

Universidad de las Ciencias Informáticas

Facultad 7



**Título: Sistema para el Diagnóstico
Neuropsicológico Automatizado
Diana.**

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

Autor(es): Yoeny Caballero González.
Aylín Parra Pérez.

Tutor(es): Lic. Eduardo Solis Céspedes.

Ciudad de la Habana, Junio de 2007.

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los 21 días del mes de Junio del año 2007.

Yoeny Caballero González.

Lic. Eduardo Solís Céspedes.

Aylín Parra Pérez.

DATOS DE CONTACTO

Lic. Eduardo Solis Céspedes (email: esolis@uci.cu).

Profesor graduado de Licenciatura en Radioquímica y especialista en Bioinformática. Ha impartido las asignaturas de Física I, Física II y Práctica Profesional. Es profesor de la facultad 7, jefe de proyecto dentro del Grupo de Procesamiento de Imágenes (GPI) de la Universidad de las Ciencias Informáticas (UCI), actualmente se desempeña como vice decano de producción de la facultad.

AGRADECIMIENTOS

A la Universidad de Ciencias Informáticas por permitir formarnos como ingenieros en ciencias informáticas. A todas las personas involucradas en la implementación de la solución informática que hoy traemos como ponencia de tesis y en especial a la Revolución por su grandeza.

DEDICATORIA

Dedico este trabajo a mis padres, a mi hermana Yany, a Yaima, a mi suegra; en fin a familiares y amigos que me han dado todo su apoyo emocional y espiritual en este tiempo. Ellos han aguantado a mi lado en mis momentos buenos, difíciles y malos con tanto amor, cariño y comprensión. Lo cual me ha favorecido en resistencia, poder y sabiduría, para mi soporte emocional. Dando a mi vida fuerzas de seguir adelante y lograr mi sueño, convertirme en ingeniero informático.

Yoeny.

A Jesus, porque me ha cuidado y ayudado tanto... A mi mamá Idania y mi abuela Ramona, que ya no están junto a mí y esto es lo que más deseaban. A Ana, mi nueva madre, porque sin ella no hubiera sido posible. A mi nueva familia, mi compañero Rilder que me ha apoyado tanto en estos momentos tan difíciles y Alex, mi niño del alma.

Aylín.

RESUMEN

El grupo de investigación neuropsicológica que pertenece al CNC¹, desarrolló en el año 1996 un sistema de evaluación cognitiva llamado Diana² v2.0, caracterizado como una herramienta eficiente y práctica, con una amplia gama de aplicaciones en la evaluación de funciones cognitivas y objetivas en sujetos, tanto adultos como niños. Actualmente, se encuentra en desuso debido al crecimiento elevado y variación de las tecnologías de la computación, díganse hardware y software, por presentar incompatibilidad con las mismas. De ahí que se imponga la necesidad de implantar un sistema que automatice estos procesos, garantizando la centralización, integridad y disponibilidad de este tipo de información.

El presente trabajo propone desarrollar un sistema que cumpliendo con las características arquitectónicas de las aplicaciones clientes servidor, que pueda lograr la gestión centralizada de pruebas neuropsicológicas automatizadas y la persistencia en los datos referentes a todo el proceso de evaluación cognitiva en sistemas automatizados de este tipo.

Nuestra solución informática cuenta con todo lo requerido en materia de gestión de pruebas neuropsicológicas, poniéndose a la altura de los principales sistemas que constituyen hitos de la evaluación cognitiva. Además, presta un servicio único de su tipo que se basa en el almacenamiento de los datos referentes al proceso de evaluación neuropsicológica en una base de datos, lo cual garantiza la persistencia de los mismos.

¹ Centro de Neurociencias de Cuba

² Diagnóstico Neuropsicológico Automatizado v2.0

PALABRAS CLAVES

Diana, persistencia de datos, proceso de evaluación cognitiva, pruebas neuropsicológicas.

Tabla de Contenidos

DECLARACIÓN DE AUTORÍA	I
DATOS DE CONTACTO	II
AGRADECIMIENTOS.....	III
DEDICATORIA	IV
RESUMEN.....	V
PALABRAS CLAVES.....	VI
INTRODUCCIÓN.....	1
CAPITULO 1 FUNDAMENTACIÓN TEÓRICA	6
UTILIZACIÓN DE LOS ESTÁNDARES EN LA MEDICINA.	6
1.1 CLASIFICADOR DE TRASTORNOS MENTALES CIE 10.....	6
1.2 MANUAL DE DIAGNÓSTICO ESTADÍSTICO DE LOS TRASTORNOS MENTALES DSM IV	7
1.3 BENEFICIOS DE LA NEUROPSICOLOGÍA EN SISTEMAS AUTOMATIZADOS.	7
1.4 ACTUALIDAD.	7
1.5 PRINCIPALES BATERÍAS NEUROPSICOLÓGICAS Y SISTEMAS USADOS COMO REFERENCIA.	8
1.5.1 Batería Neuropsicológica Luria-Nebraska, 1969.....	8
1.5.2 SCAN (Schedules for Clinical Assessment in Neuropsychiatry).....	9
1.5.3 WAIS/ WISC (1982 - 1991).....	9
1.5.4 PSISOFT (Softel).	9
1.5.5 SELSOFT (Softel).	10
1.5.6 DIANA versión 2.0 (NEURONIC S.A.).	10
1.6 PRINCIPALES HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS.....	11
1.6.1 RDAs (Aplicaciones Enriquecidas de Escritorio).....	11
1.6.2 Microsoft Visual Studio 2005	12
1.6.3 Visual CSharp .NET 2.0	13
1.6.4 Flash.	14
1.6.5 Web Services.	14
1.6.6 PostgreSQL 8.1.....	14
1.6.7 Enterprise Architect 6.5.	15
1.6.8 RUP (Proceso Unificado de Rational).....	16
1.6.9 UML (Lenguaje Unificado de Modelado).....	17
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	18
2.1 OBJETO DE AUTOMATIZACIÓN.	18
2.2 PROPUESTA DE SISTEMA.....	19
2.3 MODELO DEL NEGOCIO.....	19
2.4 DIAGRAMA DE CASOS DE USO DEL NEGOCIO.	21
2.5 DIAGRAMAS DE ACTIVIDADES DE LOS CASOS DE USO DEL NEGOCIO.	22
2.6 DIAGRAMA DE CLASES DEL MODELO DE OBJETOS.	24
2.7 ESPECIFICACIÓN DE LOS REQUERIMIENTOS DEL SOFTWARE.	25

2.7.1 Requerimientos Funcionales.	25
2.7.2 Requerimientos no Funcionales.	29
2.8 CASOS DE USO DEL DIANA.	30
2.8.1 Actores del sistema a automatizar.	30
2.8.2 Casos de Uso.	32
2.8.3 Diagrama de Casos de Uso.	37
2.8.4 Planificación de los casos de uso por ciclos de desarrollo.	39
2.8.5 Descripción extendida de los Casos de Uso.	42
CAPÍTULO 3: ARQUITECTURA Y DISEÑO DEL SISTEMA	48
3.1 MODELO ARQUITECTÓNICO.	48
3.2 MODELO DEL DISEÑO	51
3.3 DESCRIPCIÓN DE LAS CLASES DE DISEÑO	51
3.4 DISEÑO DE LA BASE DE DATOS.	51
3.5 DESCRIPCIÓN DE LAS TABLAS.	51
CAPÍTULO 4: IMPLEMENTACIÓN	52
4.1 DIAGRAMA DE COMPONENTES.	52
4.2 DIAGRAMA DE DESPLIEGUE.	54
CONCLUSIONES	55
RECOMENDACIONES	56
REFERENCIAS BIBLIOGRÁFICAS	57
BIBLIOGRAFÍA	59
ANEXO 1: DIAGRAMAS DE REQUERIMIENTOS	61
A.1.1 DIAGRAMA DE REQUERIMIENTOS.	61
A.1.2 CLIENTE DIANA.	62
A.1.3 REQUERIMIENTOS DEL CLIENTE DIANA, PRINCIPALES SERVICIOS.	63
A.1.4 REQUERIMIENTOS DEL CLIENTE DIANA, PRINCIPALES SERVICIOS, CONSULTAR PACIENTES.	64
A.1.5 REQUERIMIENTOS DEL CLIENTE DIANA, PRINCIPALES SERVICIOS, REPORTES.	65
A.1.6 REQUERIMIENTOS DEL CLIENTE DIANA, GESTIÓN	66
A.1.7 REQUERIMIENTOS DEL REPOSITORIO DE INFORMACIÓN	67
ANEXO 2: REALIZACIÓN DE CASOS DE USO DEL DISEÑO	68
A.2.1 CASO DE USO CONSULTAR PACIENTES	68
A.2.2 CASO DE USO CONFIGURAR PRUEBAS NEUROPSICOLÓGICAS	70
A.2.4 CASO DE USO CONFIGURAR CONEXIÓN AL REPOSITORIO.	72
A.2.5 CASO DE USO GENERAR REPORTES.	74
A.2.6 CASO DE USO ALMACENAR INFORMACIÓN EN LA BASE DE DATOS.	76
ANEXO 3: DESCRIPCIÓN DE LAS CLASES DEL DISEÑO	78
A.3.1 DESCRIPCIÓN DE LA CLASE DIANASERVICE.	78
A.3.2 DESCRIPCIÓN DE LA CLASE GLOBALTEST	78
A.3.3 DESCRIPCIÓN DE LA CLASE COMPRENSIONSINTACTICA.	80

A.3.4 DESCRIPCIÓN DE LA CLASE DATABASECONFIGURATION.....	80
A.3.5 DESCRIPCIÓN DE LA CLASE SERVER.....	81
A.3.6 DESCRIPCIÓN DE LA CLASE CONFIGURATION	82
A.3.7 DESCRIPCIÓN DE LA CLASE INFORMATIONXML	83
A.3.8 DESCRIPCIÓN DE LA CLASE SPECIALIST.....	84
A.8.9 DESCRIPCIÓN DE LA CLASE CONSUTATESTÍTEM	85
A.8.10 DESCRIPCIÓN DE LA CLASE REPORTE	85
A.8.11 DESCRIPCIÓN DE LA CLASE PATIENT	86
A.8.12 DESCRIPCIÓN DE LA CLASE BUSCARPACIENTE.....	87
A.8.13 DESCRIPCIÓN DE LA CLASE CONSULTA.....	88
ANEXO 4: MODELO DE DATOS	90
A.4.1 DIAGRAMAS ENTIDAD - RELACIÓN.....	90
A.4.2 DESCRIPCIÓN DE LAS TABLAS DE LA BASE DE DATOS.....	92
A.4.2.1 Gestión de datos generales.	92
A.4.2.2 Pruebas neuropsicológicas	94
ANEXO 6: MODELO DE DESPLIEGUE DEL SISTEMA	100
GLOSARIO DE TÉRMINOS	101

Índice de Figuras

Figura 1 – Flash + . NET	11
Figura 2 – Lógica de funcionamiento de los RDAs	11
Figura 3 – Actores del cliente diana	30
Figura 4 – Actores del repositorio de información.	30
Figura 5 – Diagrama de Caso de Uso del Sistema.....	36
Figura 6 - Diagrama de Casos de Uso del Sistema.....	37
Figura 7 – Diagramas de Casos de Uso de Sistema	38
Figura 8 – Arquitectura Física del sistema	48
Figura 9 – Capas del sistema.....	49
Figura 10 – Diagrama de Componentes	52
Figura 11 – Requerimientos funcionales	60
Figura 12 – Requerimientos funcionales del cliente diana.....	61
Figura 13 – Requerimientos funcionales, principales servicios del cliente diana	62
Figura 14 – Requerimientos funcionales, consultar pacientes.....	63
Figura 15 – Requerimientos funcionales, reportes	64
Figura 16 – Requerimientos funcionales, gestión.....	65
Figura 17 – Requerimientos funcionales, repositorio.....	66
Figura 18 – Diagrama de Secuencia – Consultar Pacientes.....	67
Figura 19 – Diagrama de Clases – Consultar Pacientes	68
Figura 20 – Diagrama de Secuencia – Configurar pruebas neuropsicológicas.....	69
Figura 21 – Diagrama de Clases – Configurar pruebas neuropsicológicas	70
Figura 22 – Diagrama de Secuencia – Configurar Conexión al repositorio.....	71
Figura 23 – Diagrama de Clases – Configurar Conexión al repositorio	72
Figura 24 – Diagrama de Secuencia – Generar Reportes.....	73
Figura 25 – Diagrama de Clases – Generar Reportes.....	74
Figura 26 – Diagrama de Secuencia – Almacenar información en la base de datos	75
Figura 27 – Diagrama de Clases – Almacenar información en la base de datos	76
Figura 28 – Diagrama Entidad Relación, gestión de datos generales	89
Figura 29 – Diagrama Entidad Relación, pruebas neuropsicológicas.....	92
Figura 30 – Diagrama de Despliegue.....	101

INTRODUCCIÓN

Los últimos 30 años han visto el surgimiento y proliferación de la automatización de la evaluación de las funciones cognitivas en general. La utilización de herramientas de alta tecnología ha hecho posible la implementación de los modelos teóricos informacionales de la arquitectura cognitiva y, de este modo, la posibilidad de examinar funciones y capacidades humanas antes virtualmente inaccesibles tanto a nivel de macroanálisis (simulación de contextos cuasi-ecológicos: los simuladores de vuelo, de conducción de autos, etc.), como a nivel de microanálisis (diseño de tareas cognitivas elementales y/o diseño de redes neuronales que reproducen la competencia de una función cognitiva humana).

En correspondencia con la perspectiva del microanálisis, en la actualidad se dispone de una cantidad considerable de sistemas automatizados de evaluación de las funciones cognitivas que se aplican en los más diversos campos de las ciencias del comportamiento (neurología y psiquiatría; medicina ocupacional, legal del deporte; selección, entretenimiento y evaluación del personal) y no sólo en el diagnóstico sino también y cada vez más, en la rehabilitación y el entrenamiento de capacidades.

El Diana (Diagnóstico Neuropsicológico Automatizado) es uno de los software por excelencia para la evaluación de estas funciones de cognición, resulta un sistema de utilidad para el diagnóstico, la rehabilitación y el entrenamiento, tanto desde una óptica clínica, en poblaciones geriátricas, con problemas de toxicidad, trastornos del aprendizaje, como por afecciones en general del sistema nervioso. Diana 2.0 incluye 27 tareas o pruebas que abarcan los principales dominios o sistemas de funciones cognitivas e incluye una tarea para la exploración del estado de las funciones afectivas [1].

Estas pruebas fueron creadas por distintas personas en distintos años y algunas de ellas fueron usadas para evaluar psicológicamente a soldados en la segunda guerra mundial. Desde aquel entonces y durante mucho tiempo se realizaban a papel y lápiz, lo que nos hace ver que, los instrumentos que se utilizaban para medir el tiempo e inclusive el estado emocional del especialista o técnico que aplicaba la prueba, influía en el resultado de las mismas, de ahí la necesidad de automatizarlas.

En neuropsicología, las ventajas de los métodos automatizados son evidentes: los criterios cronométricos pueden ser aplicados con absoluta confianza; los resultados no están sujetos a las variaciones propias de

las diferencias entre evaluadores o del mismo evaluador en el curso del tiempo; la recogida de los resultados no exige un personal altamente entrenado y se efectúa precisa e inmediatamente; el procesamiento de dichos resultados puede ser también inmediato y realizarse así, dentro de una misma sesión, las modificaciones necesarias en cada caso; en el caso de niños en especial, los métodos computarizados garantizan un alto nivel de motivación [2].

Hasta el momento se contaba con la versión 2.0 de Diana que fue construida en 1996 por la necesidad de evaluar el estado de las funciones nerviosas de trabajadores del petróleo en Colombia, y hasta hoy se ha utilizado esta versión también en Panamá, donde se vendieron alrededor de 48 licencias a la Caja del Seguro Social. En Cuba la utilizaban varias instituciones como el INDER, el MININT, el CIREN, Instituto de Medicina del Trabajo, Centro de Neurociencias y el MINFAR.

A raíz de que el sistema Diana 2.0 fue implementado y desplegado a mediados de la revolución informática en que estamos, y no se tuvo en cuenta el diseño de una arquitectura capaz de prevalecer al paso de las nuevas tecnologías.

Actualmente se cuenta con nuevos requerimientos en el ámbito de las pruebas psicológicas y la gestión de la información referente a las mismas que incluye el almacenamiento persistente de los datos relacionados con este tipo de estudio. Los nuevos sistemas y tecnologías son más exigentes en lo que respecta a robustez, rendimiento (performance), y en su inmensa mayoría suelen ser incompatible con las viejas tecnologías. Diana 2.0 no funciona bajo las arquitecturas de hardware existentes, de ahí que el análisis de la problemática expuesta afluya al planteamiento del siguiente **problema**: ¿Cómo solucionar el problema de la gestión de consultas y pruebas neuropsicológicas que realizaba el diana en su versión 2.0, garantizando además la disponibilidad, el almacenamiento y la seguridad de éstas?

Con vistas a dar solución al problema planteado, quedan definidos como **objeto de estudio** las pruebas neuropsicológicas de los dominios de memoria, procesos ejecutivos, procesos viso espaciales, atención y lenguaje. A raíz de lo cual, el **campo de acción** se enmarca en la automatización de las pruebas de los dominios antes mencionados.

El **objetivo general de la investigación** es desarrollar una solución de software que funcione según las nuevas tecnologías, con perspectiva y futuro, capaz de gestionar las pruebas neuropsicológicas.

En la investigación que nos ocupa camino al estudio y desarrollo de nuestra solución quedan definidas una serie de **tareas** a realizar:

- ✓ Investigar sobre el estado del arte del Diana 2.0 y de los sistemas de este tipo existentes en el mundo.
- ✓ Investigar sobre las nuevas tecnologías en el mundo para seleccionar las más apropiadas para el nuevo sistema.
- ✓ Definir una arquitectura acorde con la situación actual del proceso de evaluación cognitiva.
- ✓ Realizar diseño del sistema informático utilizando patrones que mejoren la reusabilidad y extensibilidad.
- ✓ Implementar el sistema.

Para poder cumplir con las tareas antes mencionadas, se emplearon los siguientes **métodos científicos**:

- ✓ Métodos Teóricos.

Modelación: Este método lo consideramos de gran importancia para la confección de nuestro sistema puesto que para su realización hemos utilizado como metodología RUP³, con la cual se hace muy necesaria la creación de varios artefactos en los diferentes modelos que se construyen, esto nos permite una reproducción amplia de la realidad.

Analítico – Sintético: Lo que nos ha permitido avanzar en nuestra solución es conocer las teorías y documentos que ocupan nuestro objetivo de investigación, extrayendo los aspectos más importantes relacionados con nuestro objeto de estudio.

Histórico – Lógico: El mismo nos ayuda a comprender la evolución en el mundo de los sistemas para el Diagnóstico Neuropsicológico Automatizado a lo largo de la historia, además, nos facilita la comprensión de la esencia de nuestro objeto de estudio.

³ Rational Unified Process

✓ Métodos Empíricos.

Observación: Este método nos ha sido de suma importancia puesto que nos ha permitido observar mediante el registro visual lo que ocurre en la situación real que nos ocupa.

