil”. Páginas 8-26, 40-55.
6. Jeffrey E.; “Advanced Studies in Biometrics”. Páginas 20-21
7. Jacobson, I.; Booch, G. y Rumbaugh, J.; “El Proceso Unificado de Desarrollo de software”. 2000. Addison-Wesley., Apéndice A. Visión General de UML, Apéndice B.
8. Jacobson, I.; Booch, G. y Rumbaugh, J.; “El Proceso Unificado de Desarrollo de software”. 2000. Addison-Wesley. Páginas 115-119.
9. Pressman, Roger; Ingeniería de software. Un enfoque práctico. 2002. McGraw.Hill/Interamericana de España.

GLOSARIO

ACK: Tipo de respuesta Acuse de Recibo.

AFIS: Sistema Automatizado de Identificación de Huellas Dactilares.

API: Interfaz de Programación de Aplicaciones.

ASCII: American Standard Code for Information Interchange o Código Estadounidense Estándar para el Intercambio de Información.

Asincrónica: Acción que se realiza en paralelo a la interacción con el usuario.

AUT: Tipo de Solicitud de autenticación.

BD: Base de Datos

BIOID: Identificador biométrico.

Biometría: Ciencia que se encarga de convertir características físicas en números.

CASE: Herramienta (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador).

CC: Clase controladora

CE: Clase Entidad

CI: Clase Interfaz

CUS: Caso de Uso del Sistema

Datos biométricos dactilares: Huellas.

EAR: Tipo de respuesta de autenticación.

EIR: Tipo de respuesta de identificación.

ERR: Tipo de respuesta Mensaje de error.

HTML: HyperText Markup Language, en español Lenguaje de Marcado de Hipertexto.

HTTP: Hypertext Transfer Protocol, en español Protocolo de de Transferencia de Hipertexto.

IDE: Integrated Development Environment o Entorno Integrado de Desarrollo

IDT: Tipo de Solicitud de identificación.

IMAP: Internet Message Access Protocol. Protocolo de red de acceso a mensajes electrónicos almacenados en un servidor.

IMP: Tipo de impresión de huellas.

IMR: Tipo de respuesta de imágenes.

IRQ: Tipo de Solicitud de imágenes.

JUV: Juvenil.

live-scan: Es un dispositivo de captura de huellas en vivo.

Middleware: Es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas.

MIME: Multipurpose Internet Mail Extensions en español Extensiones de Correo Internet Multipropósito.

MPPRIJ: Ministerio del Poder Popular para las Relaciones del Interior y Justicia

MS-DOS: MicroSoft Disk Operating System, Sistema operativo de disco de Microsoft.

MTA: Mail Transfer Agent o Agente de Transporte de Correo, constituye la base de un servidor de correo.

NIST: National Institute of Standards and Technology en español Instituto Nacional de Estándares y Tecnología.

NuSOAP: Es un kit de herramientas (ToolKit) para desarrollar Web Services bajo el lenguaje PHP.

OMG: Object Management Group.

ONIDEX: Oficina Nacional de Identificación y Extranjería.

OR: Oficina Regional

PDA's: Asistente Digital Personal en español Personal Digital Assistant. Computador de mano.

PGSQL: Librería de abstracción a datos con postgresQL.

PHP: Acrónimo de Hypertext Preprocessor (Preprocesador de Hipertexto).

POP3, POP: Post Office Protocol. Es un servidor-cliente, un protocolo para obtener los mensajes de correo electrónico almacenados en un servidor remoto

Python: Lenguaje de programación, de scripting independiente de plataforma y orientado a objetos.

'plugins': Componente enchufable. Aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica.

RAM: Random Access Memory - Memoria de acceso aleatorio.

Respuestas AFIS: Mensajes que se reciben del AFIS Civil como respuesta a una funcionalidad solicitada por el cliente mediante una Solicitud AFIS.

RF: Requisito Funcional

RTN: Request Transaction control Number. Número de control (TCN) de la transacción cuyo estado se quiere recuperar.

RUP: Rational Unified Process, en español Proceso Unificado Racional.

RUP: Rational Unified Process, en español Proceso Unificado Racional.

SAIME: Servicio Autónomo de Identificación Migración y Extranjería.

SC: Servicio de Comunicación.

SCA: Servicio Central AFIS o Servicio de Comunicación AFIS.

SDA: Servicio de Distribución AFIS.

Servicios o Demonios: Proceso informático que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario.

Servicios web: Colección de protocolos y estándares que sirve para intercambiar datos entre aplicaciones.

SGML: Standard Generalized Markup Language, en español Lenguaje de Señalización General Normalizado.

SLA: Servicio Local AFIS

SMTP: Simple Mail Transfer Protocol. Protocolo Simple de Transferencia de Correo.

SOAP: Simple Object Access Protocol, en español protocolo simple de acceso a objetos.