Entrevista: Mediante la aplicación de este método hemos podido saber los criterios del cliente respecto al sistema, además, mediante él surgen nuevos requerimientos para el software.

El trabajo consta de cuatro capítulos donde se realizará un estudio crítico y descriptivo de las tecnologías seleccionadas, se plantearán las características del sistema comenzando por una vista de los procesos del negocio existentes, se diseñará el sistema propuesto y luego se implementará cumpliendo así con los principales flujos del ciclo de desarrollo.

En el Capítulo 1 se tratarán los temas relacionados con la fundamentación teórica que justifica nuestra investigación, se analizará el estado actual del tema que nos ocupa a nivel nacional e internacional así como las nuevas tendencias y las tecnologías usadas en nuestra solución, se fundamentará la metodología usada y se abordarán otros temas de interés.

En el Capítulo 2 se abordarán las características del sistema propuesto, se describirán el objeto de automatización y la propuesta de sistema, se analizará además el Modelo de Negocio que incluye la presentación de los actores y trabajadores, los diagramas de casos de uso del negocio y los de actividades, se presentará además el diagrama de clases correspondiente al Modelo de Objetos y se describirán los requerimientos del sistema que constituyen la base para la posterior realización del Modelo de Casos de Uso.

En el Capítulo 3 se abordará la arquitectura y el diseño del sistema a través de los modelos arquitectónicos y de diseño donde se detallan las características de la arquitectura empleada y se definen las clases del diseño y sus relaciones respectivamente, se realizará también el Modelo de Datos que incluye el diseño de la base de datos y la descripción de las tablas que la componen.

En el Capítulo 4 se tratarán los artefactos generados por el Modelo de Implementación y Despliegue, esencialmente los diagramas de componentes y despliegue donde se hace un profundo análisis de cada componente y la relación entre ellos así como su integración en cada nodo.

Los Anexos incorporados al final del documento contendrán diferentes artefactos generados durante cada fase y flujo de trabajo del ciclo de desarrollo según la metodología empleada, se incluye la descripción detallada de las clases, de diferentes elementos del sistema, entre otros.

En el Glosario de Términos se definen claramente los términos relacionados con el tema que nos ocupa u otro tema que pueden ser de difícil comprensión.

CAPITULO 1 FUNDAMENTACIÓN TEÓRICA

En este capítulo se describirán varios tipos de estándares CIE 10 y el DSM IV. Se abordan los temas del proceso de evaluación cognitiva para ubicar el contexto en el que puede enmarcarse el sistema propuesto por la presente investigación. Se hace un estudio del estado del arte sobre gestión automatizada de las pruebas neuropsicologías a nivel internacional y nacional. Se presentan las principales tendencias en el mundo de las tecnologías para darle solución a problemas similares al nuestro. Se describen las principales características de las tecnologías utilizadas en el diseño e implementación del sistema propuesto.

Utilización de los estándares en la medicina.

Estándar es una norma a seguir, un consentimiento general. La utilización de los estándares en las ciencias médicas es de vital importancia, se garantiza la compatibilidad de los resultados y el entendimiento de los mismos por todos los especialistas que traten estos temas. Además, de servir como base a los implicados en la construcción de los sistemas enmarcados en este campo de acción.

1.1 Clasificador de trastornos mentales CIE 10.

La clasificación de trastornos mentales del CIE 10 está diseñada para promover la comparación internacional de la recolección, procesamiento, clasificación y presentación de estadísticas, asignándole a las enfermedades un código fijo de hasta seis caracteres de longitud con el siguiente formato X00.00. El propósito de esta clasificación es permitir el registro sistemático, el análisis, la interpretación y la comparación de datos de mortalidad y morbilidad recolectados en diferentes países o áreas, y en diferentes épocas [3].

Este codificador es el más usado en las pruebas neuropsicológicas automatizadas para la lectura de enfermedades de esta rama, las pruebas del Diana están regidas por este estándar, los resultados obtenidos por la aplicación de las pruebas se encuentran preparados para en cualquier momento hacer comparaciones a cualquier nivel, los cuales se encuentran estandarizados por este estándar médico.

1.2 Manual de diagnóstico estadístico de los trastornos mentales DSM IV .

El Manual de Diagnóstico estadístico de los trastornos mentales proporciona descripciones claras de los diagnósticos, con el fin de que los especialistas y los investigadores puedan diagnosticar, estudiar e intercambiar información y tratar los distintos trastornos mentales fácilmente [4]. Este es el clasificador más usado en el mundo de la neuropsicología.

1.3 Beneficios de la neuropsicología en sistemas automatizados.

En su sentido moderno, la neuropsicología es el estudio de las funciones psíquicas en general y cognitivas en particular, que ocurren como consecuencia de lesiones del sistema nervioso central y del cerebro en especial. La razón de su popularidad se debe a que contribuye a elaborar los perfiles clínicos de las alteraciones, ofrece elementos decisivos para formular inferencias acerca de la organización y funcionamiento del cerebro en su estado normal y aporta datos a tener en consideración en los planes de tratamiento y rehabilitación. La utilización de herramientas de alta tecnología ha hecho posible la implementación de los modelos teóricos informacionales de la arquitectura cognitiva y, de este modo, la posibilidad de examinar funciones y capacidades humanas [5].

1.4 Actualidad.

El tema de la neuropsicología es de primer impacto en el mundo y por ello muchas organizaciones y empresas se dedican tanto al estudio de este tema como al desarrollo de herramientas que faciliten en general la gestión requerida por ellos.

En Estados Unidos existe una organización llamada APA (American Psychological Association), que es una de las más grandes a nivel mundial y tiene como objetivo favorecer el avance de la psicología como ciencia y profesión, además de servir como medio de promoción para la salud, educación y la asistencia social en este aspecto [6].

En 1998 surgió en Valencia el grupo Previ, el primer equipo multidisciplinario especializado en las distintas áreas de la Psicología Clínica. Desde su creación, el equipo Previ ha atendido a más de 1.000 pacientes, abordando la evaluación y el tratamiento psicológico desde una perspectiva cognitivo-comportamental. Una perspectiva que cuenta con el aval de la investigación científica rigurosa, y que analiza la manera en que las personas interpretan la realidad que les rodea. En este sentido, el Grupo Previ ofrece tratamientos

empíricamente validados atendiendo al *Informe sobre Promoción y Difusión de Procedimientos Psicológicos* elaborado por la División de Psicología Clínica de la American Psychological Association (APA) [7]. Existen además a nivel mundial sistemas informáticos, herramientas y baterías de pruebas que se encargan del estudio de este tema.

El Centro de Neurociencias de Cuba es un paradigma en el mundo de la neuropsicología, tiene como mérito haber hecho grandes aportes en materia de herramientas y soluciones informáticas en esta rama, el sistema más importante con el que hoy cuenta este centro es el DIANA en su versión 2.0 aunque hoy en día existen varios problemas con el uso de este importantísimo software.

Actualmente se desarrolla en la UCI⁴ la nueva solución denominada Diana, que constituye una nueva versión, por parte del grupo de proyectos GPI⁵ perteneciente a la Facultad 7, en el seno la infraestructura productiva perteneciente a la dirección de producción de la UCI.

1.5 Principales baterías neuropsicológicas y sistemas usados como referencia.

Las baterías de pruebas automatizadas, son un conjunto de pruebas agrupadas por dominios, existen de distintas ramas de la ciencia, las que nos ocupan en nuestro trabajo son las neuropsicológicas. Generalmente son comercializadas con los sistemas que se encargan de la gestión de las misma.

1.5.1 Batería Neuropsicológica Luria-Nebraska, 1969.

La batería neuropsicológica Luria-Nebraska (LNNB en sus siglas en inglés) es una batería derivada de procedimientos clínicos y basada en la teoría del funcionamiento cortical superior del neuropsicólogo ruso Aleksandr R. Luria. En contraste con un enfoque más psicométrico y cuantitativo usado por la mayoría de los neuropsicólogos clínicos, Luria utilizaba técnicas cualitativas no estructuradas para detectar pacientes con deterioro neurológico. La recopilación, sistematización y divulgación en Europa y Norteamérica, de su trabajo, fue realizada por la psicóloga danesa Anne Lise Christensen, quien en 1975 publicó el primer Manual de Aplicación, para las pruebas de Luria. En Estados Unidos, Golden, Purisch y Hammeke sistematizaron los indicadores de Luria y presentaron a finales de los 70, un instrumento de medición diseñado para discriminar pacientes con daño neurológico, dentro del marco de una neuropsicología

⁴ Universidad de las Ciencias Informáticas

⁵ Grupo de Procesamiento de Imágenes

clínica estadounidense. A éste instrumento le denominaron la Batería Neuropsicológica Luria-Nebraska [8].

1.5.2 SCAN (Schedules for Clinical Assessment in Neuropsychiatry).

El Sistema SCAN, denominado en español "Cuestionario para la Evaluación Clínica en Neuropsiquiatría", ha sido desarrollado por la Organización Mundial de la Salud, y se compone de un conjunto de instrumentos destinados a analizar, medir y clasificar la psicopatología y la conducta asociada a las principales alteraciones psiquiátricas de la edad adulta. Es uno de los instrumentos más prometedores de los que dispone la psiquiatría actual [9]. Tiene grandes aplicaciones en la clínica, la docencia y la investigación.

1.5.3 WAIS/ WISC (1982 - 1991).

Existe una batería para niños (WISC) y otra para adultos (WAIS) que están conformadas por una serie de subpruebas que se agrupan en una Escala Verbal y otra Escala Manual. El desempeño de la persona en la prueba se resume en un Coeficiente Intelectual Total y en un coeficiente para cada subescala, es decir un CI Verbal y un CI Manual, los cuales permiten ubicar y comparar su rendimiento respecto de lo esperado para su grupo de edad. Además, la última versión entrega cuatro puntajes índices, basados en un análisis factorial, que permiten obtener un análisis más fino del funcionamiento intelectual del sujeto y que evalúa la Comprensión Verbal, la Organización Perceptual, la Destructibilidad (WISC) o Memoria de Trabajo (WAIS) y la Velocidad de Procesamiento de la información [10].

1.5.4 PSISOFT (Softel).

Sistema automatizado para la manipulación de la historia clínica general del paciente y la obtención del Diagnóstico Psiquiátrico por el DSM IV o el CIE-10 a través de una discusión diagnóstica que parte de definir un diagnóstico Síndromo Presuntivo. Puede usarse con fines docentes o investigativos [11]. Desarrollado por la empresa cubana de software Softel aledaña a nuestra universidad, el sistema también puede ser utilizado con fines docentes e investigativos. Se encuentra instalado en el reconocido Hospital Psiquiátrico de La Habana, y en la clínica del MININT, y fue adquirido además, por especialistas de Venezuela, México, Argentina, Chile y República Dominicana.

Psisoft facilita el trabajo de psiquiatras y psicólogos, además, posibilita obtener un diagnóstico psiquiátrico utilizando los clasificadores que se han logrado imponer internacionalmente. Igualmente permite la lectura de los resultados de las pruebas a partir de un scanner y proporciona pruebas con imágenes o sin ellas, entre otras ventajas.

1.5.5 SELSOFT (Softel).

Es un sistema automatizado para la selección de personal. Pruebas (test) para realizar una evaluación de inteligencia, de personalidad, de habilidad, de aptitud y de preferencias vocacionales. Múltiples perfiles para puestos de trabajo y otras características que le serán de gran utilidad [12]. Selsoft surge a raíz de los resultados obtenidos por el despliegue de la solución antes vista Psisoft, a petición de la Banca Mexicana. Es un Sistema informático para la selección de Recursos Humanos, sale al mercado en el año 1995 y se basa en la evaluación integral de los requisitos exigidos por el perfil psicológico en el proceso de selección y evaluación de los puestos de trabajo.

La evaluación neuropsicológica se realiza mediante la utilización de un grupo de pruebas psicométricas de personalidad, aptitud, habilidad, inteligencia y preferencias vocacionales, que permiten crear el expediente del candidato con la hoja de vida y los resultados de las pruebas aplicadas facilitando la evaluación integral del mismo.

1.5.6 DIANA versión 2.0 (NEURONIC S.A.).

Sistema automatizado de evaluación cognitiva. El propósito principal de Diana es brindar una herramienta eficiente y práctica, con una amplia gama de aplicaciones en la evaluación de funciones cognitivas y objetivas de sujetos adultos, no obstante, mucha de las pruebas disponibles en Diana, pueden ser confiablemente administradas a niños desde 6 años en adelante. Diana resulta un sistema de utilidad para el diagnóstico, la rehabilitación y el entrenamiento; conserva las características de estandarización y tipicidad que distinguen a cada una de las pruebas implementadas y ofrece, al mismo tiempo, la mayor sencillez de operación y la mayor facilidad de administración [13].

1.6 Principales herramientas y tecnologías utilizadas.

1.6.1 RDAs (Aplicaciones Enriquecidas de Escritorio).

¿Por qué utilizar la tecnología RDA?, se ejecutan de la misma forma que una aplicación de escritorio, son aplicaciones que utilizan servicios en línea (web services), pero que se ejecutan dentro de nuestra máquina, estos sistemas tienen la capacidad de aprovechar los recursos de nuestro sistema y sobre todo le da la posibilidad al Flash Player de acceder a ficheros de la PC, aspecto de suma importancia para la solución propuesta.

RDA, Flash + .NET

Decidimos usar .NET para lograr la funcionalidad deseada con la que no contaba el flash player, acceso al sistema de archivos y Flash para implementar las pruebas neuropsicológicas debido a la facilidad y riqueza de interfaz que le da a los usuarios. Flash aporta su rica interfaz de usuario, aspecto necesario para la batería de pruebas, y .NET aporta el control total sobre el cliente.

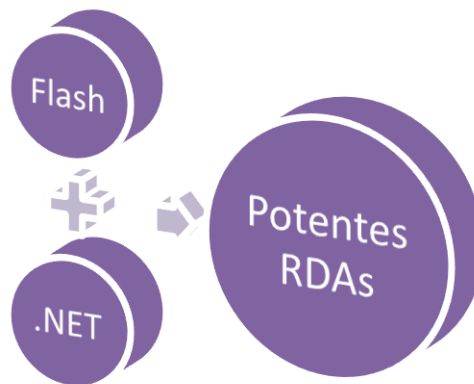


Figura 1. Flash + .NET.

Se logra con la utilización de esta tecnología un desarrollo rápido, software potente y una entrega rápida al cliente. La plataforma .NET está diseñada para ser multiplataforma. Una desventaja de usar esta tecnología es que requiere del Framework de .NET pero las licencias para su uso ya fueron adquiridas por la comercializadora del CNC (Neuronic SA), como también las del Visual Studio.

¿Qué se requiere para nuestra solución? Se requiere implementar las pruebas neuropsicológicas empujadas flash, configurar la prueba dinámicamente, almacenar los resultados de las pruebas en un archivo y enviar información hacia un servidor de base de datos. El uso de un objeto COM que nos provee la plataforma para incrustar Flash Player dentro de una aplicación de escritorio y el espacio de nombres ShockwaveFlashObjects posee las clases necesarias para controlar Flash Player desde .NET. Otras de las funciones que nos provee .NET, es la utilización de la tecnología ADO para el flujo de los XMLs.

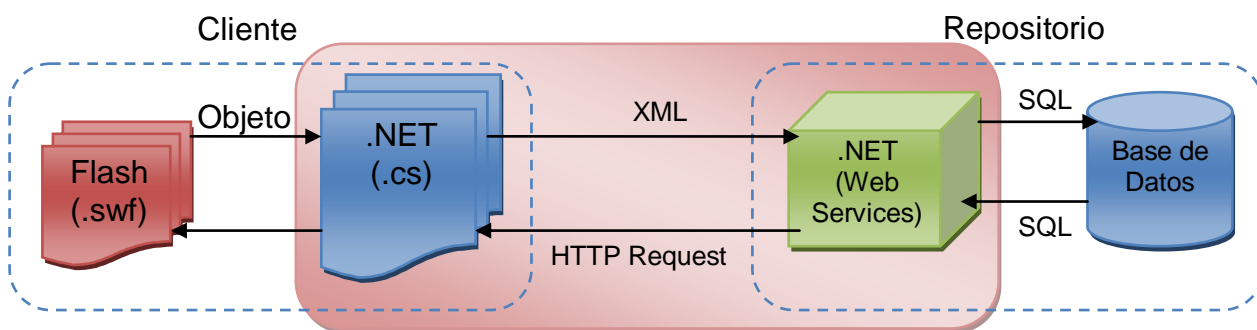


Figura 2. Lógica de funcionamiento de los RDAs.

1.6.2 Microsoft Visual Studio 2005

Visual Studio .NET es un conjunto completo de herramientas de desarrollo para la construcción de aplicaciones Web ASP, servicios Web XML, aplicaciones para escritorio, archivos DLL, aplicaciones de consola y aplicaciones móviles.

Ofrece algunas características exclusivas de alta productividad, tales como: IntelliSense; diseñadores visuales de Web Forms y Windows Forms, esquemas XML y datos; un depurador de varios lenguajes que alterna sin problemas entre códigos escritos en lenguajes diferentes; una estrecha integración con .NET Framework; ayuda dinámica, que proporciona una ayuda contextual continua mientras se escribe; lista de tareas, que muestra los errores del compilador y las tareas pendientes; características de diseño de arquitecturas como la integración con Visio y un explorador de servidores para obtener acceso visual a

bases de datos, servicios de Windows, contadores de rendimiento y componentes de aplicaciones del lado del servidor [14].

Visual Studio es actualmente el IDE por excelencia para la mayoría de los desarrolladores, plataforma solo comparada en algunos aspectos con el NetBeans y el Eclipse ambos para el lenguaje java, los antes mencionados no cumplen con todas las características vistas. Todas estas particularidades hacen del Visual Studio la mejor y más práctica herramienta para desarrollar nuestra aplicación. También utiliza APIs para la criptografía, las cookies de seguridad y la autenticación y ofrece la opción de controlar el acceso de lectura y escritura sobre determinados recursos por lo que sin lugar a dudas este es el IDE ideal para el desarrollo de nuestro sistema propuesta.

1.6.3 Visual CSharp .NET 2.0

Visual C#.NET es un lenguaje de programación diseñado para crear un amplio número de aplicaciones que se ejecutan en .NET Framework. Es sencillo, moderno, proporciona seguridad de tipos y es orientado a objetos. Visual C# utiliza plantillas de proyecto, diseñadores, páginas de propiedades, asistentes de código y un modelo de objetos [15]. Además de otras características que lo hacen ser una opción excelente si se trata de utilización de patrones de diseño, metodologías ágiles y herramientas.

La clase UserControl del espacio de nombres System.Windows.Forms, proporciona la capacidad de crear controles que se pueden usar en varios lugares de una aplicación. Puede incluir todo el código necesario para la validación de datos frecuentes que se solicitan al usuario. El uso más interesante para aplicaciones enmarcadas en nuestro campo de acción es que por ejemplo, puede cargar previamente un ComboBox o un ListBox con elementos estáticos que se usan normalmente en casi todas las aplicaciones, como por ejemplo, países, regiones, ciudades, provincias y ubicaciones de usuarios etc.