Solicitudes AFIS: Mensajes que se le puede enviar al AFIS Civil para solicitarle una determinada funcionalidad.

SRQ: Tipo de Solicitud de estado.

TCN: Transaction Control Number, en español Número de control de la transacción.

TCP/IP: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP).

TCR: Transaction Control Response, es español Número de control de la transacción de respuesta.

TOT: Type Of Transacción, en español tipo de transacción.

Transacciones AFIS: Se refiere indistintamente a una solicitud AFIS o respuesta AFIS.

UML: Unified Modeling Language, en español Lenguaje Unificado de Modelado.

US-ASCII: Término ASCII

Verificaciones: Solicitud que realiza el cliente para crear una solicitud de tipo IDI o ATU.

WS: Web Service o Servicio Web.

WSDL: Web Services Description Language O Lenguaje de Descripción del Servicio Web.

WSQ: Wavelet Scalar Quantization. Estándar para almacenar y transportar impresiones dactilares por medios electrónicos. Formato.

XML: Extensible Markup Language en español Lenguaje de Marcaje Extensible.

ANEXOS I

Tabla 36: Operacionalización de las Variables

Variable Conceptual	Dimensión	Indicadores	Unidad de medida
<i>Servicios de comunicación.</i>	Factibilidad	Tiempo de desarrollo	▪ Extenso
			▪ Moderado
			▪ Breve
		Costo	▪ Costoso
			▪ Moderado
			▪ Barato
		Esfuerzo	▪ Alto
			▪ Moderado
			▪ Despreciable
<i>Envío de solicitudes de identificación y autenticación desde cualquier sistema cliente hacia el AFIS Civil</i>	Mejor rendimiento	Complejidad	▪ Alta
			▪ Media
			▪ Baja
		Control	▪ Bueno
			▪ Deficiente
		Organización del trabajo	▪ Bueno
			▪ Deficiente
		Importancia	▪ Alta
			▪ Media
			▪ Baja
		Tiempo de ejecución	▪ Rápido
			▪ Moderado
▪ Lento			

Tabla 37: Recursos humanos

Nombre del Recurso	Rol	Fecha entrada al proyecto	Fecha salida del proyecto	Horas mensuales en el proyecto	Salario Básico Mensual
Yilian Rodríguez Grille	Líder de proyecto	Nov/2007	Jun/2008	100h	100 MN
Yunierki Verdecia Viltres	Líder de proyecto	Nov/2007	Jun/2008	100h	150 MN
Yurdik Cervantes Mendoza	Colaborador o tutor	Nov/2007	Jun/2008	24h	400 MN

Tabla 38: Recursos materiales

Recursos	Cantidad	Costo en CUC	Costo Total	Fuente de Financiamiento
Material gastable				
Materiales de oficina				
• Paquete de Hojas	3	5.65	16.95	Institución(UCI)
• Tóner	1	25.00	25.00	
Equipamiento necesario				
Equipos de computación				
• Computadora	1	700.00	700.00	Institución(UCI)
• Impresora	1	300.00	300.00	
• Scanner de huella	1	500.00	500.00	
Total			1541,25	

ANEXOS II

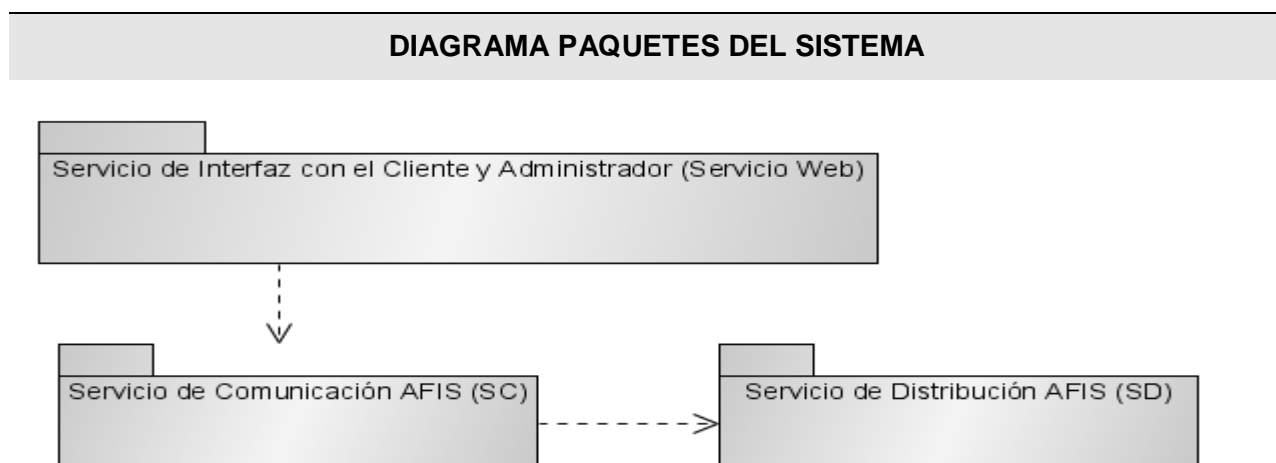


Figura II.1 Diagrama de paquetes del sistema

Tabla 39: Descripción del CUS “Mostrar Listado de Verificaciones”