Visual C# nos brinda también la posibilidad de usar el objeto DataSet de ADO.NET, es una representación de datos residente en memoria que proporciona un modelo de programación relacional coherente independientemente del origen de datos que contenga. Un DataSet es capaz de ordenar y restringir datos, así como establecer relaciones entre las tablas.

1.6.4 Flash.

Es una herramienta sencilla y fácil de usar, nos provee de una gran riqueza en materia de diseño para las pruebas neuropsicológicas. Utiliza como lenguaje de programación ActionScript, que en sus versiones más recientes tiene todas las ventajas del paradigma orientado a objetos. Es un software estándar de edición profesional para la creación de publicaciones Web y de escritorio de gran impacto [16]. Para utilizar Flash, no es necesario utilizar ActionScript, pero si se desea que los usuarios puedan interactuar, trabajar con objetos que no sean los incorporados en Flash (como por ejemplo, botones y clips de película) o convertir un archivo SWF en una experiencia de usuario más fiable, se tiene que utilizar este lenguaje.

1.6.5 Web Services.

Un servicio web o web service como lo indica su nombre en inglés, es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes y ejecutadas sobre cualquier plataforma pueden utilizar los servicios web para intercambiar datos en redes de computadoras como la red de redes o en una red local que es nuestro caso. La interoperabilidad se consigue mediante la adopción de estándares abiertos.

Los servicios web aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen. Fomentan los estándares y protocolos basados en texto que hacen más fácil acceder a su contenido y entender su funcionamiento. Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado. Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados. Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estandarizados [17].

1.6.6 PostgreSQL 8.1.

Como sistema de gestión de base de datos a utilizar se seleccionó el PostgreSQL en su versión 8.1. PostgreSQL es un potente gestor de Bases de Datos, es software libre, corre en un gran número de sistemas operativos, siendo multiplataforma. Tiene todas las características de los SGBD modernos, como

son las llaves foráneas, vistas, procedimientos almacenados, varios tipos modernos de datos, etc.; posee una característica importante, que es la realización de múltiples “backups” o salvadas de la información.

Agrega madurez y performance (funcionamiento), entregando manejabilidad comparable a los más conocidos sistemas de bases de datos. Además de sintaxis compatible, extendida e interfaces que hacen más fácil para nuevos DBAs (Diagramas de base de datos) usar todas sus características avanzadas.

PostgreSQL cuenta además con una característica llamada Write - Ahead Logging (WAL) que permite asegurar la atomicidad y durabilidad de la información. Dos de las características ACID de las bases de datos. Usando WAL las modificaciones son escritas primero a un log o bitácora antes de ser aplicadas a la base de datos. Esto garantiza que las operaciones de deshacer o rehacer se puedan ejecutar sin problemas y garantizar la integridad de los datos [18]. Además de garantizar las otras dos características ACID, consistencia e integridad.

Teniendo en cuenta que trabajaremos en el área del software para la salud, y que la información que se manipulará será sensible y relevante, por ser información relativa a las personas; así como a los estudios y diagnósticos asociados a éstas, conviene entonces, escoger un SGBD (Sistema de Gestión de Base de Datos) que cumpla con las funciones de respaldo, y PostgreSQL deviene candidato ideal, además de que no encareceríamos el sistema final usando un gestor como Oracle o SqlServer.

1.6.7 Enterprise Architect 6.5.

Enterprise Architect es una herramienta CASE que combina el poder de la última especificación UML 2.1 con alto rendimiento, interfaz intuitiva para traer modelado avanzado al escritorio y para el equipo completo de desarrollo e implementación. Con un gran conjunto de características y un valor sin igual económicamente, EA puede equipar a un equipo entero, incluyendo analistas, evaluadores, administradores de proyectos, personal del control de calidad, equipo de desarrollo y más, por una fracción del costo de algunos productos competitivos. Es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad, aspecto de vital importancia.

EA se encarga de administrar la complejidad con herramientas para rastrear las dependencias, brinda soporte para modelos muy grandes, control de versiones con proveedores CVS, SCC, subversión y TFS. Líneas Base por cada punto del tiempo e interfaz intuitiva y de alto rendimiento con vista de proyecto como un "explorador". Provee una generación poderosa de documentos y herramientas de reporte con un editor de plantilla completo WYSIWYG. Soporta generación e ingeniería inversa de código fuente para muchos lenguajes populares, incluyendo C++, C#, Java, Delphi, VB.Net, Visual Basic, PHP y ActionScript por solo mencionar algunos. También posee Add-ins gratis para CORBA y Python disponibles.

Soporta además transformaciones de arquitectura avanzada dirigida por Modelos (MDA) usando plantillas de transformaciones de desarrollo y fáciles de usar. Exporta modelos de otras herramientas Case mediante XMI desde 1.1 hasta 2.1 [19].

1.6.8 RUP (Proceso Unificado de Rational).

Definidas las herramientas que darán soporte a la base teórica es necesario definir la metodología de ingeniería de software que guiará el proceso de automatización, en este caso se optó por el Proceso Unificado de Desarrollo. Constituye en efecto un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema informático, es un marco de trabajo genérico especializado en una gran variedad de sistemas de software, diferentes áreas de aplicación, tipos de organización, diferentes tamaños de proyecto, etc.

Lo esencial para la construcción del software es conocer lo que los usuarios necesitan. Estas necesidades serán los futuros casos de uso del sistema que guiarán todo el proceso de desarrollo. Teniendo en cuenta que los deseos o decisiones de los usuarios puede cambiar en cualquier momento del ciclo de vida y el proceso de construcción debe ser flexible frente a esta situación. Por supuesto, el proceso debe tener diferentes vistas que lo guíen y evolucionen es por ello que el Proceso Unificado tiene como características ser iterativo e incremental, centrado en la arquitectura y guiado por casos de uso. Además, está basado en componentes, lo cual quiere decir que el sistema está formado por componentes interconectados a través de interfaces bien definidas.

El Proceso Unificado utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, **UML**) para construir la mayoría de los artefactos del sistema. En general el Proceso Unificado es una solución del problema del software [20].

1.6.9 UML (Lenguaje Unificado de Modelado).

UML es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, la versión 2.0 que es la utilizada en la modelación de nuestro sistema está apoyada en gran medida por la OMG (Object Management Group por sus siglas en inglés). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. Entre sus objetivos principales están:

- ✓ Hacer el lenguaje de modelado mucho más extensible de lo que era.
- ✓ Permitir la validación y ejecución de modelos creados mediante el UML.

Es interesante destacar que el UML 2.0 puede definirse a sí mismo. Es decir, su estructura y organización es modelable utilizando el propio UML 2.0; de esta manera, se da un ejemplo de utilización del UML en un dominio distinto al del desarrollo de software [21].

En el desarrollo de un sistema, la elección de las herramientas y la metodología a utilizar forman un paso vital, que nos traslada desde la etapa de investigación y familiarización con el problema hasta el mismo proceso de desarrollo del software. En este capítulo se realizó una investigación del estado del arte a nivel mundial y nacional, también se fundamentó la tecnología utilizada en el desarrollo de la solución propuesta al problema planteado así como la metodología empleada.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En este capítulo se abordan las características del sistema propuesto, se describen el objeto de automatización y la propuesta del sistema, se analizan además el modelo de negocio que incluye la presentación de los actores y trabajadores, los diagramas de casos de uso del negocio y los de actividades. Se presentan también el diagrama de clases correspondiente al modelo de objetos y se describen los requerimientos del sistema que constituyen la base para la realización del modelo de casos de uso.

2.1 Objeto de automatización.

Nuestros procesos del negocio serán tomados antes de la existencia del Diana en sus dos versiones y después de la problemática de la solución informática debido a la incompatibilidad con las nuevas tecnologías, serían los mismos procesos del negocio por la inexistencia de un software capaz de cumplir con las funciones requeridas en materia de diagnóstico neuropsicológico antes cumplidas por este sistema. Funciones surgidas en los momentos actuales debido al agigantado crecimiento de la tecnologías, así como de las capacidades humanas. A continuación los procesos antes mencionados.

✓ Consultar Pacientes.

Se recogen los datos de los pacientes pero no se tiene en cuenta si el paciente ya estuvo admitido en la institución previamente posibilitando así que los datos se repitan, se seleccionan las pruebas neuropsicológicas que van a ser aplicadas que están en formato duro y el paciente debe realizarlas a lápiz, corriendo el riesgo de no incentivarlos en su realización, además de convertirse en largas y tediosas.

✓ Entrega resultados.

Una vez que hayan sido aplicadas las pruebas neuropsicológicas, se analizan los resultados y se escribe un reporte final que contiene los datos del especialista, nombre del estudio neuropsicológico, datos del paciente y resultados de las pruebas. Se le entrega al paciente los datos del diagnóstico, tratamiento y observación.

✓ Registro de la Información.

Proceso en el que se archivan todos los estudios realizados, los mismos contienen los datos de las consultas, se corre el riesgo de que el estudio este archivado y se cree un nuevo registro con el mismo nombre.

2.2 Propuesta de Sistema.

De acuerdo con la investigación realizada, se plantea la tarea de implementar un sistema que gestione las pruebas neuropsicológicas de forma centralizada, desarrollando las actividades relacionadas con el proceso de consulta de los pacientes.

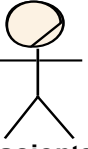
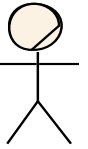
A continuación, mencionamos las principales actividades: capturar los datos del paciente, en caso de que el paciente ya esté en el sistema se busca para no correr el riesgo de repetirlo; en caso de que el especialista no se encuentre almacenado en el sistema, se capturan sus datos; se seleccionan las pruebas neuropsicológicas a ser aplicadas; el sistema permitirá la creación de paquetes de pruebas, estos contienen pruebas de un dominio determinado o diversos dominios. El sistema permitirá gestionar la información generada y manejada por el proceso antes mencionado.

Con la automatización de estos procesos, no sólo el paciente es beneficiado por el negocio, el especialista también obtiene los beneficios que reporta la implementación y puesta en marcha de un sistema de este tipo. Para lograr lo antes mencionado, hemos obtenido los requisitos de acuerdo a las funcionalidades pedidas por nuestros clientes del “Centro de Neurociencias de Cuba”.

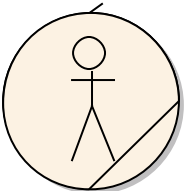
Nos trazamos como objetivo concreto, desarrollar un producto donde su base fundamental, las pruebas neuropsicológicas, cumplan con los estándares internacionales.

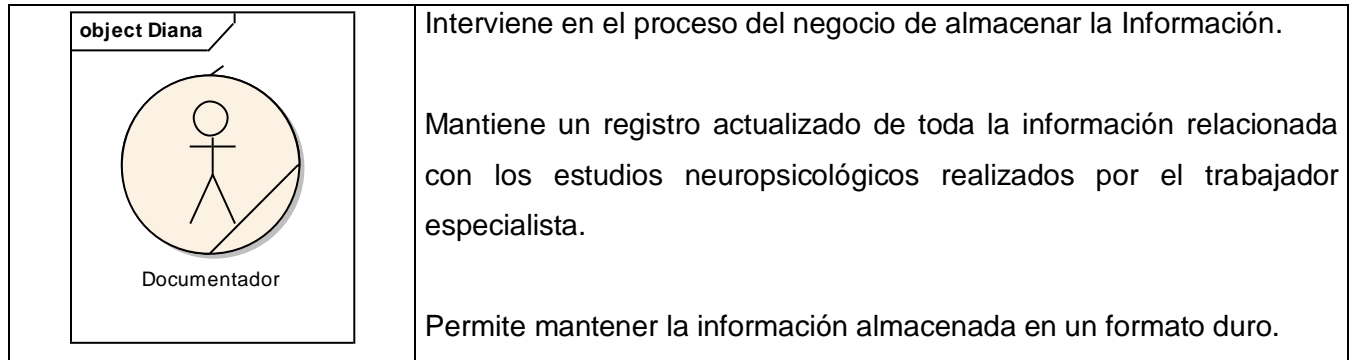
2.3 Modelo del Negocio.

Los actores del negocio son las personas o sistemas que se benefician directamente con los procesos del negocio y obtienen un resultado de valor de los mismos. Se muestran a continuación:

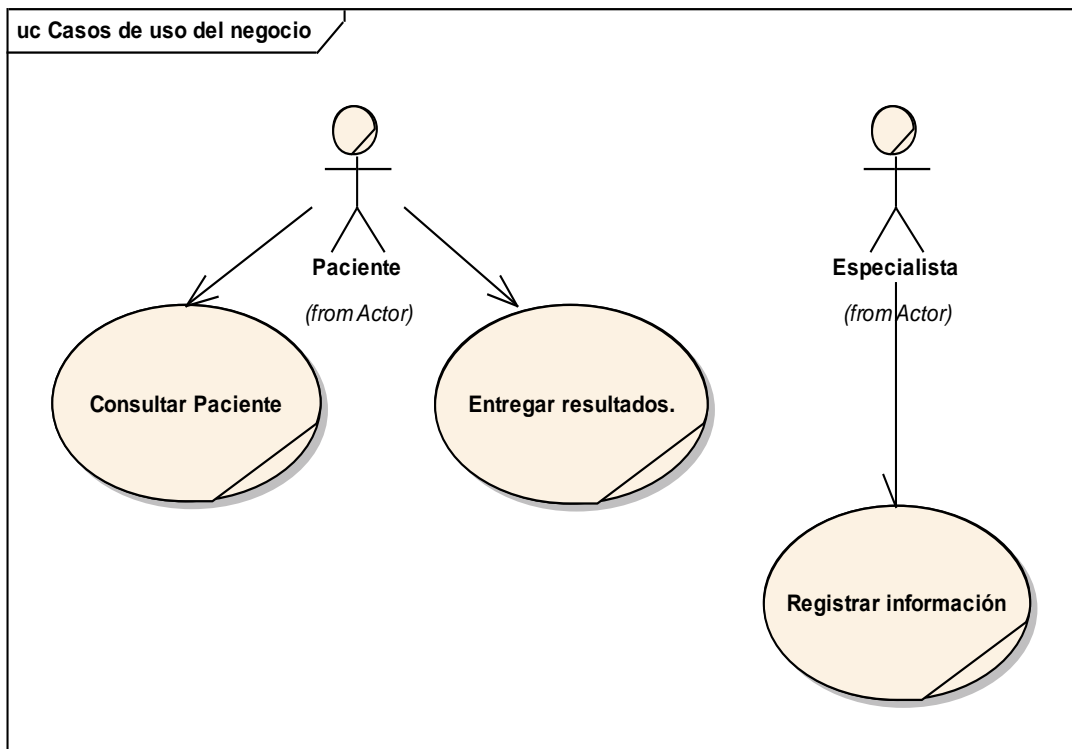
Actores del negocio.	Justificación.
<div data-bbox="224 348 464 657" style="border: 1px solid black; padding: 5px;"> <p data-bbox="232 352 386 380">uc Actor</p>  <p data-bbox="280 575 399 602">Paciente</p> </div>	<p data-bbox="513 348 1024 375">Interviene en los procesos del negocio.</p> <p data-bbox="513 443 1498 527">Se beneficia con los servicios brindados en las entidades encargadas de los procesos neuropsicológicos.</p>
<div data-bbox="224 667 464 982" style="border: 1px solid black; padding: 5px;"> <p data-bbox="232 672 386 699">uc Actor</p>  <p data-bbox="264 894 415 921">Especialista</p> </div>	<p data-bbox="513 667 1024 695">Interviene en los procesos del negocio.</p> <p data-bbox="513 762 1243 789">Se beneficia con el proceso de Registro de Información.</p> <p data-bbox="513 863 1498 947">Es la principal figura en los procesos del negocio, gracias a la iteración de este actor es que se pueden realizar los mismos.</p>

Los trabajadores del negocio son las personas o sistemas que rara vez reciben un beneficio de él, se encuentran involucrados en uno o más procesos del negocio, participan en ellos y son posibles candidatos a convertirse en actores del sistema. Nosotros tenemos un caso muy puntual que es el especialista, este se comporta como un actor del negocio y a la vez como trabajador del mismo.

Trabajadores del negocio.	Justificación.
<div data-bbox="240 1400 521 1724" style="border: 1px solid black; padding: 5px;"> <p data-bbox="248 1404 402 1432">object Diana</p>  <p data-bbox="321 1650 440 1677">Especialista</p> </div>	<p data-bbox="586 1400 1498 1484">Interviene en el proceso del negocio de consultar pacientes y en la entrega de los resultados.</p> <p data-bbox="586 1547 1498 1682">Estos procesos son de suma importancia ya que son la base para la creación de un sistema neuropsicológico automatizado, el especialista realiza una serie de actividades de suma importancia.</p>



2.4 Diagrama de Casos de uso del negocio.



2.5 Diagramas de actividades de los casos de uso del negocio.

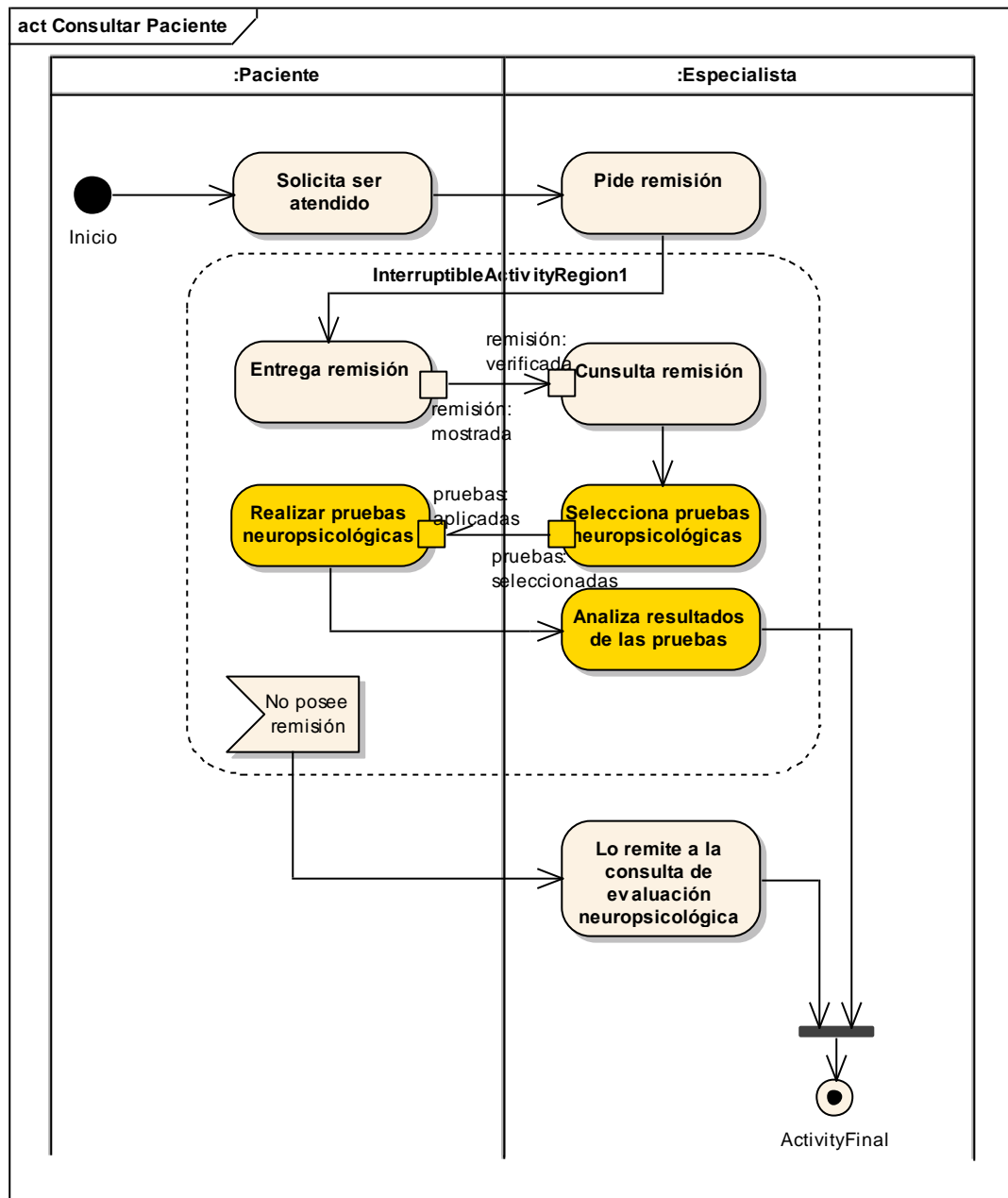


Diagrama de Actividades – CUN Consultar Pacientes.

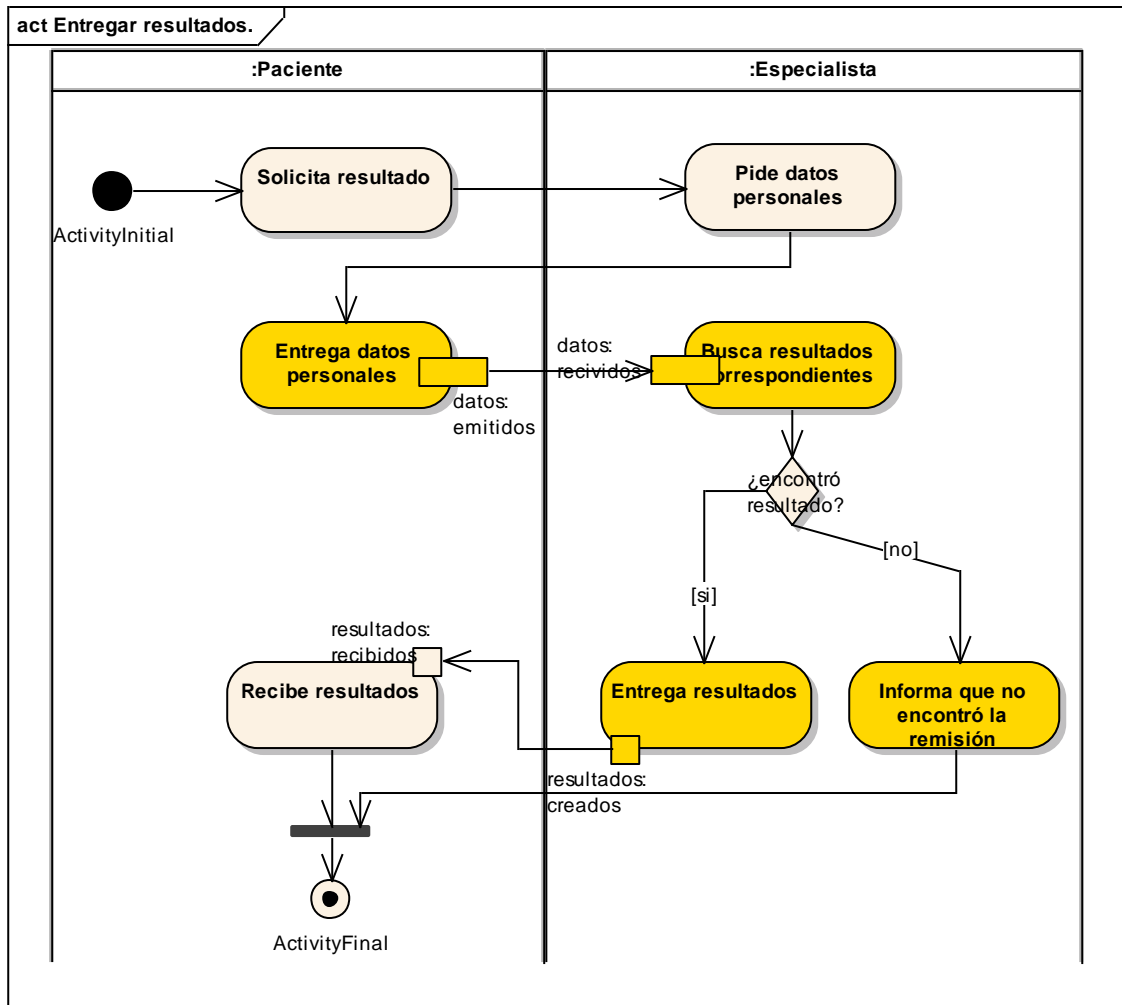


Diagrama de Actividades – CUN Entregar Resultados.

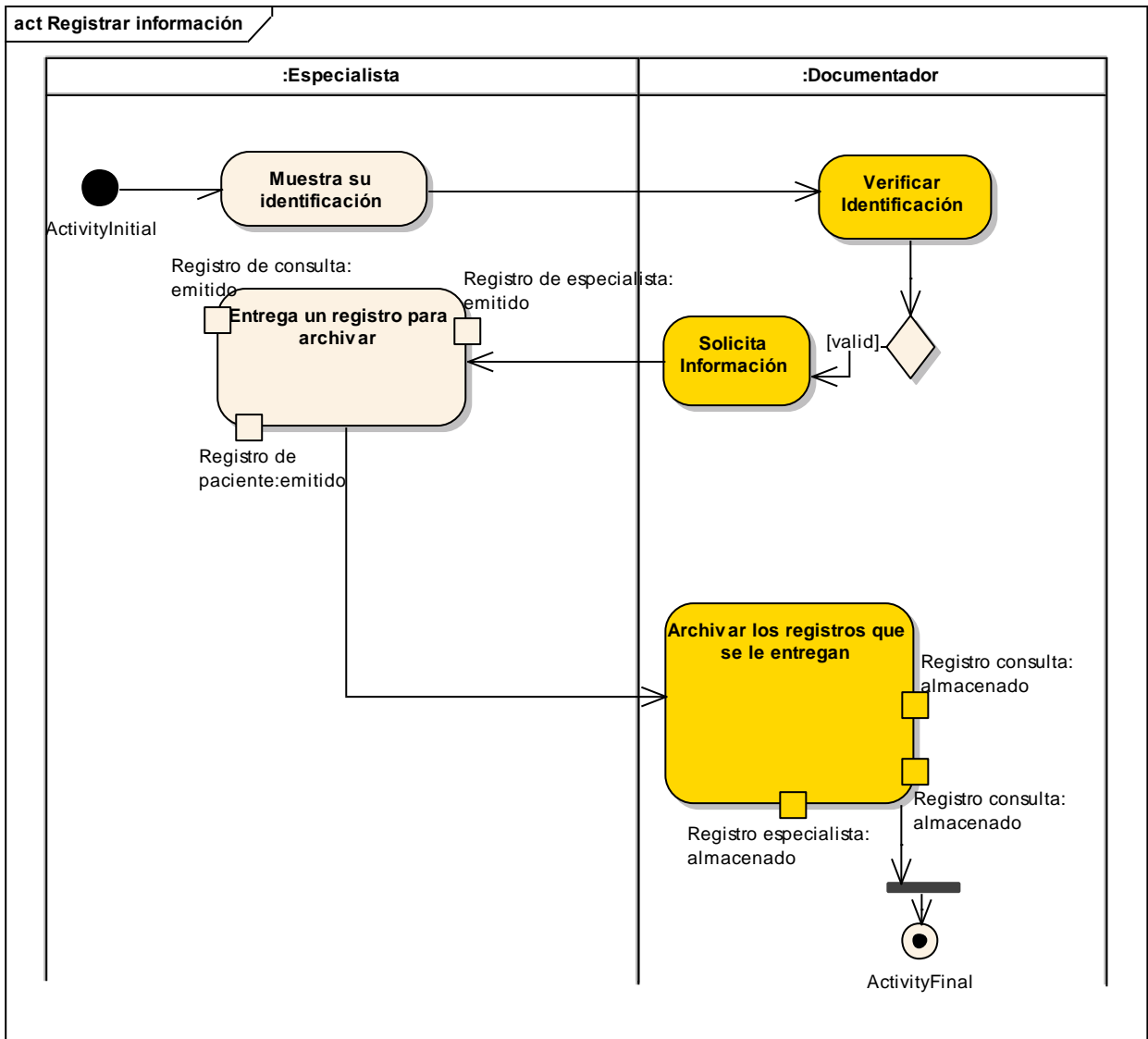
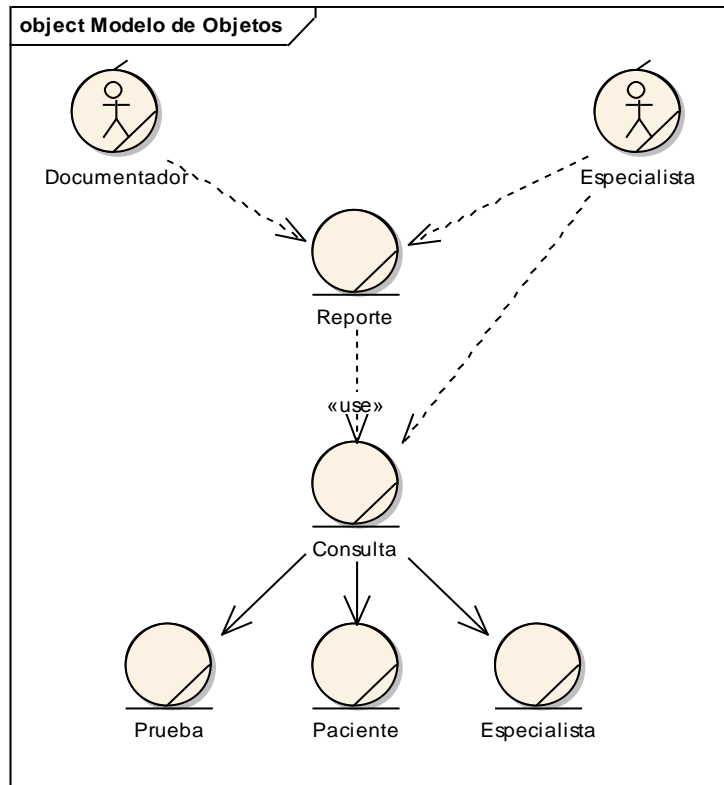


Diagrama de Actividades - CUN Registrar Información.

2.6 Diagrama de clases del modelo de objetos.

El modelo de objetos del negocio expresa las relaciones que existen entre los trabajadores del negocio y las entidades del mismo.



2.7 Especificación de los requerimientos del software.

Para una mejor comprensión del sistema, lo hemos dividido en dos subsistemas: el Cliente Diana, que va a contener la aplicación cliente junto a un servicio que se ejecuta en el mismo ordenador y el repositorio de información, que contiene un servicio web y la base datos.

2.7.1 Requerimientos Funcionales.

Los requerimientos funcionales del sistema no son más que las funcionalidades que este debe cumplir para satisfacer a sus clientes, se encuentran agrupados en dos paquetes:

- ✓ Cliente Diana.

Requerimientos asociados a la aplicación cliente y el servicio. Este se divide en subpaquetes donde se exponen los principales servicios del cliente y los requisitos de gestión.

- ✓ Repositorio de Información.

Requerimientos asociados a el servicio web y la base datos.

A continuación, se expone la distribución en paquetes de los requerimientos del sistema. Para una vista detallada de los mismos ver ANEXO 1: DIAGRAMAS DE REQUERIMIENTOS.

Cliente Diana.

Principales servicios del Cliente Diana.

Requisito	Descripción
REQ: 1.001 Crear especialista.	Son capturados todos los datos del especialista, salvados y cargados por el sistema.
REQ: 1.002 Crear estudio neuropsicológico.	Se capturan los datos del estudio neuropsicológico, son salvados en el sistema.
REQ: 1.03 Crear paquete de pruebas neuropsicológicas.	<p>Los paquetes de pruebas neuropsicológicas contienen pruebas de diversos dominios de evaluación cognitiva o del mismo dominio.</p> <p>Se pueden crear en el momento de aplicar las pruebas o en el momento que estime conveniente el especialista.</p> <p>Son creados con el fin de aplicarlos a los pacientes.</p>
REQ1.004-Configurar pruebas neuropsicológicas.	<p>Se configuran las pruebas neuropsicológicas, se salvan y cargan por el sistema.</p> <p>Esta configuración no comprende las pruebas que están en paquetes.</p>
REQ1.005-Enviar información para la base de datos	<p>Se configura la conexión para mandar los datos generados por el sistema.</p> <p>Se guarda configuración en un fichero.</p> <p>Se levanta el servicio, toma datos de la configuración y establece conexión con la base de datos mediante un servicio</p>

	web.
REQ1.006-Ver estado de la conexión.	El estado de la conexión a la base de datos.
REQ1.007-Exportar datos.	Exportar datos de los estudios realizados fuera del Cliente Diana, estos datos están en un fichero de texto.
REQ1.008-Importar datos.	Importar datos de los estudios realizados por el Cliente Diana. Se exportan en un fichero de texto.
REQ1.021-Visualizar Consultas	Da una vista de las consultas que han sido completadas.
I) Paquete Consultar Pacientes.	Contiene los requisitos para la consulta de pacientes.
II) Paquete generar reportes.	Contiene los requisitos para la generación de reportes.

Para mayor comprensión del Diagrama de Requerimientos del paquete Principales Servicios del Cliente Diana ver en ANEXO 1: PRINCIPALES SERVICIOS DEL CLIENTE DIANA.

Consultar Pacientes.

Requisito	Descripción
REQ1.009-Capturar datos del paciente.	Se capturan los datos del paciente en caso de que el especialista decida no aplicarle las pruebas como anónimo.
REQ1.010-Buscar Paciente.	Se busca el paciente en caso de que ya haya sido consultado con anterioridad por el Cliente Diana.
REQ1.011-Cargar las pruebas neuropsicológicas.	El cliente carga los ficheros “.swf” que son las pruebas neuropsicológicas.
REQ1.012-Cargar configuración de forma automática.	Se carga automáticamente la configuración que se le dio a la prueba. Puede cargar la configuración predeterminada.
REQ1.013-Visualizar pruebas neuropsicológicas.	Visualiza las pruebas neuropsicológicas para que puedan ser realizadas por el paciente.
REQ1.014-Guardar los datos generales de la consulta.	Al terminar la consulta del paciente se guardan los datos generales de la misma.

Para mayor comprensión del Diagrama de requisitos del paquete Consultar Pacientes ver en ANEXO 1: CONSULTAR PACIENTES.

Generar Reportes

Requisito	Descripción
REQ1.015-Mostrar las consultas almacenadas que aún no han sido reportadas.	Muestra todas las consultas neuropsicológicas que aún no han sido completadas. Se encuentran almacenadas en el sistema.
REQ1.016-Mostrar estudios neuropsicológicos almacenados en el sistema.	Muestra todos los estudios neuropsicológicos que se encuentren almacenados en el sistema.
REQ1.017-Insertar Diagnóstico.	Valoración del especialista según los resultados de las pruebas neuropsicológicas aplicadas en la consulta.
REQ1.018-Insertar Tratamiento.	Tratamiento emitido por el especialista según los resultados de las pruebas neuropsicológicas aplicadas en la consulta.
REQ1.019-Insertar Observaciones.	Principales observaciones descritas por el especialista de la consulta realizada.
REQ1.020-Mostrar resultado de las pruebas realizadas en la consulta.	Se muestran todos los resultados de las pruebas neuropsicológicas que han sido aplicadas en cada consulta.
REQ1.021-Salvar el reporte.	Se salva el reporte luego de ser emitido.

Para mayor comprensión del Diagrama de requisitos del paquete Generar Reportes ver en ANEXO 1: GENERAR REPORTES.

Gestión.

Requisito	Descripción
REQ2.001-Modificar datos del Paciente.	Se modifican los datos del paciente, también incluye eliminar paciente.
REQ2.002-Modificar datos del paquete.	Se modifican los datos del paquete, también incluye eliminar paquete.
REQ2.003-Modificar datos del estudio neuropsicológico.	Se modifican los datos del estudio neuropsicológico, también incluye eliminar estudio.

REQ2.004-Modificar datos del especialista.	Se modifican los datos del especialista, también incluye eliminar especialista.
REQ2.005-Eliminar estudio neuropsicológico.	Elimina un estudio que se encuentre almacenado en el sistema.
REQ2.006-Eliminar Paquete de pruebas neuropsicológicas.	Elimina un paquete de pruebas neuropsicológicas que se encuentre almacenado en el sistema.
REQ2.007-Eliminar especialista.	Elimina un especialista que se encuentre almacenado en el sistema.
REQ2.008-Eliminar consulta.	Elimina una consulta que se encuentre almacenada en el sistema.
REQ2.009-Eliminar paciente.	Elimina un paciente que se encuentre almacenado en el sistema.

Para mayor comprensión del Diagrama de requisitos del paquete Gestión ver en ANEXO 1: GESTIÓN. Repositorio de Información.

Requisito	Descripción
REQ3.001-Recepcionar la información	Se recepciona toda la información enviada por el servicio del Cliente Diana.
REQ3.002- Verificar procedencia del envío.	Verificar que la procedencia de los envíos sea desde un cliente autorizado.
REQ3.003- Almacenar la información en la Base de Datos.	El servicio web almacena la información en la base de datos, lográndose la persistencia de los mismos.

Para mayor comprensión del Diagrama de Requerimientos del paquete Repositorio de Información ver en ANEXO 1: REPOSITORIO DE INFORMACIÓN.

Luego de hacer un análisis extendido de los requerimientos funcionales que van a llevar a la implementación de nuestro sistema, listamos los requisitos no funcionales que este debe cumplir.

2.7.2 Requerimientos no Funcionales.

Usabilidad.

El sistema debe ser lo más atractivo posible y debe facilitar el trabajo de los especialistas, los futuros usuarios no tienen que poseer un nivel avanzado para poder usarlo.

Legales.

El sistema y toda la documentación generada con el mismo pertenecen al Grupo de Procesamiento de Imágenes y Señales de la Universidad de la Ciencias Informáticas y al Centro de Neurociencias de Cuba.

Interfaz.

La aplicación propuesta poseerá una interfaz amigable y cómoda para los usuarios a quien va dirigida.

Software.

La solución informática propuesta requiere del uso del NET Framework 2.0 o superior, sistema de gestión de base de datos PostgreSQL 8.1 o superior y IIS 5.0 o superior.

Hardware.

La aplicación propuesta tendrá dependencias de hardware como: 256Mb de memoria RAM (mínimo), CPU Pentium IV 2.4GHz (recomendado).