Caso de Uso:	Mostrar Listado de Verificaciones
Actores:	Sistema Usuario
Propósito:	Conocer las verificaciones hechas hasta el momento.
Resumen:	Se obtienen el listado de las verificaciones hechas y los datos correspondientes.
Precondiciones:	El sistema cliente ha creado al menos una verificación.
Referencias	RF 6 (6.1, 6.2)
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El sistema cliente se conecta al servicio web para obtener el listado de verificaciones hechas hasta el momento.	
	2. El servicio web gestiona el listado de verificaciones, obteniendo de la base de datos local los datos correspondientes.
	3. El servicio web devuelve al sistema cliente el listado de verificaciones atendiendo a los siguientes datos: 3.1. TCN. 3.2. Letra y número de cédula.

	<p>3.3. Fecha de inicio, para cuando fue creada la verificación y fin cuando obtiene la respuesta final.</p> <p>3.4. Descripción.</p> <p>3.5. Hit o No Hit.</p>
Poscondiciones	Se mostró el listado de verificaciones.

Tabla 40: Descripción del CUS “Generar Listado de Transacciones”

Caso de Uso:	Mostrar Listado de Transacciones	
Actores:	Sistema Usuario	
Propósito:	Conocer las transacciones hechas así como su estado.	
Resumen:	Se obtienen los listados de todas las transacciones hechas y los datos correspondientes, como el estado en que se encuentran permitiendo así la supervisión de datos de las solicitudes y respuestas.	
Precondiciones:	El sistema usuario requiere mostrar listado de transacciones. Debe existir alguna transacción hecha.	
Referencias	RF 7 (7.1, 7.2, 7.3)	
Prioridad	Secundario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El sistema usuario se conecta al servicio web.		
	2. El servicio web gestiona el listado de transacciones, obteniendo de la base de datos local o central los datos correspondientes.	
	<p>3. El servicio web devuelve al sistema usuario el listado de transacciones según petición hecha.</p> <p>3.2. Listado de solicitudes.</p> <p>3.2.1. TCN.</p> <p>3.2.2. TOT, tipo de solicitud.</p> <p>3.2.3. Estado de la solicitud.</p> <p>3.2.4. Fecha en que se realizó.</p> <p>3.2.5. Prioridad.</p> <p>3.2.6. Transferencia.</p> <p>3.2.7. Origen y destino.</p>	

	3.3. Listado de respuestas. 3.3.1. TCN, TOT. 3.3.2. Estado de respuesta. 3.3.3. Fecha y prioridad. 3.3.4. Origen y destino.
Poscondiciones	Se mostraron los listados necesarios.

Descripción de CU para Servicio de Interfaz con el Administrador (Servicio Web)

Tabla 41: Descripción del CUS “Generar Reportes de Solicitudes”

Caso de Uso:	Generar Reportes de Solicitudes.
Actores:	Sistema Administración
Propósito:	Controlar cantidad de transacciones.
Resumen:	El caso de uso recoge las funciones correspondientes al control de las cantidades de solicitudes que se procesan correctamente de las aplicaciones externas, cantidad de solicitudes con error por cliente, cantidad de solicitudes por tipo y por cliente, tiempo de demora promedio por solicitud.
Precondiciones:	Existen solicitudes correctas, con error, por tipo, por cliente.
Referencias	RF 9 (9.1, 9.2, 9.3, 9.4)
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El sistema administración se conecta al servicio web para obtener el reporte de solicitudes.	
	2. El sistema genera el reporte de solicitudes procesadas por el servicio central teniendo en cuenta: <ul style="list-style-type: none"> 2.1. Cantidad de solicitudes procesadas correctamente de las aplicaciones externas. 2.2. Cantidad de solicitudes con error por cliente. 2.3. Cantidad de solicitudes por tipo y por cliente. 2.4. Tiempo de demora promedio por solicitud.

Poscondiciones	Se hizo efectivo el paso de generar reportes de solicitud.
-----------------------	--

Tabla 42: Descripción CUS “Solicitar Envío de Solicitud SRQ”

Caso de Uso:	Solicitar Envío de Solicitud SRQ	
Actores:	Sistema Usuario	
Propósito:	Solicitar enviar una solicitud SRQ.	
Resumen:	El sistema usuario requiere conocer en que estado se encuentra una solicitud determinada y solicita enviar una solicitud de estado (SRQ).	
Precondiciones:	Existe demora en la respuesta de una solicitud.	
Referencias	RF 8	
Prioridad	Secundario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El sistema usuario obtiene los datos correspondientes a la solicitud de estado y se conecta al servicio web para solicitar el envío de dicha solicitud para alguna solicitud ya enviada.		
	2. El servicio web obtiene los datos de la solicitud de estado 2.1.	
	3. El servicio escribe en la BD local la solicitud de estado y devuelve mensaje informándolo al sistema cliente. Si no fue creada la solicitud de estado ver flujos alternos.	
Flujos Alternos		
Acción del actor	Respuesta del Sistema	
	3.1. El servicio devuelve mensaje informando que no se creó satisfactoriamente la solicitud de estado.	
Poscondiciones	Se hizo efectivo el paso de solicitud de envío de solicitud SRQ.	

Tabla 43: Descripción del CUS “Recibir Respuestas AFIS”

Caso de Uso:	Recibir Respuestas AFIS.
Actores:	Reloj_SC
Propósito:	Recibir y escribir en BD Local las respuestas AFIS.

Resumen:	El Reloj_SC revisa cada un determinado tiempo si existen nuevos mensajes enviados por el servicio de distribución. Si es detectado entonces se reciben y se escriben en BD Local.
Precondiciones:	El servicio de distribución envió una respuesta AFIS que le llegó anteriormente desde el AFIS.
Referencias	RF 4 (4.1, 4.2, 4.3, 4.4)
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El reloj_SC chequea y detecta la existencia de un nuevo mensaje enviado por el servicio de distribución.	
	2. El servicio de comunicación establece una sesión IMAP con el servidor de correos con los comandos necesarios para recibir el mensaje.
	3. El servicio recibe el mensaje y lo decodifica. 3.1. Extrae el fichero adjunto. 3.2. Del fichero con formato NIST obtenido se extraen los datos de la solicitud.
	4. Escribe en BD Local los datos correspondientes a la respuesta recibida.
Poscondiciones	Se escribió en base de datos local la respuesta AFIS obtenida.

Tabla 44: Descripción del CUS “Enviar Transacciones”

Caso de Uso:	Enviar Transacciones.
Actores:	Reloj_SD
Propósito:	Enviar solicitudes al sistema AFIS y respuestas al servicio de comunicación AFIS.
Resumen:	Periódicamente, cada determinado quantum de tiempo el reloj_SD realiza una revisión en la BD Central hasta detectar una nueva transacción. Luego se envían las solicitudes al sistema AFIS y las respuestas al servicio de comunicación AFIS.
Precondiciones:	El servicio de comunicación ha enviado una solicitud. El sistema AFIS ha procesado una solicitud y enviado una respuesta de la misma.

Referencias	RF 2 (2.1, 2.2, 2.3, 2.4, 2.5)
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El reloj_SD chequea y detecta la existencia de una nueva transacción en la BD Central, teniendo en cuenta su prioridad.	
	2. El servicio obtiene los datos de la transacción escrita en la BD Central. 2.1.
	3. El servicio conforma el fichero NIST con los datos correspondientes a la transacción.
	4. El servicio codifica el mensaje electrónico MIME, con el fichero NIST adjunto.
	5. El servicio establece una sesión SMTP con el servidor de correos.
	6. El servicio envía el mensaje de correo correspondiente a la transacción, al AFIS o al servicio de comunicación.
Poscondiciones	Se envió la transacción al AFIS o al servicio de comunicación AFIS.

Anexos III

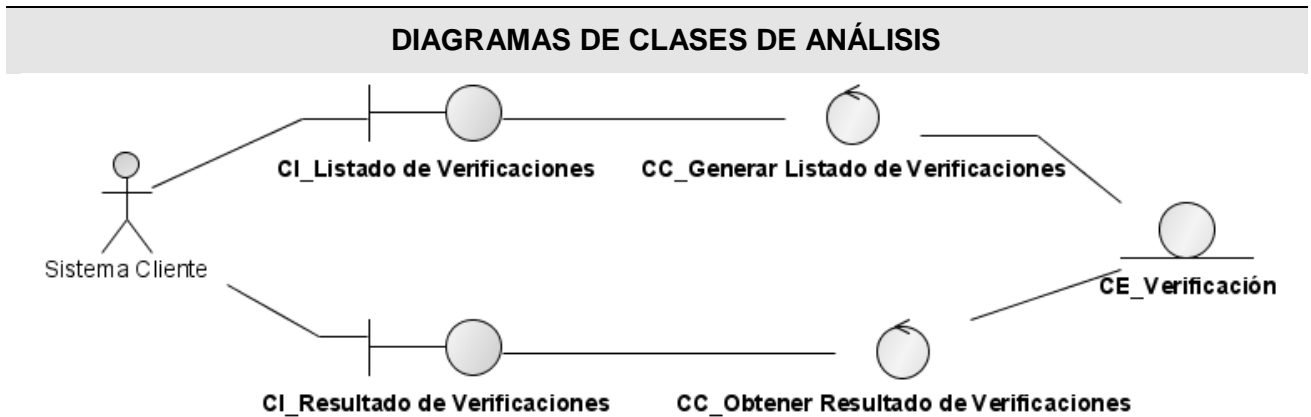


Figura III.1 Diagrama de clases del análisis para el sub-paquete servicio web con el cliente