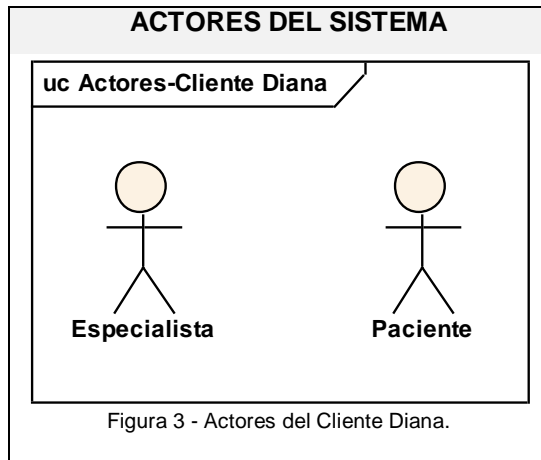
Persistencia.

Debe mantenerse la integridad de los datos almacenados (consultas).

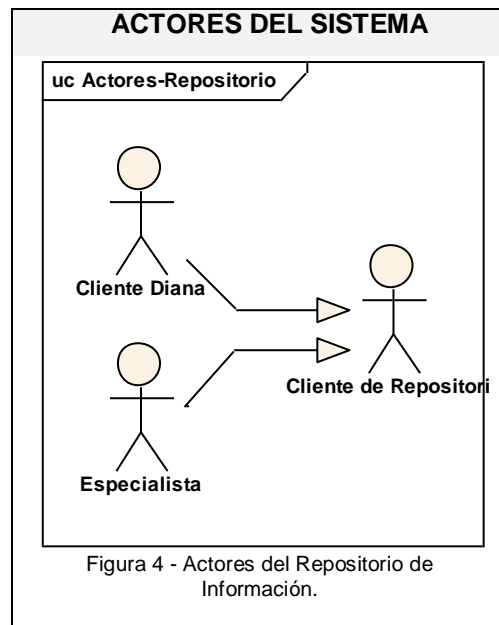
2.8 Casos de Uso del Diana.

2.8.1 Actores del sistema a automatizar.

Actores del Cliente Diana.



Actores del Repositorio de Información.



Actores	Justificación
Especialista.	Es el responsable de cumplir la mayoría de las funciones desde que el paciente llega a la consulta, es consultado, hasta que sale con los resultados.

	<p>También es una generalización de ClienteRepositorio pues el actor es el encargado de iniciar el caso de uso Configurar Conexión al repositorio donde se especifica como va a ser el plan de envío.</p> <p>Envía información para el repositorio mediante la generalización ClienteDiana.</p> <p>Cumple el rol fundamental dentro del sistema.</p>
Paciente.	<p>Inicia el caso de uso Realizar Pruebas Neuropsicológicas.</p> <p>Es el responsable de la realización de estas pruebas.</p>
Cliente Diana.	<p>El subsistema Cliente Diana va a ser un actor para el repositorio de información mediante la generalización ClienteRepositorio.</p> <p>Sólo inicia el caso de uso Almacenar información en la base de datos cuando en el caso de uso incluido Plan de Envío, tiene que conectarse a una hora determinada.</p>

2.8.2 Casos de Uso.

CU1	Consultar pacientes.
Actor	Especialista.
Descripción	Este caso de uso se inicia cuando el especialista decide consultar a los pacientes, agregando mediante la captura los datos del paciente en caso de no estar ya en el

	sistema.
Referencia	REQ1.009, REQ1.010, REQ1.011, REQ1.012, REQ1.013, REQ1.014.

CU2	Insertar datos del especialista en la consulta.
Actor	Especialista.
Descripción	Este caso de uso se inicia cuando el especialista que no está insertado en el sistema decide introducir sus datos en el momento de consultar un paciente.
Referencia	REQ1.001.

CU3	Crear Paquete.
Actor	Especialista.
Descripción	Este caso de uso se inicia cuando el especialista decide crear un paquete de pruebas para aplicarlo en las consultas neuropsicológicas.
Referencia	REQ1.003.

CU4	Crear Estudio Neuropsicológico.
Actor	Especialista.
Descripción	Este caso de uso se inicia cuando el especialista introduce los datos del estudio neuropsicológico que esta realizando.
Referencia	REQ1.002.

CU5	Capturar datos del paciente.
Actor	Especialista.
Descripción	Este caso de uso es extendido del caso de uso Consultar Pacientes y de Generar Reportes.
Referencia	REQ1.009.

CU6	Buscar Paciente.
Actor	Especialista.
Descripción	Este caso de uso es extendido del caso de uso Capturar Datos del Paciente, se inicia

	cuando el especialista decide buscar el paciente en el sistema.
Referencia	REQ1.010.

CU7	Configurar Pruebas Neuropsicológicas.
Actor	Especialista.
Descripción	Este caso de uso es iniciado por el Especialista, este configura las pruebas neuropsicológicas y salva la configuración en el sistema.
Referencia	REQ1.004.

CU8	Configurar Conexión al Repositorio.
Actor	Especialista.
Descripción	Este caso de uso se inicia cuando el Especialista desea configurar la conexión al repositorio, incluye el plan de envío, es cuando el Especialista configura si va a mandar la información de forma manual o a una hora determinada.
Referencia	REQ1.005, REQ1.006.

CU9	Realizar pruebas neuropsicológicas.
Actor	Paciente.
Descripción	Este caso de uso se inicia cuando el Paciente se encuentra realizando las pruebas neuropsicológicas. La prueba es iniciada por el sistema cuando el especialista lo indica.
Referencia	REQ1.013.

CU10	Generar Reportes.
Actor	Especialista.
Descripción	Este caso de uso es iniciado por el Especialista, el sistema muestra las consultas realizadas que no han sido reportadas, muestra los estudios neuropsicológicos realizados, además de los resultados de las pruebas.

	<p>Se capturan los datos del paciente en caso de no haberseles tomado durante la consulta.</p> <p>Y se describe un diagnóstico, tratamiento y observaciones con respecto a la consulta, se salva la consulta completada.</p>
Referencia	REQ1.015, REQ1.016, REQ1.017, REQ1.018, REQ1.019, REQ1.020, REQ1.021.

CU11	Visualizar Consultas.
Actor	Especialista.
Descripción	Este caso de uso se inicia por el Especialista, este visualiza las consultas que han sido realizadas.
Referencia	REQ1.022

CU12	Importar Datos.
Actor	Especialista.
Descripción	Este caso de uso se inicia por el Especialista, importa datos en un formato de diana para que este pueda transportar la información de las consultas desde donde desee.
Referencia	REQ1.008.

CU13	Exportar Datos.
Actor	Especialista.
Descripción	Este caso de uso se inicia por el especialista, exporta datos en un de diana para el Cliente Diana.
Referencia	REQ1.007..

CU14	Editar especialista
Actor	Especialista.
Descripción	<p>Este caso de uso se inicia por el Especialista, edita los datos del especialista.</p> <p>También puede eliminar el especialista del sistema.</p>

Referencia	REQ2.004, REQ2.007.
------------	---------------------

CU15	Editar paciente
Actor	Especialista.
Descripción	Este caso de uso se inicia por el especialista, edita los datos del paciente. También puede eliminar el paciente del sistema.
Referencia	REQ2.001, REQ2.009.

CU16	Editar paquete
Actor	Especialista.
Descripción	Este caso de uso se inicia por el especialista, edita los datos del paquete. También puede eliminar el paquete del sistema.
Referencia	REQ2.002, REQ2.006.

CU17	Editar estudio neuropsicológico.
Actor	Especialista.
Descripción	Este caso de uso se inicia por el especialista, edita los datos del estudio neuropsicológico. También puede eliminar el estudio neuropsicológico del sistema.
Referencia	REQ2.003, REQ2.005.

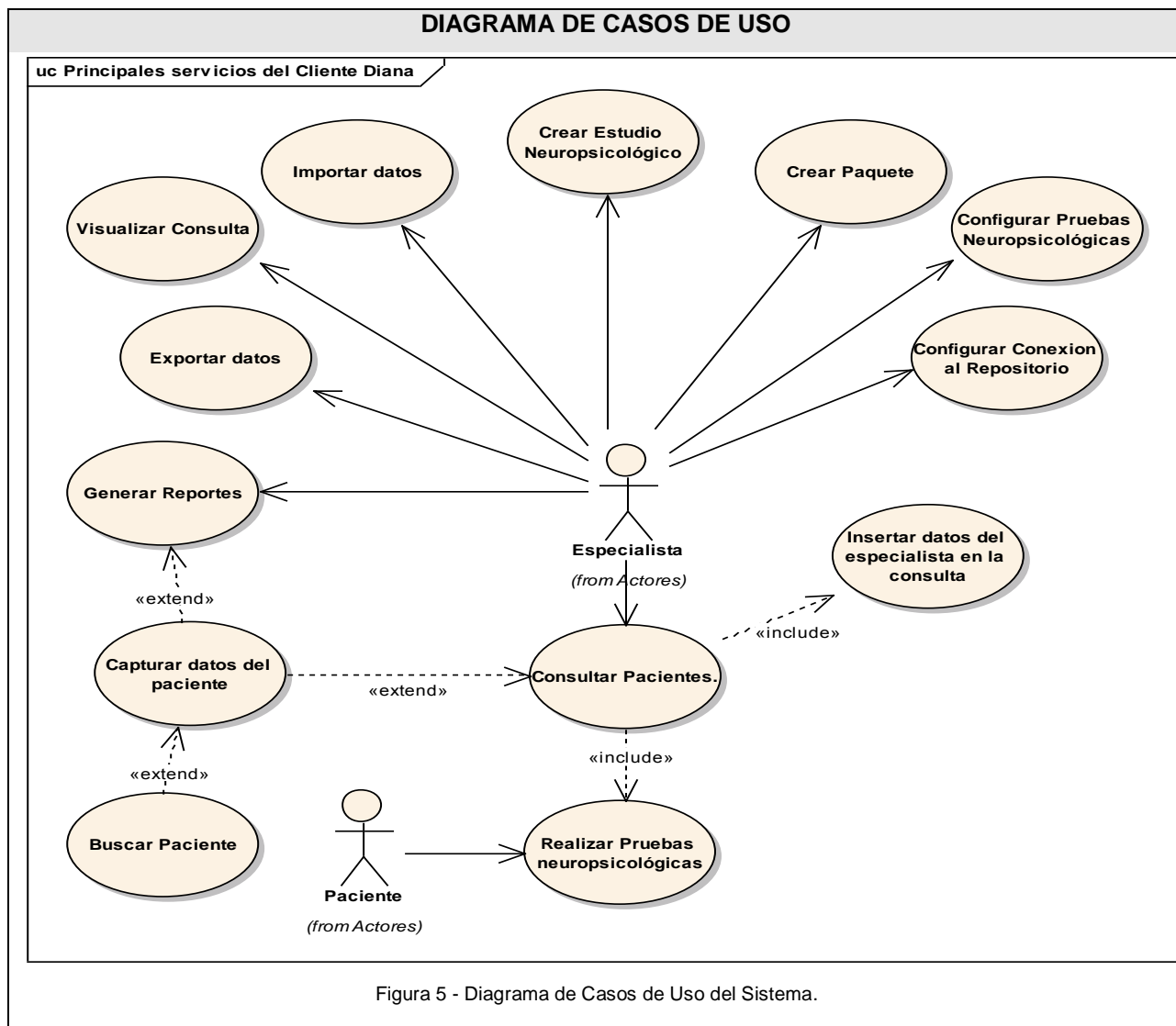
CU18	Almacenar información en la Base de datos.
Actor	Cliente de Repositorio.
Descripción	Este caso de uso es iniciado por el actor Cliente de Repositorio, este actor es una generalización del especialista y del sistema Cliente Diana que son los encargados de enviar información para el repositorio.

	Almacena información en la base de datos.
Referencia	REQ3.001, REQ3.002, REQ3.003.

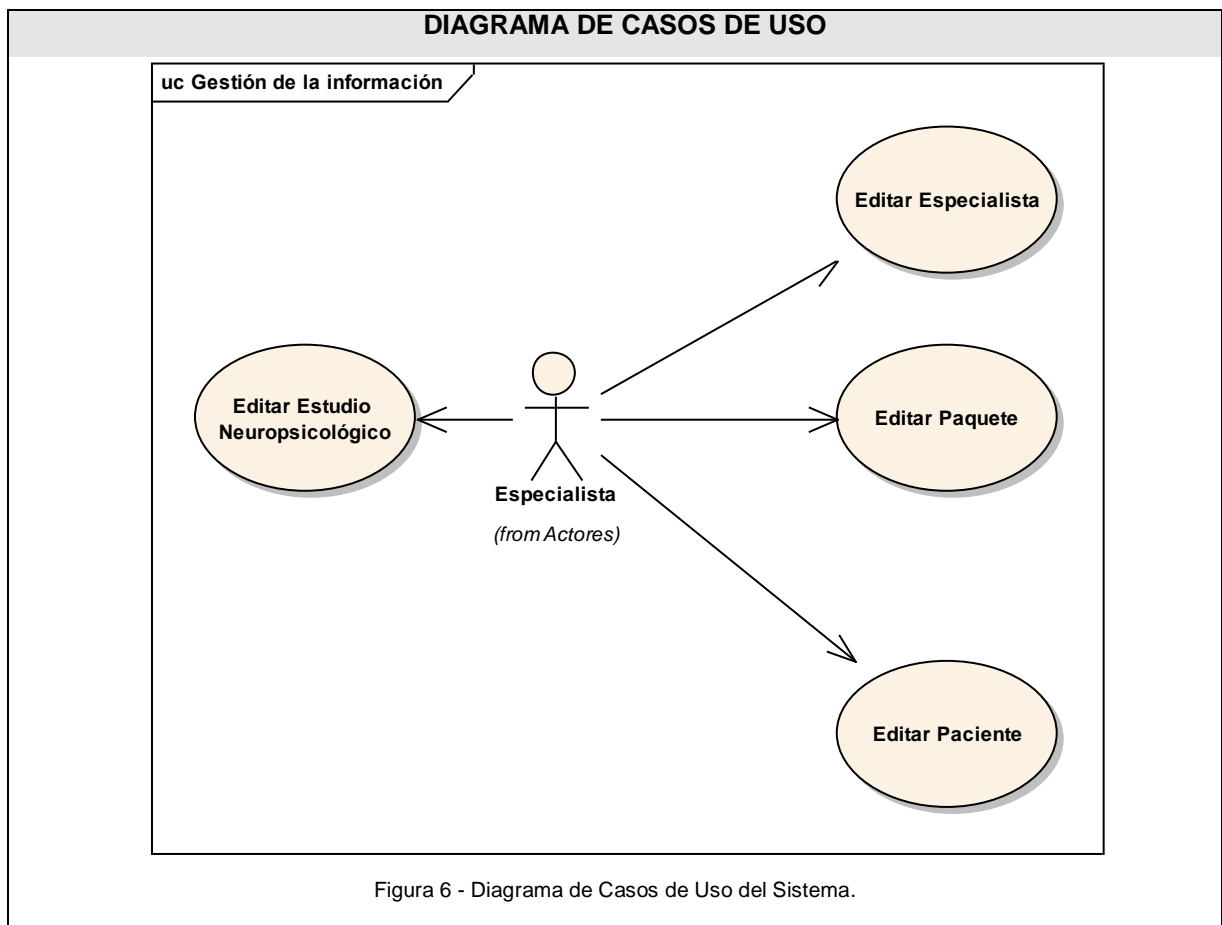
2.8.3 Diagrama de Casos de Uso.

Cliente Diana

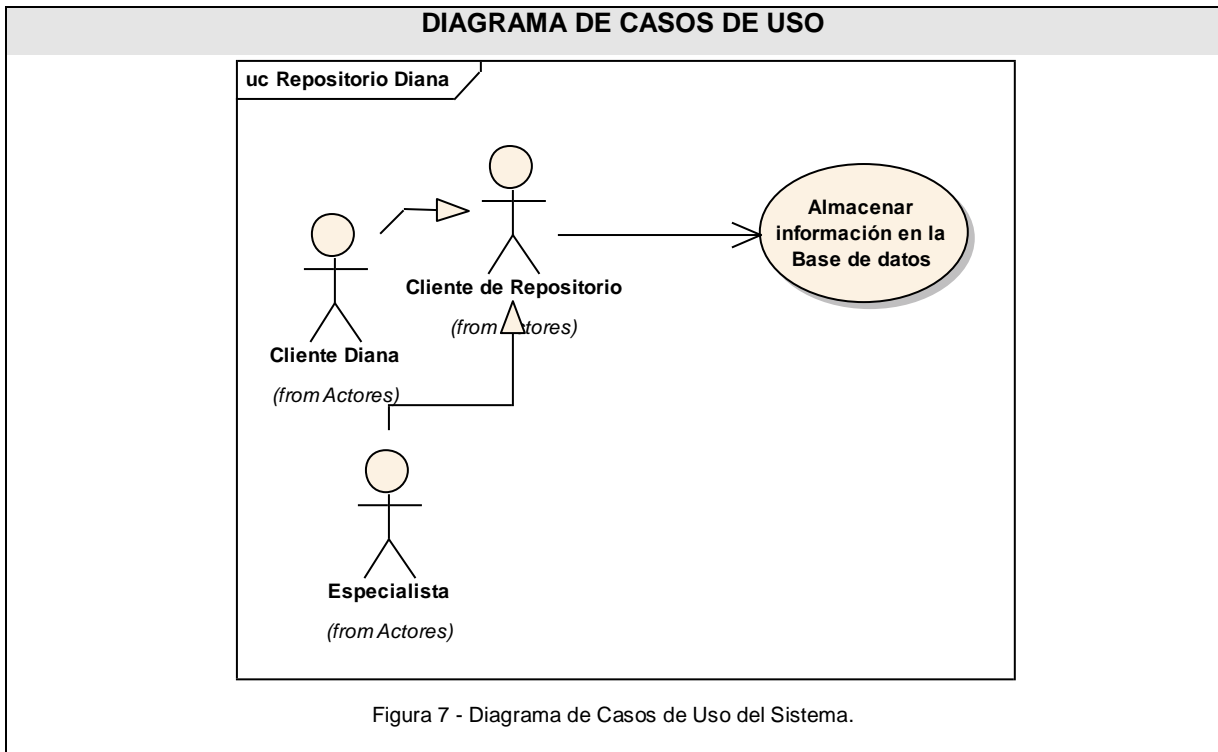
Principales servicios del Cliente Diana.



Gestión en el Cliente Diana.



Repositorio



2.8.4 Planificación de los casos de uso por ciclos de desarrollo.

Ciclo 1:

Cód.	Nombre de caso de uso	Paquete	Justificación de la selección.
CU1	Consultar pacientes.	Cliente Diana.	Caso de uso crítico para la arquitectura de nuestro sistema. Es el que regula la mayoría de los procesos que se llevan a cabo en el sistema para la aplicación de las pruebas.
CU2	Insertar datos del especialista en la consulta.	Cliente Diana.	Caso de uso incluido en el CU1. Caso de uso con una gran importancia en la

			<p>interacción del actor especialista con el sistema a la hora de realizar alguna consulta.</p> <p>Una vez que el especialista esté almacenado en el sistema es que se puede proceder a consultar.</p>
CU8	Configurar conexión al repositorio.	Cliente Diana.	<p>Caso de uso crítico en la arquitectura de nuestro sistema.</p> <p>Es el caso de uso que permite la interacción del Cliente Diana con el repositorio de información.</p>
CU7	Configurar Pruebas Neuropsicológicas.	Cliente Diana	Caso de uso que permite la configuración de las pruebas neuropsicológicas.
CU10	Generar Reportes.	Cliente Diana.	<p>Esencial en la construcción de nuestro sistema.</p> <p>Completa las consultas realizadas, creando un reporte general sobre la misma.</p>
CU5	Capturar Datos del Paciente.	Cliente Diana.	El caso de uso es extendido de CU10 y CU1.
CU9	Realizar pruebas neuropsicológicas.	Cliente Diana.	Caso de uso orientado al paciente, es el que le permite la interacción con las pruebas neuropsicológicas.
CU18	Almacenar información en la base de datos.	Repositorio de Información.	<p>Caso de uso crítico en la arquitectura de nuestro sistema.</p> <p>Es el encargado de recibir la información enviada desde el Cliente Diana y almacenarla en la base de datos.</p>

Ciclo 2:

Cód.	Nombre de caso de uso	Paquete	Justificación de la selección.
CU3	Crear paquete.	Cliente Diana.	Se crea el paquete para más comodidad para el especialista al aplicar las pruebas neuropsicológicas.
CU6	Buscar Paciente.	Cliente Diana.	Caso de uso que pudiera complementar el CU5 , puesto que si el paciente ya se había consultado con anterioridad sólo se busca en el sistema.
CU14	Editar especialista.	Cliente Diana.	Caso de uso que se encarga de la modificación de los datos del especialista, también permite eliminarlo.
CU15	Editar paciente	Cliente Diana.	Caso de uso que se encarga de la modificación de los datos del paciente, también permite eliminarlo.
CU16	Editar paquete.	Cliente Diana.	Caso de uso que se encarga de la modificación de los datos del paquete, también permite eliminarlo.
CU17	Editar estudio neuropsicológico.	Cliente Diana.	Caso de uso que se encarga de la modificación de los datos del estudio neuropsicológico, también permite eliminarlo.

Ciclo 3:

Cód.	Nombre de caso de uso	Paquete	Justificación de la selección.
CU11	Visualizar Consulta.	Cliente Diana.	Visualiza las consultas reportadas.
CU12	Importar datos.	Cliente Diana.	<p>Importa datos de los que manipula el Cliente Diana, díganse pacientes, especialistas, etc.</p> <p>Importa los datos en un fichero predeterminado.</p>

CU13	Exportar datos.	Cliente Diana.	Exporta datos de los que manipula el Cliente Diana, díganse pacientes, especialistas, etc. Exporta los datos de un fichero predeterminado.
-------------	-----------------	----------------	---

2.8.5 Descripción extendida de los Casos de Uso.

Nombre del Caso de Uso:	Consultar Pacientes.		
Actores:	Especialista (Inicia)		
Propósito:	Aplicarle pruebas neuropsicológicas al paciente (consultar).		
Resumen:	<p>El actor captura sus datos, selecciona las pruebas neuropsicológicas que va a aplicar.</p> <p>El especialista envía una solicitud al sistema de ejecución, el mismo responde cargando las pruebas neuropsicológicas con su configuración predeterminada o prevista por el especialista.</p>		
Referencias:	REQ1.001, REQ1.009, REQ1.010, REQ1.011, REQ1.012, REQ1.013, REQ1.014.		
Tipo:	Crítico		
Precondiciones:	Se debe haber seleccionado el especialista.		
Curso Normal de los eventos			
Acciones de los Actores		Respuesta del Sistema	
1- Este Caso de Uso se inicia cuando el especialista se selecciona en una lista que muestra el sistema.		1.1 El sistema inserta el especialista en la consulta.	
2- Selecciona las pruebas neuropsicológicas que va a aplicar.			
3- Selecciona opción de comenzar ejecución de las pruebas neuropsicológicas.		3.1- Recibe evento de solicitud de ejecución.	
		4- Carga pruebas neuropsicológicas con la	

	configuración predeterminada o la especificada por el actor.
5- Recibe como respuesta la visualización de las pruebas neuropsicológicas.	
6- Caso de uso incluido CU9.	
	7- Guardar los datos generales de la consulta de forma automática.
Curso Alternativo de los eventos	
Línea 1: El especialista no está en la lista que muestra el sistema.	
1. El especialista captura sus datos.	
Poscondiciones:	Como resultado se tienen las consultas realizadas.

Nombre del Caso de Uso:	Realizar pruebas neuropsicológicas.
Actores:	Paciente (Inicia).
Propósito:	Que el paciente realice las pruebas neuropsicológicas para que pueda ser consultado (resultados).
Resumen:	El caso de uso es una inclusión del CU1, el sistema visualiza las pruebas neuropsicológicas y el actor paciente realiza las pruebas.
Referencias:	REQ1.013.
Tipo:	Crítico
Precondiciones:	Visualizar las pruebas neuropsicológicas.
Curso Normal de los eventos	
Acciones de los Actores	Respuesta del Sistema
1- Realizar la prueba neuropsicológica.	
	2- Terminar ejecución del ciclo de pruebas.
	3- Salvar resultados de las pruebas.
Curso Alternativo de los eventos	
Línea 1- El paciente no realiza las pruebas.	

Poscondiciones:	Resultados de las pruebas.
Nombre del Caso de Uso:	Configurar Pruebas Neuropsicológicas.
Actores:	Especialista (Inicia).
Propósito:	El paciente configura las pruebas neuropsicológicas según estime conveniente.
Resumen:	Caso de uso que se basa en la configuración de las pruebas neuropsicológicas.
Referencias:	REQ1.004.
Tipo:	Crítico
Precondiciones:	
Curso Normal de los eventos	
Acciones de los Actores	Respuesta del Sistema
1- Seleccionar prueba que va a configurar.	
2- Introduce datos de configuración.	
3- Selecciona opción salvar configuración.	
	4- Salva la configuración de la prueba neuropsicológica.
Curso Alternativo de los eventos	
Linea 2- El especialista selecciona opción de configuración predeterminada.	
Poscondiciones:	Configuración.

Nombre del Caso de Uso:	Configurar conexión al repositorio.
Actores:	Especialista (inicia).
Propósito:	Envíar información para el repositorio.
Resumen:	El especialista configura la conexión al servicio web y activa el plan de actualización.
Referencias:	REQ1.005, REQ1.006.
Tipo:	Crítico.

Precondiciones:	
Curso Normal de los eventos	
Acciones de los Actores	Respuesta del Sistema
1- Configura conexión al servicio web.	
2- Selecciona opción del plan de actualización.	
	5- Salva en un fichero la configuración.
	6- El servicio diana toma la configuración y trata de establecer comunicación con el web service.
	7- Envía información para el web service.
Curso Alternativo de los eventos	
Linea 6. El servicio no está instalado.	
Poscondiciones:	Queda establecida la conexión.

Nombre del Caso de Uso:	Generar Reporte.
Actores:	Especialista (inicia).
Propósito:	Completar la información de las consultas (consulta completada).
Resumen:	El especialista completa los datos de la consulta, datos como: diagnóstico, tratamiento, observación y captura los datos del paciente en caso de que se haya consultado de forma anónima.
Referencias:	REQ1.014, REQ1.015, REQ1.016, REQ1.017, REQ1.018, REQ1.019, REQ1.020.
Tipo:	Crítico.
Precondiciones:	Haber realizado la consulta.
Curso Normal de los eventos	
Acciones de los Actores	Respuesta del Sistema
	1- Muestra las consultas realizadas que aún no han sido completadas.
	2- Muestra los estudios neuropsicológicos

	almacenados en el sistema.
	3- Muestra los resultados de la prueba neuropsicológica aplicada en la consulta.
4- Selecciona la consulta.	
5- Selecciona estudio neuropsicológico.	
6- Inserta diagnóstico, tratamiento y observación.	
	7- Salva consulta completa en el sistema.
Curso Alternativo de los eventos	
Linea 5. Captura datos del paciente.	
	5- Muestra vista para capturar los datos del paciente.
6- Inserta datos del paciente.	
Poscondiciones:	Queda completada la consulta.

Nombre del Caso de Uso:	Almacenar información en la base de datos.
Actores:	Cliente de Repositorio (inicia).
Propósito:	Guardar toda la información enviada desde el Cliente Diana en la base de datos.
Resumen:	Se encarga de recibir la información desde el Cliente Diana, verifica la procedencia de los envíos y finalmente guarda la información recibida en la base de datos.
Referencias:	REQ3.001, REQ3.002, REQ3.003.
Tipo:	Crítico.
Precondiciones:	
Curso Normal de los eventos	
Acciones de los Actores	Respuesta del Sistema
1- El caso de uso es iniciado cuando recibe la petición de almacenar información.	
	2- Verifica procedencia del pedido.

3- Envía información a almacenar.	
	4- Almacena la información en la base de datos.
Curso Alternativo de los eventos	
Línea 2: Procedencia del pedido corrupta.	
	4- Cierra conexión.
Poscondiciones:	Información almacenada.

En el capítulo se han analizado los procesos de negocio que serán objeto de automatización. Se describió el sistema propuesto. Se modeló el flujo de los procesos del negocio representados en los diagramas de actividades. Se expusieron los requerimientos funcionales y no funcionales, mostrándose además los diagramas de casos de uso que se han modelado y una descripción detallada de los mismos pertenecientes al Modelo de Casos de Uso.

CAPÍTULO 3: ARQUITECTURA Y DISEÑO DEL SISTEMA

En el capítulo, se definen las clases del diseño del software de los casos de usos del primer ciclo de desarrollo; así como los diagramas de secuencias cumpliendo con las descripciones extendidas de los casos de uso. Se expone la arquitectura del sistema, favoreciendo el cumplimiento de los requisitos no funcionales. Describiéndose las clases entidades y las controladoras del flujo de trabajo del diseño y de las tablas de la base datos; de las que se muestran los diagramas Entidad-Relación agrupados por paquetes de responsabilidad para su mayor entendimiento.

3.1 Modelo arquitectónico.

La forma más práctica de utilizar los estilos arquitectónicos es mediante las cuatro C⁶: los componentes o elementos que forman el sistema; las conexiones que existen entre estos elementos (dispositivos de red), protocolos de comunicación; las restricciones tanto de comportamiento o como de características y finalmente la configuración que rige la unión de los elementos.

El Sistema para el Diagnóstico Neuropsicológico Automatizado presenta una arquitectura cliente servidor como se muestra en la Figura 8. Los elementos presentes en la misma, no son más que respuestas a las decisiones arquitectónicas tomadas para el cumplimiento de los requerimientos no funcionales y en algunos aspectos claves, los funcionales. Un ejemplo preciso es el tema del almacenamiento de las consultas neuropsicológicas en una base datos para lograr la persistencia de los mismos. El Cliente Diana se comunica con la base datos mediante un servicio web. También podemos observar como los clientes están separados del servidor logrando una alta cohesión y un bajo acoplamiento entre los componentes que lo soportan.

El Cliente Diana solamente utiliza los servicios del Repositorio de Información cuando se necesita almacenar las consultas neuropsicológicas completadas. Para esto utiliza un servicio que se mantiene activo en el cliente, se configura la conexión al repositorio y se reinicia el servicio. Este se encarga de establecer comunicación con la base datos mediante un servicio web que es el que se responsabiliza de verificar la procedencia de los pedidos, recibir la información y almacenarla. Es el encargado del negocio entre el Cliente Diana y la base datos.

⁶ Components, Connections, Communications and Constraints.

Arquitectura Física de Diana

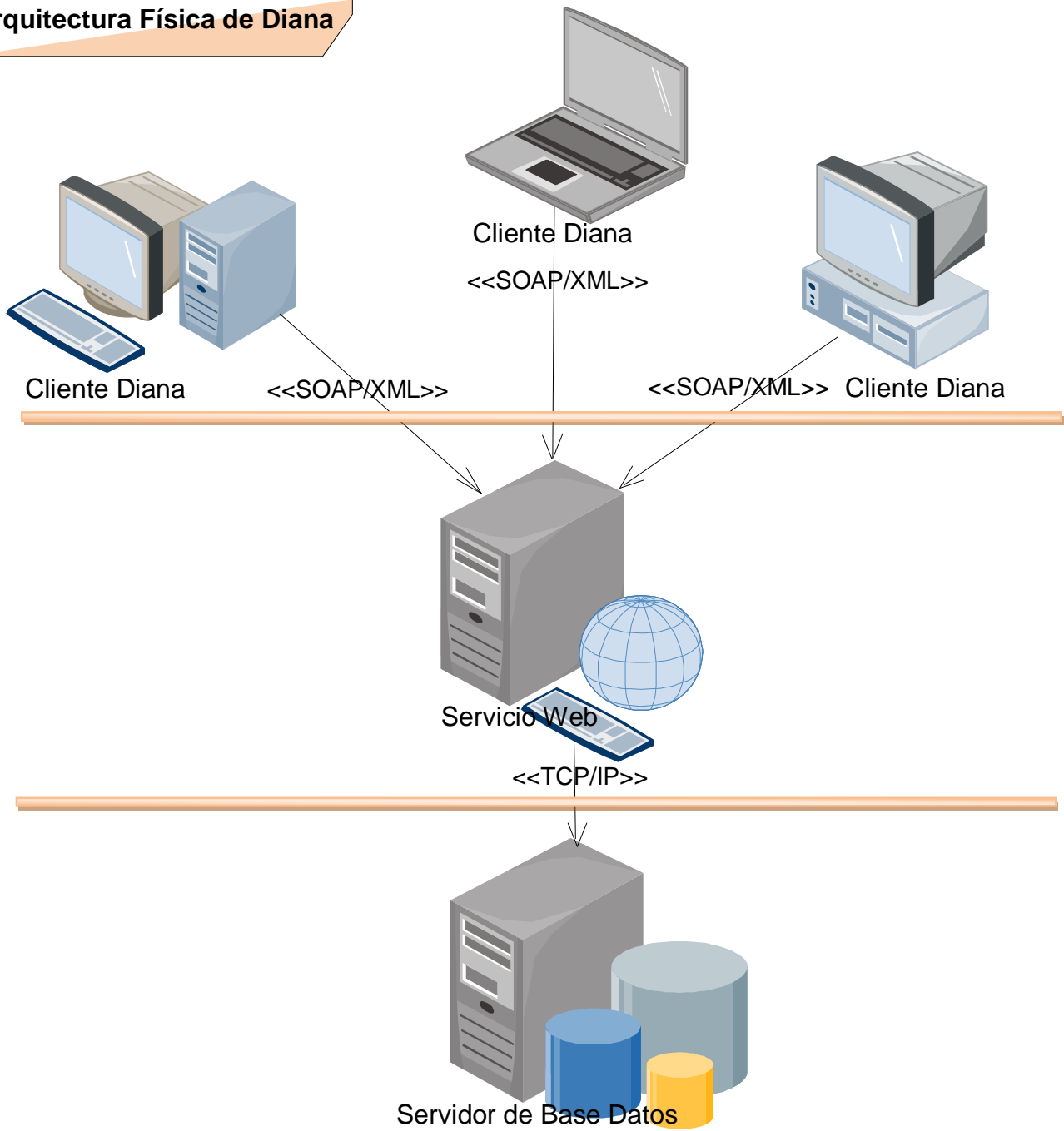


Figura 8 – Arquitectura Física del Sistema.

La estructura interna del sistema está distribuida en capas, ver Figura 9; el flujo de información se realiza jerárquicamente. La capa de presentación está compuesta por el Cliente Diana y el Servicio Diana, esta se comunica con la de acceso a datos mediante un intermediario que en nuestro caso es el servicio web que se encuentra localizado según la jerarquía, en la del negocio.

¿Qué beneficios nos trae la aplicación de una arquitectura como la antes descrita? Centralización de los aspectos de seguridad y transaccionalidad, que serían responsabilidad de la capa del negocio. No replicación de lógica del negocio en los clientes: esto permite que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de usuarios, reduciendo los costos de mantenimiento. Brinda mayor sencillez de los clientes.

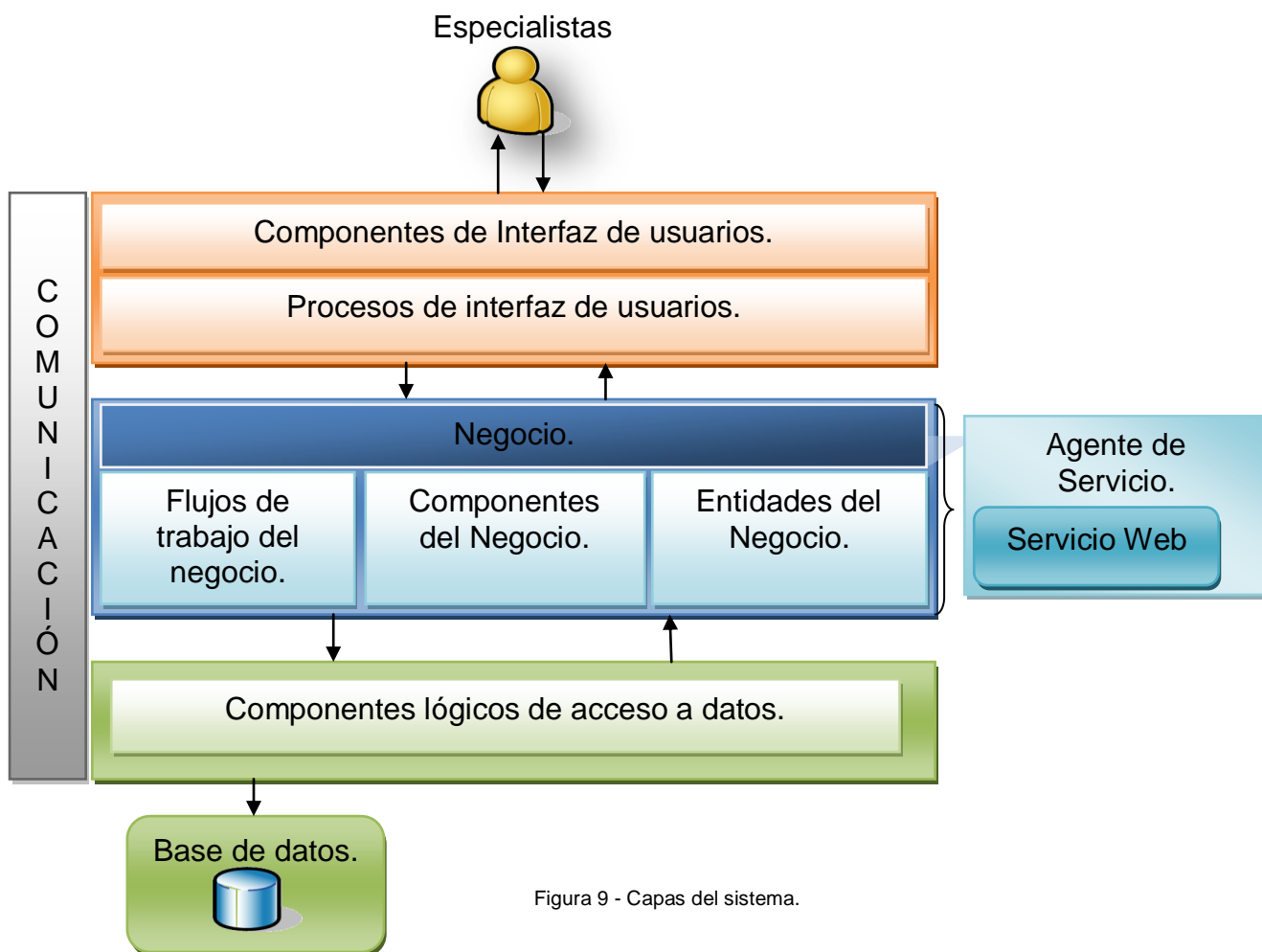


Figura 9 - Capas del sistema.