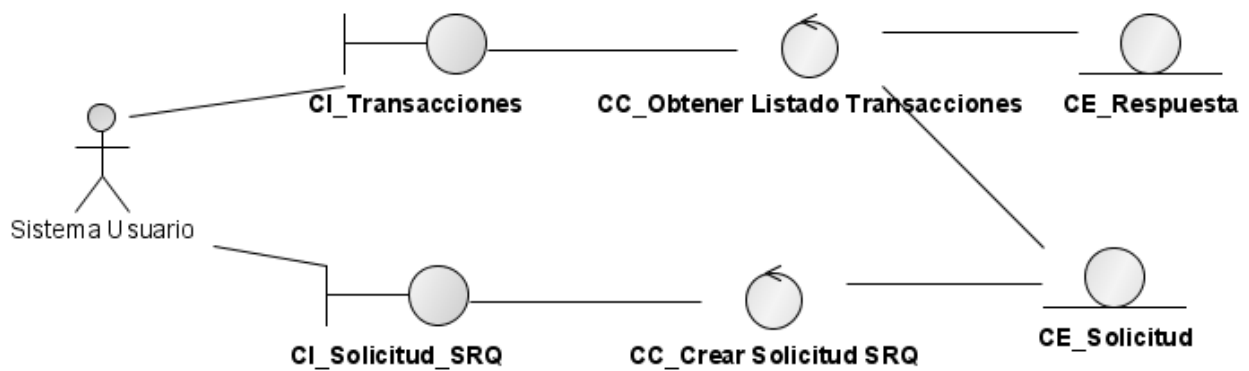


Figura III.2: Diagrama de clases del análisis para el paquete de servicios web

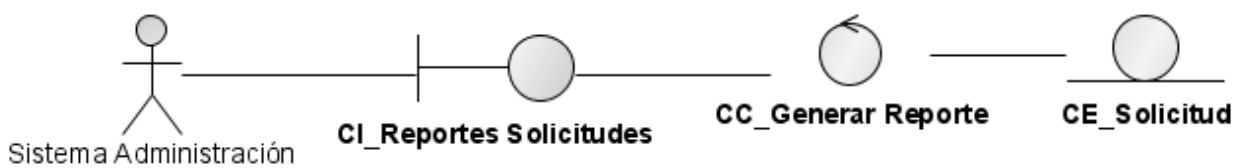


Figura III.3: Diagrama de clases del análisis para el sub-paquete servicio web con el administrador

Tabla 45: Descripción de Clase controladora. Validator

Nombre: validator	
Tipo de clase: Controladora	
Atributo	Tipo
--	--
Para cada responsabilidad:	
Nombre:	isInteger(\$cad)

Descripción:	Responsable de validar que los datos enviados de una cadena sea un entero.
Nombre:	isIP(\$cad)
Descripción:	Retorna verdadero o falso si la cadena pasada por parámetro es un ip.
Nombre:	isEmpty(\$cad)
Descripción:	Retorna verdadero o falso si la cadena pasada por parámetro está vacía.
Nombre:	isEmail(\$cad)
Descripción:	Retorna verdadero o falso si la cadena pasada por parámetro es una dirección de correo electrónico.
Nombre:	isUrl(\$cad)
Descripción:	Retorna verdadero o falso si la cadena pasada por parámetro es una Url.
Nombre:	isFloat(\$cad)
Descripción:	Retorna verdadero o falso si la cadena pasada por parámetro es un float.
Nombre:	isDate(\$cad)
Descripción:	Retorna verdadero o falso si la cadena pasada por parámetro fecha.
Nombre:	isName(\$cad)
Descripción:	Retorna verdadero o falso si la cadena pasada por parámetro es nombre correcto.
Nombre:	compare(\$cad, \$cad_compare)
Descripción:	Retorna verdadero o falso si las cadenas pasadas por parámetro son iguales.
Nombre:	isString(\$cad)
Descripción:	Retorna verdadero o falso si la cadena pasada por parámetro es un string.
Nombre:	isNull(\$array)
Descripción:	Retorna verdadero o falso si el arreglo pasado por parámetro es null.
Nombre:	validarNumCedula(\$numcedula)
Descripción:	Retorna verdadero o falso si la cadena pasada por parámetro es un número de cédula correcto.
Nombre:	validarLetra(\$letra)
Descripción:	Retorna verdadero o falso si el carácter pasado por parámetro es letra (V o E).
Nombre:	validarTCN(\$tcn)
Descripción:	Retorna verdadero o falso si la cadena pasada por parámetro es un TCN (identificador de una transacción) correcto.

DIAGRAMA DE INTERACCIÓN

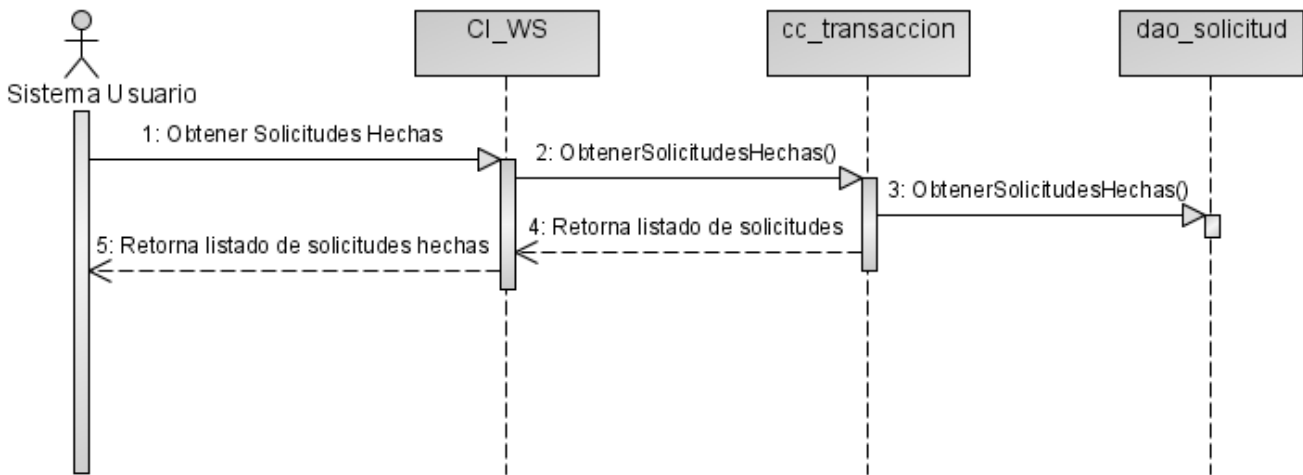


Figura III.4 Diagrama de secuencia del diseño del caso de uso “Mostrar Listado de Transacciones”. Escenario Obtener Solicitudes

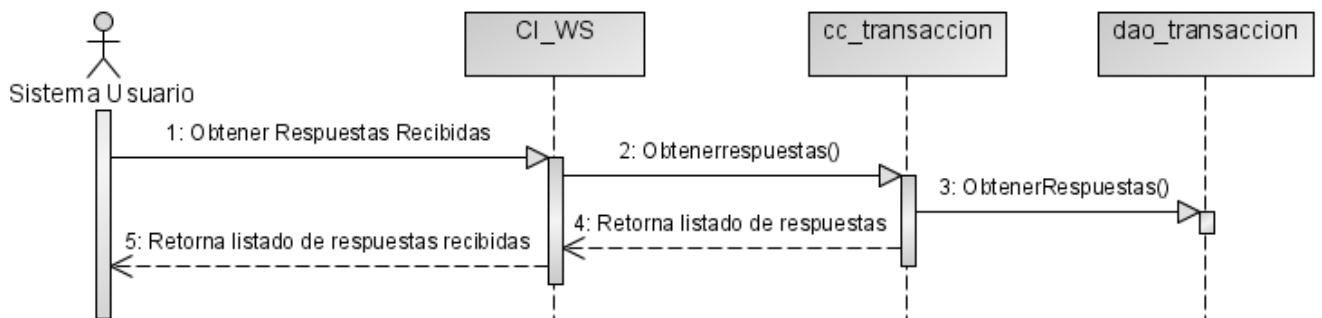


Figura III.5: Diagrama de secuencia del diseño del caso de uso “Mostrar Listado de Transacciones”. Escenario Obtener Respuestas

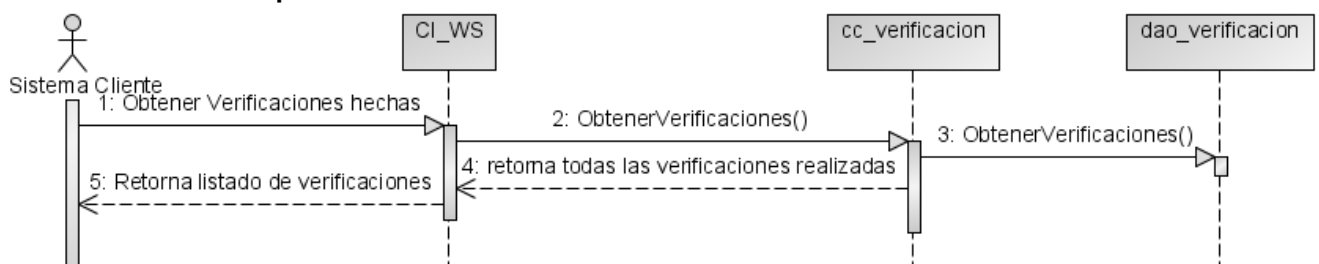


Figura III.6 Diagrama de secuencia del diseño del caso de uso “Mostrar Listado de Verificaciones”