3.2 Modelo del Diseño

Las clases del diseño y las realizaciones de los casos de uso del primer ciclo de desarrollo describen una vista lógica, estructural y dinámica del sistema. Se describen las clases entidades y controladoras, percibiendo la interacción entre ellas y detallando particularidades del lenguaje CSharp, en que ha sido implementado el software. (Ver ANEXO 2: Realización de Casos de Uso del Diseño)

3.3 Descripción de las clases de diseño

Para entender mejor los detalles de las clases de diseño se puede ver la descripción de estas en ANEXO 3: DESCRIPCIÓN DE LAS CLASES del Diseño. Estas detallan la estructura de las clases controladoras y entidades involucradas en las realizaciones de los casos de usos más críticos en el funcionamiento del software.

3.4 Diseño de la Base de Datos.

En este epígrafe se exponen los diagramas ER (Entidad Relación), que se mostrarán agrupados por funcionalidad. Ver ANEXO 4: MODELO ENTIDAD RELACIÓN.

3.5 Descripción de las tablas.

Se describen los atributos que tendrá cada tabla de la Base Datos para el almacenamiento de la información requerida en la realización de los casos de usos, la descripción se encuentra en ANEXO 4: DESCRIPCIÓN DE LAS TABLAS DE LA BASE DATOS.

En este capítulo, se describió la arquitectura del sistema dándole cumplimiento a los requerimientos no funcionales del mismo. Se modeló en detalle, el sistema mediante las realizaciones de casos de uso del diseño, especificando los flujos de eventos e interacciones entre las clases mediante los diagramas de clases y secuencia.

CAPÍTULO 4: IMPLEMENTACIÓN

En el capítulo quedan expuestos los componentes del sistema Diana, se expresa en cada componente las funcionalidades del mismo y la interacción entre ellos. Al mismo tiempo, se presenta el despliegue propuesto para el sistema.

4.1 Diagrama de Componentes.

El núcleo de nuestro sistema es el Cliente Diana, que cuenta con dos módulos fundamentales: el de la aplicación que gestiona las pruebas neuropsicológicas de forma centralizada y el Servicio Diana. Este se mantiene ejecutándose en el mismo ordenador que la aplicación antes mencionada. El Repositorio de Información cuenta con el servicio web y la base datos. Los diagramas de componentes definen clases, artefactos, recursos y otros elementos de bajo nivel que se unen para conformar componentes que funcionan al más alto nivel. Todos ellos son artefactos del software compilado, que se convierten en esenciales para el funcionamiento del sistema informático. Estos trabajan acoplados, brindando el comportamiento que se requiere para el cumplimiento de los requerimientos antes expuestos en capítulos anteriores.

El sistema utiliza los componentes para poder visualizar las pruebas neuropsicológicas, lograr las funcionalidades de los sistemas RDAs, basados en el acceso a ficheros del Flash Player. Además de la librería de clases persistente, que provee los servicios, seguridad y conexión a la base datos.

Los componentes se encuentran repartidos por las distintas capas que plantea nuestra arquitectura, descrita en el capítulo anterior.

DIAGRAMA DE COMPONENTES

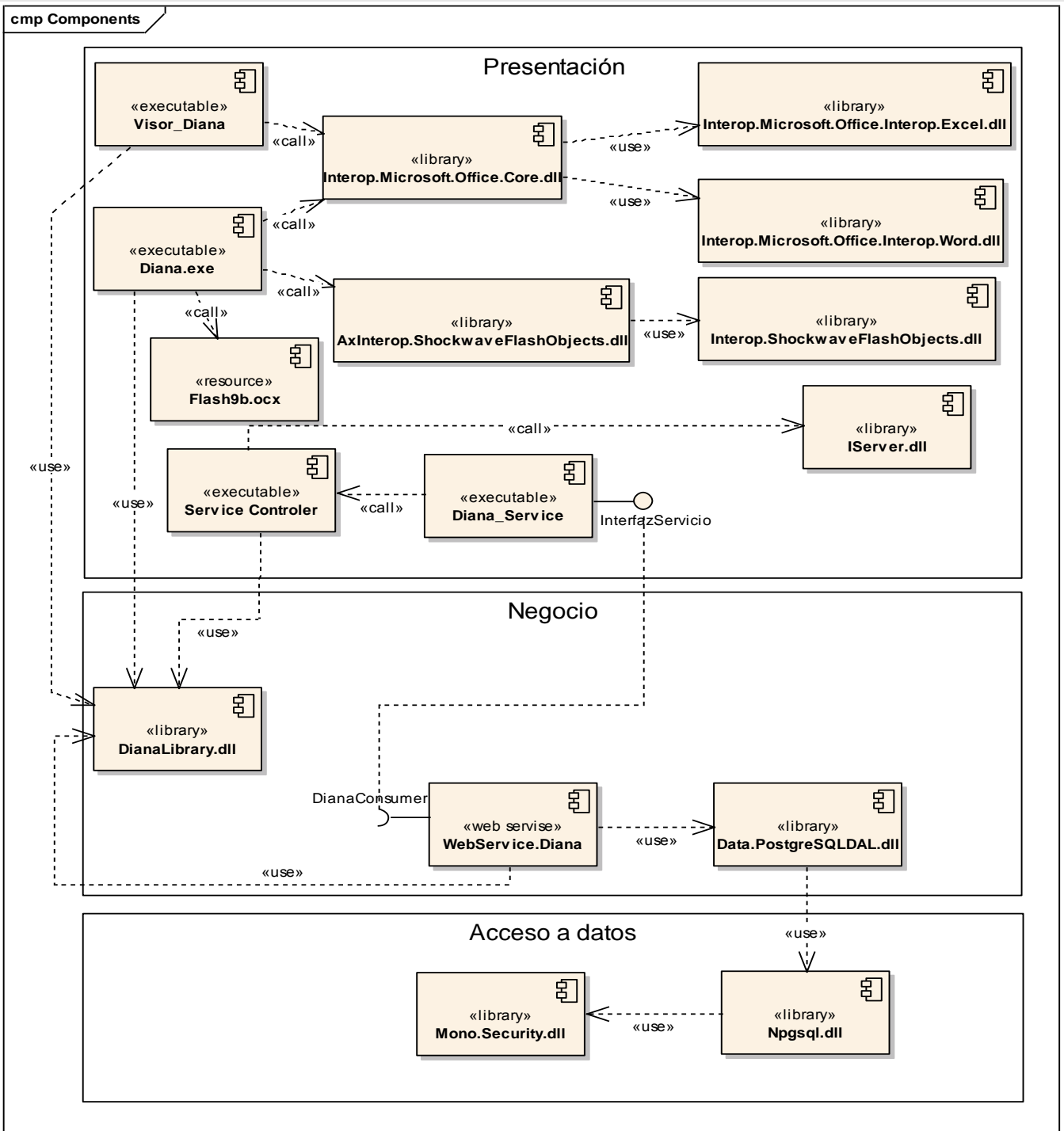


Figura 10 - Diagrama de Componentes

4.2 Diagrama de Despliegue.

Este diagrama describe cómo y dónde será instalado el sistema. Los dispositivos y procesadores son reflejados como nodos y su estructura interna puede ser representada adicionando otros nodos o artefactos. También se representan los componentes pertenecientes a cada uno. Para una mayor comprensión de cómo será instalado el sistema ver ANEXO 6: DIAGRAMA DE DESPLIEGUE.

En el capítulo se realizaron los diagramas de componentes y despliegue. Estos reflejan el resultado final del desarrollo del software, permitiendo apreciar el estilo arquitectónico predominante, los elementos que componen el sistema, la interacción entre ellos y su distribución física.

CONCLUSIONES

Con el desarrollo del trabajo se han llegado a las siguientes conclusiones:

Se implementó un sistema, teniendo en cuenta el estado actual de la neuropsicología en el mundo y en nuestro país. Se valoraron las tendencias y tecnologías líderes en el mundo del desarrollo informático y se obtuvo como resultado una aplicación enriquecida.

Se definió una arquitectura cliente servidor acorde a las necesidades de los clientes, esta provee un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios y recursos.

Se diseñó el sistema, teniendo en cuenta algunos de los principales patrones existentes logrando así la extensibilidad, flexibilidad y reusabilidad del código. Además, de la funcionalidad, la facilidad de configuración. Se crearon ambientes e interfaces amigables, teniendo en cuenta que las pruebas serán aplicadas para evaluar funciones neuropsicológicas, y que los futuros usuarios del sistema no tienen un nivel avanzado en las técnicas computacionales.

Se considera relevante que el sistema realizado constituye el principio y base fundamental de una herramienta que permitirá realizar estudios poblacionales y estadísticos, que pudiesen ser de gran impacto a nivel mundial.

RECOMENDACIONES

Sobre la presente investigación se recomienda:

- ✓ Desarrollar nuevas herramientas para facilitar la búsqueda de información en la base datos.
- ✓ Someter el sistema a pruebas de calidad de software para su posterior instalación en instituciones que requieran de sus servicios.
- ✓ Mantener el sistema de atención al cliente para obtener retroalimentación.
- ✓ Investigar sobre la tecnología **Apollo** para una futura migración del sistema y así lograr que sea multiplataforma.

REFERENCIAS BIBLIOGRÁFICAS

1. **NEURONIC.** *Manual del Usuario de Diana v2.0.* 1996.
2. Igual a Referencia 1.
3. **MADRID.** *Clasificación Estadística Internacional de Enfermedades y problemas relacionados con la salud (CIE10).* [Online] 2007. <http://www.madrid.org/iestadis/fijas/clasificaciones/cie10.htm>.
4. **PSICOMED.** *Manual Diagnostico y estadístico de los trastornos mentales.* [Online] 2006. http://personal.telefonica.terra.es/web/psico/dsmiv/dsmiv_int.html.
5. Igual a Referencia 1.
6. APA. American Psychological Association: About. [Online] 2007. <http://www.apa.org/>.
7. PREVI. Psicología y realidad virtual . Previ, el bienestar como terapia. [Online] 2003. <http://www.previsl.com/es/sprensa/dossier.asp>.
8. Alsina, Eduardo. Batería Neuropsicológica Luria-Nebraska, Formas I y II. [Online] 2006. <http://www.terapia.cl/website/parser.php?page=articulos&id=27>.
9. CNEURO. Schedules for Clinical Assessment in Neuropsychiatry. [Online] 2007. <http://www.cneuro.edu.cu/scan/>.
10. CERIL. Unidad de Evaluacion Psicologica Ceril: Wais/ Wisc. [Online] 2006. http://ceril.cln/P59_Udep.htm.
11. TIPS. Oferta de Software Médicos: Psisoft. [Online] 2000. <http://tips.org.uy/SPA/portal/NEGTexto.asp?Entidad=CUB&Numero=5968> .

12. COLOMBIACOMPITE. Inteligencia de Mercado Cuba. [Online] 2002.
<http://www.colombiacompite.gov.co/archivos/Cuba.doc>.
13. Igual a Referencia 1.
14. MSDN.Msdn Library: Visual Studio 2005. [Online] 2007. [http://msdn2.microsoft.com/es-es/library/6x6bk1f4\(vs.80\).aspx](http://msdn2.microsoft.com/es-es/library/6x6bk1f4(vs.80).aspx).
15. —. Msdn Library: Visual C#. [Online] 2007. [http://msdn2.microsoft.com/es-es/library/kx37x362\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/kx37x362(VS.80).aspx).
16. ADOBE. About Flash Player. [Online] 2007.
http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn_15511#about.
17. WIKIPEDIA. Wikipedia. Servicio Web. [Online] 2007.
http://es.wikipedia.org/wiki/Servicio_Web#Ventajas_de_los_servicios_Web.
18. POSTGRESQL. About. [Online] 2007. <http://www.postgresql.org/about/>.
19. APEXNET. Enterprise Architect v 6.5. [Online] 2007.
<http://www.apexnet.com.ar/index.php/product/viewProducts/24/sl=0>.
20. Rumbaugh, James, Jacobson, Ivar and Booch, Grady. El Proceso Unificado de Desarrollo de Software. Madrid : Addison Wesley, 2000. 84-7829-036-2.
21. EPIDATACONSULTING. Introducción a UML 2.0. [Online] 2005.
http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15.

BIBLIOGRAFÍA

SPARXSYSTEM. Enterprise Architect: Herramienta de diseño UML. [Online] 2007.
<http://www.sparxsystems.com.ar/products/ea.html>.

POSTGRESQL. PostgreSQL 8.2.4 Documentation. [Online] 2006.
<http://www.postgresql.org/docs/current/static/release-8-2.html>.

Hernández, Rolando and Coello, Sayda. El Paradigma Cuantitativo de la Investigación Científica. Ciudad de la Habana : Eduniv, 2002.

VANRIJKOM. Software Technology, Flash and Other Things. [Online] 2007. <http://www.vanrijkom.org/>.

MULTIMEDIA. Desktop Screen Capture to Adobe Flash: Capture v2. [Online] 2007.
<http://multimedia.com/>.

ADOBE. About Flash Player. [Online] 2007.
http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn_15511#about.

—. Embedding and Communicating with the Macromedia Flash Player in C# Windows Applications. [Online] 2007. http://www.adobe.com/devnet/flash/articles/stock_history03.html.

Pitman, Neil and Pilone, Dan. UML 2.0 in a Nutshell. s.l. : O'Reilly, 2005. 0-596-00795-7.

Rumbaugh, James, Jacobson, Ivar and Booch, Grady. El Proceso Unificado de Desarrollo de Software. Madrid : Addison Wesley, 2000. 84-7829-036-2.

Larman, Craig. UML y Patrones. México : Prentice Hall, 1999. 970-17-0261-1.

WIKIPEDIA. Patrón de Diseño. [Online] 2007.
http://es.wikipedia.org/wiki/Patr%C3%B3n_de_dise%C3%B1o.

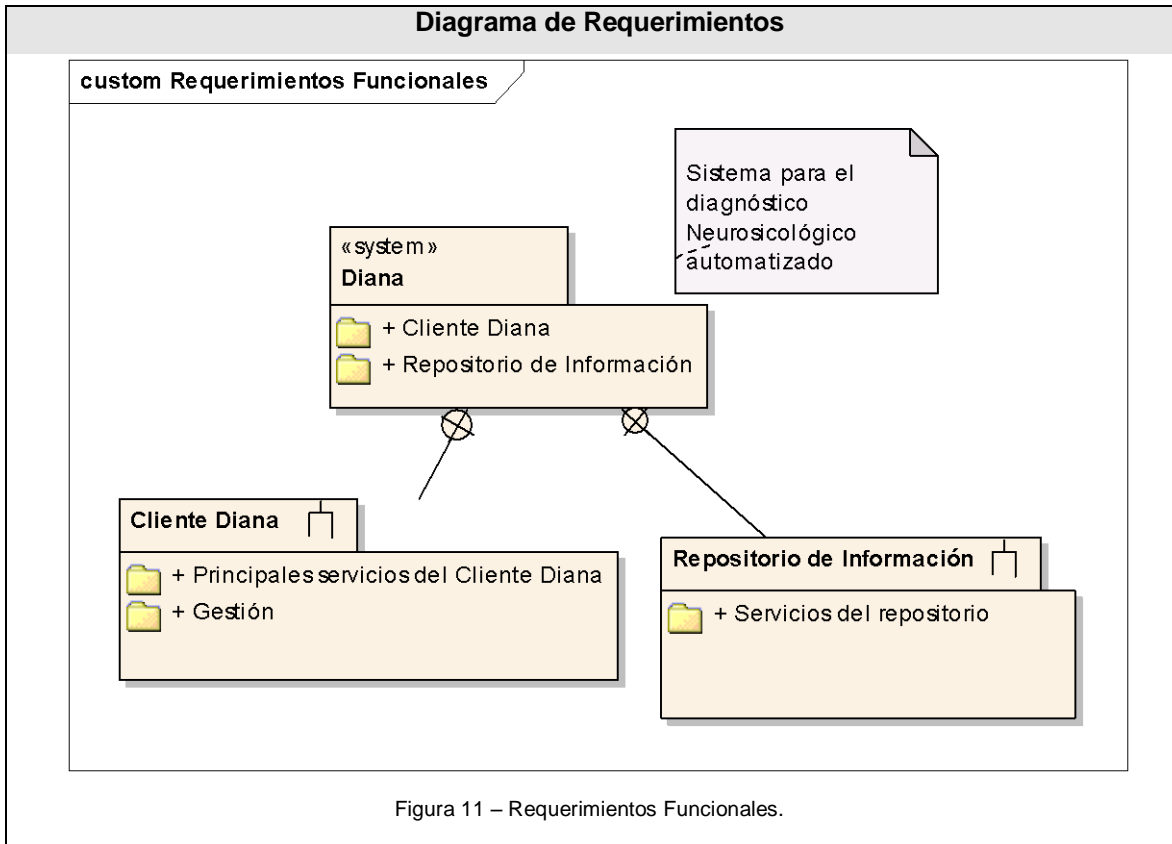
—. Cliente-servidor. [Online] 2007. <http://es.wikipedia.org/wiki/Cliente-servidor>.

Architect Academy: Seminario de Arquitectura de Software. Reynoso, Billy. Argentina : MSDN WEBCAST, 2005.

NEUROLOGIA. Revista de Neurología. [<http://www.revneurol.com/#Inicio>] Barcelona : Viguera Editores, 2007. 1576-6578.

ANEXO 1: DIAGRAMAS DE REQUERIMIENTOS

A.1.1 Diagrama de Requerimientos.