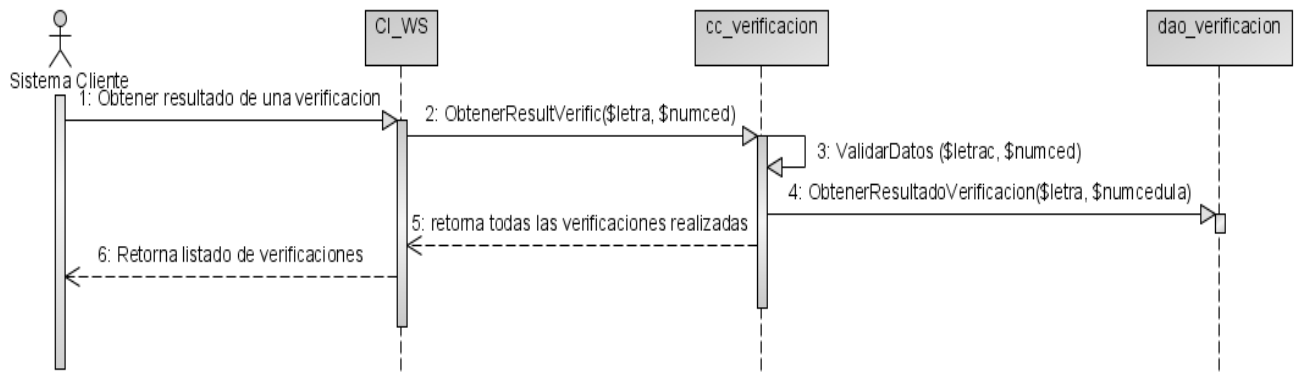


Figura III.7: Diagrama de secuencia del diseño del caso de uso “Obtener Resultados de Verificaciones”

Tabla 46: Descripción de la tabla de nTipoTransAfis del modelo entidad-relación

Nombre: nTipoTransAfis		
Descripción: Tabla que almacena el tipo de transacción AFIS y su descripción.		
Atributo	Tipo	Descripción
TOT	char(3)	Identificador de la tabla.
descripcion	varchar(50)	Descripción del tipo de transacción.

Tabla 47: Descripción de la tabla de dTSolicIdAtAFIS del modelo entidad-relación

Nombre: dTSolicIdAtAFIS		
Descripción: Tabla que almacena los datos correspondientes a la especialización de solicitud IDI, ATU.		
Atributo	Tipo	Descripción
TCN	integer(0)	Identificador de la tabla.
BIOID	integer(0)	Identificador biométrico.
JUV	char(1)	Especifica si es juvenil o no la persona a la que se le hace la solicitud de este tipo.
IMP	varchar(10)	Tipo de impresión.
DATA_ORI	char(1)	Identificador de la tabla nOrigenDatos, donde se almacena la descripción de los datos de origen.

Tabla 48: Descripción de la tabla de dTSolicIRQAfis del modelo entidad-relación

Nombre: dTSolicIRQAfis		
Descripción: Tabla que almacena los datos correspondientes a una solicitud tipo IRQ.		
Atributo	Tipo	Descripción
TCN	integer(0)	Identificador de la tabla.
BIOID	integer(0)	Identificador biométrico.
IDI	varchar(10)	Identificador del Registro de Imágenes

Tabla 49: Descripción de la tabla de dTSolicSRQAfis del modelo entidad-relación

Nombre: dTSolicSRQAfis		
Descripción: Tabla que almacena los datos correspondientes a la solicitud de tipo SRQ.		
Atributo	Tipo	Descripción
TCN	integer(0)	Identificador de la tabla.
RTN	integer(0)	Número de control (TCN) de la transacción cuyo estado de quiere recuperar.

Tabla 50: Descripción de la tabla de nServicioAfis del modelo entidad-relación

Nombre: nServicioAfis		
Descripción: Tabla que almacena los datos correspondientes a los servicios.		

Atributo	Tipo	Descripción
ideservicioafis	varchar(100)	Identificador de la tabla.
descripcion	varchar(100)	Descripción del servicio.
ESA	varchar(50)	Dirección electrónica de servicio.

Tabla 51: Descripción de la tabla de dSolicImagenAfis del modelo entidad-relación

Nombre: dSolicImagenAfis		
Descripción: Tabla que almacena los identificadores de una solicitud y los identificadores de las imágenes.		
Atributo	Tipo	Descripción
TCN	integer(0)	Identificador de la tabla dSolicitudAfis.
idImagen	integer(0)	Identificador de la tabla dImagenTranAfis.

Tabla 52: Descripción de la tabla de nEstadoTranAfis del modelo entidad-relación

Nombre: nEstadoTranAfis		
Descripción: Tabla que almacena el estado en que se encuentra una transacción.		
Atributo	Tipo	Descripción
TST	numeric(4, 0)	Identificador de la tabla.
descripcion	varchar(10)	Descripción del estado de una transacción.

Tabla 53: Descripción de la tabla de dResplImagenAfis del modelo entidad-relación

Nombre: dResplImagenAfis		
Descripción: Tabla que almacena los datos de las respuestas de imagen.		
Atributo	Tipo	Descripción
idImagen	integer(0)	Identificador de la tabla dImagenTranAfis.
TCN	integer(0)	Identificador de la tabla dRespuestaAfis.

Tabla 54: Descripción de la tabla de dTRespErrAfis del modelo entidad-relación

Nombre: dTRespErrAfis		
Descripción: Tabla que almacena los datos correspondientes a las respuestas de error.		
Atributo	Tipo	Descripción
ERM	varchar(3000)	Campo de mensaje de error.
TCN	integer(0)	Identificador de la tabla.

Tabla 55: Descripción de la tabla de nRespEstado del modelo entidad-relación

Nombre: nRespEstado		
Descripción: Tabla que almacena el estado de una respuesta.		
Atributo	Tipo	Descripción

RST	numeric(4, 0)	Identificador de la tabla.
descripcion	varchar(30)	Puede tomar los valores: Desconocida, procesando y completada.

Tabla 56: Descripción de la tabla de dTRespSTRAfis del modelo entidad-relación

Nombre: dTRespSTRAfis		
Descripción: Tabla que almacena los datos correspondientes a las respuestas de estado.		
Atributo	Tipo	Descripción
RTN	integer(0)	Campo de mensaje de error.
RST	numeric(4, 0)	Identificador de la tabla nRespEstado.
TCN	integer(0)	Identificador de la tabla.

Tabla 57: Descripción de la tabla de dTRespACKAfis del modelo entidad-relación

Nombre: dTRespACKAfis		
Descripción: Tabla que almacena el TCN de la respuesta de acuse de recibo (ACK).		
Atributo	Tipo	Descripción
TCN	integer(0)	Identificador de la tabla.

Tabla 58: Descripción de la tabla de dTRespEIRAfis del modelo entidad-relación

Nombre: dTRespEIRAfis		
Descripción: Tabla que almacena los datos correspondientes a las respuestas de identificación (EIR).		
Atributo	Tipo	Descripción
EIR_RETURN	varchar(10)	Retorno de identificación.
CANDLIST	varchar(256)	Lista de candidatos.
BIOID	integer(10)	Identificador biométrico.
TCN	integer(0)	Identificador de la tabla.

Tabla 59: Descripción de la tabla de dTRespEARAfis del modelo entidad-relación

Nombre: dTRespEARAfis		
Descripción: Tabla que almacena los datos correspondientes a las respuestas de autenticación (EAR).		
Atributo	Tipo	Descripción
TCN	integer(0)	Identificador de la tabla.
EAR_RETURN	varchar(10)	Retorno de identificación.
BIOID	integer(10)	Identificador biométrico.
FNG_REC_UPDT	char(1)	Registro de huellas actualizado.

Tabla 60: Descripción de la tabla de dTRespIMRAfis del modelo entidad-relación

Nombre: dTRespIMRAfis		
Descripción: Tabla que almacena los datos correspondientes a las respuestas de imágenes.		
Atributo	Tipo	Descripción
TCN	integer(0)	Identificador de la tabla.
CAP_DATE	varchar(10)	Fecha de Captura.
BIOID	integer(10)	Identificador biométrico.
FRP	char(1)	Registros de fichas presentes.
IRD1	varchar(20)	Puede llevar los valores: "ALONE" para indicar que es el único registro presente. "NOTALONE" para indicar que no es el único registro presente.
IRD2	varchar(20)	Puede llevar los valores: "BEST" para indicar que es el registro de mejor calidad. "NOTBEST" para indicar que no es el registro de mejor calidad.
IRD3	varchar(20)	Puede llevar los valores: "NEW" para indicar que es el registro de más reciente. "OLD" para indicar que es el registro de más antiguo.
IRD4	varchar(20)	Puede llevar los valores: "CONV" para indicar que es un registro de conversión. "NOTCONV" para indicar que no es un registro de conversión.
VFM	varchar(20)	Fila: numero de ficha (1 o 2) de origen de cada dedo de la ficha virtual, con 1 la ficha de mejor calidad.
JUV	char(1)	Especifica si es juvenil o no la persona a la que se le hace la solicitud de este tipo.
IMP	varchar(10)	Tipo de impresión.
DATA_ORI	char(1)	Tipo de origen de los datos.