A.1.2 Cliente Diana

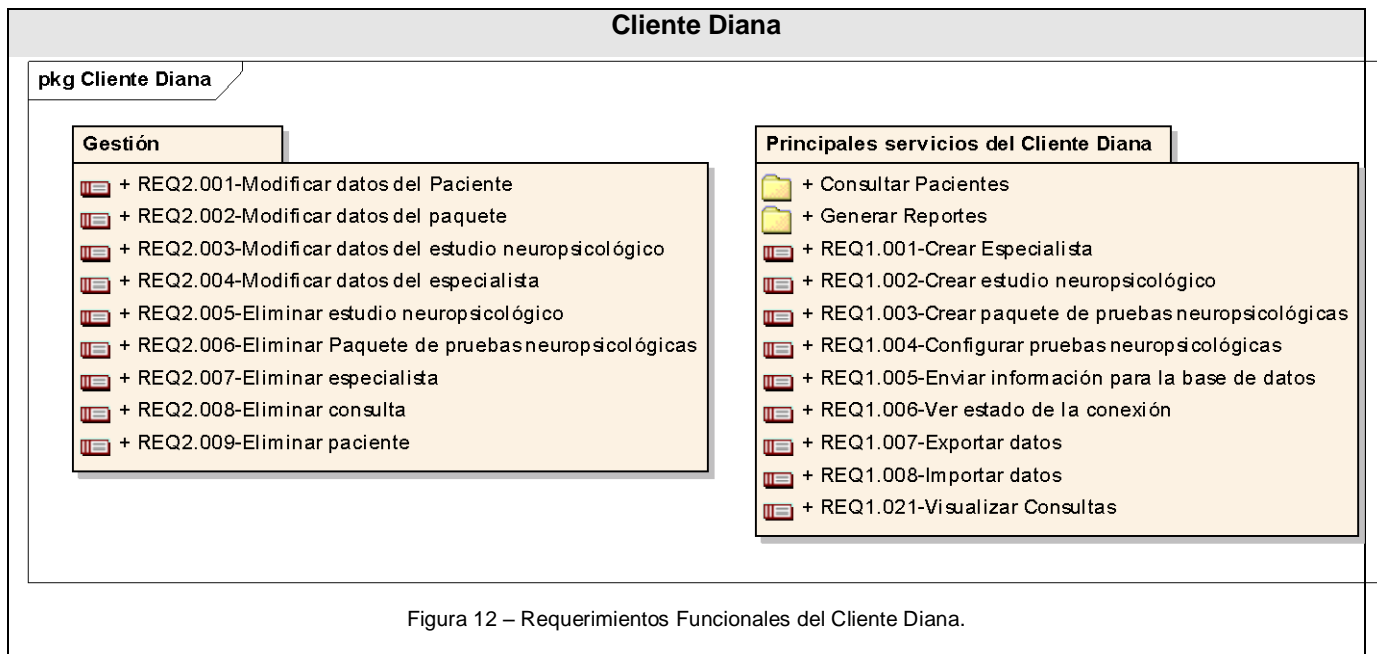


Figura 12 – Requerimientos Funcionales del Cliente Diana.

A.1.3 Requerimientos del Cliente Diana, principales servicios.

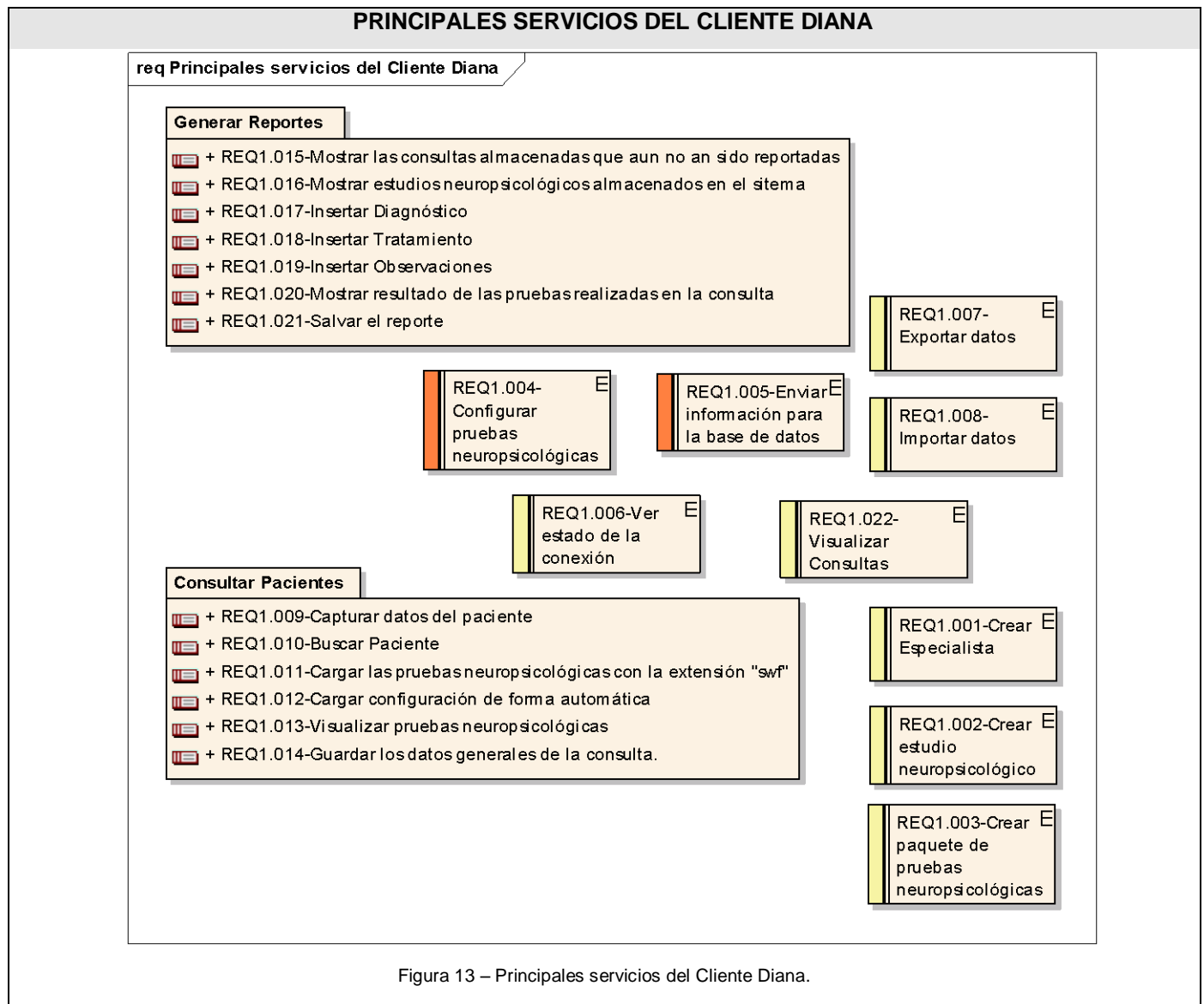
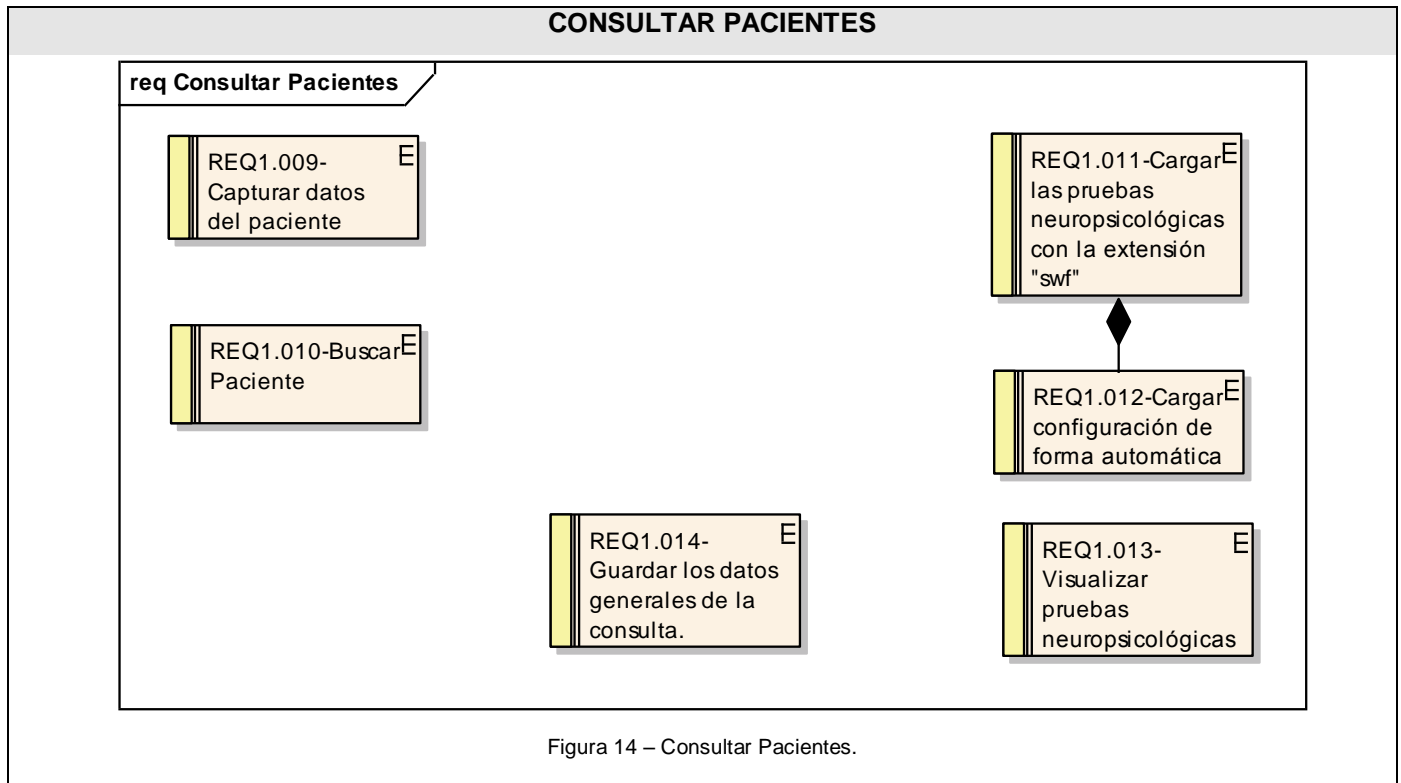
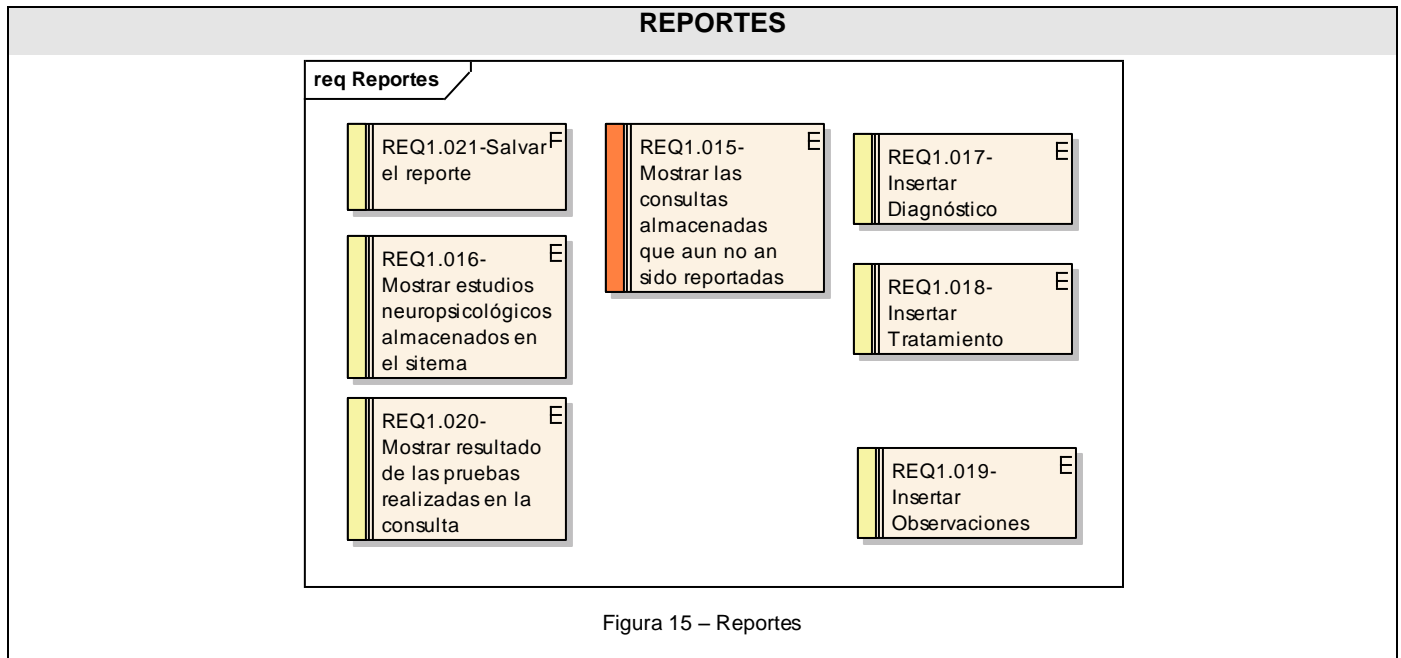


Figura 13 – Principales servicios del Cliente Diana.

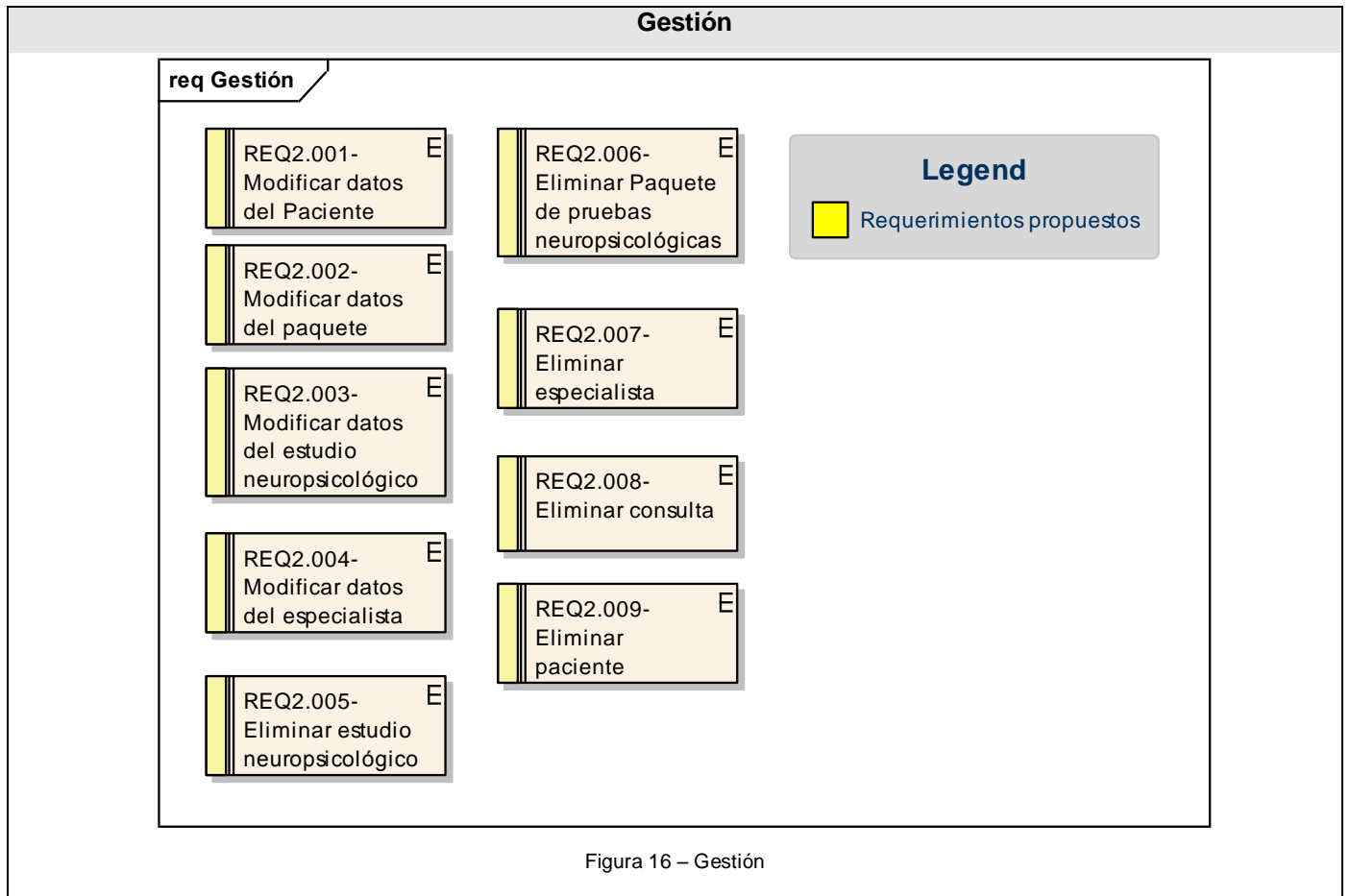
A.1.4 Requerimientos del Cliente Diana, principales servicios, consultar pacientes.



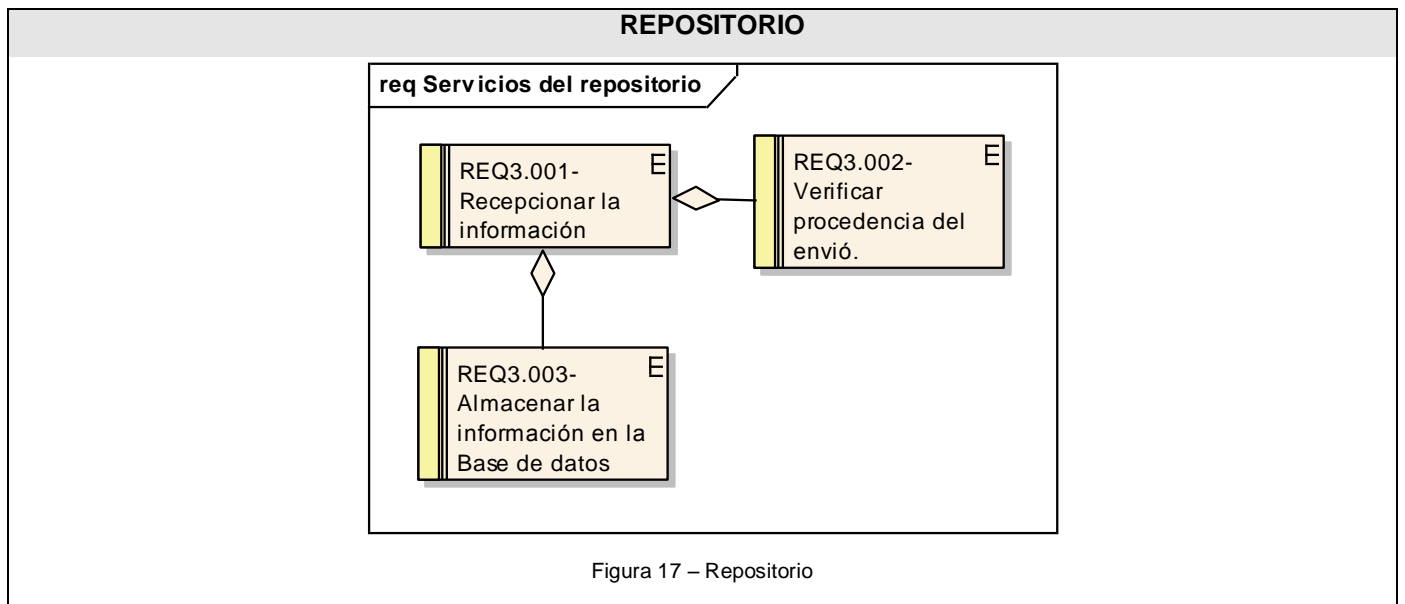
A.1.5 Requerimientos del Cliente Diana, principales servicios, reportes.



A.1.6 Requerimientos del Cliente Diana, gestión



A.1.7 Requerimientos del repositorio de información



ANEXO 2: REALIZACIÓN DE CASOS DE USO DEL DISEÑO

A.2.1 Caso de Uso Consultar Pacientes

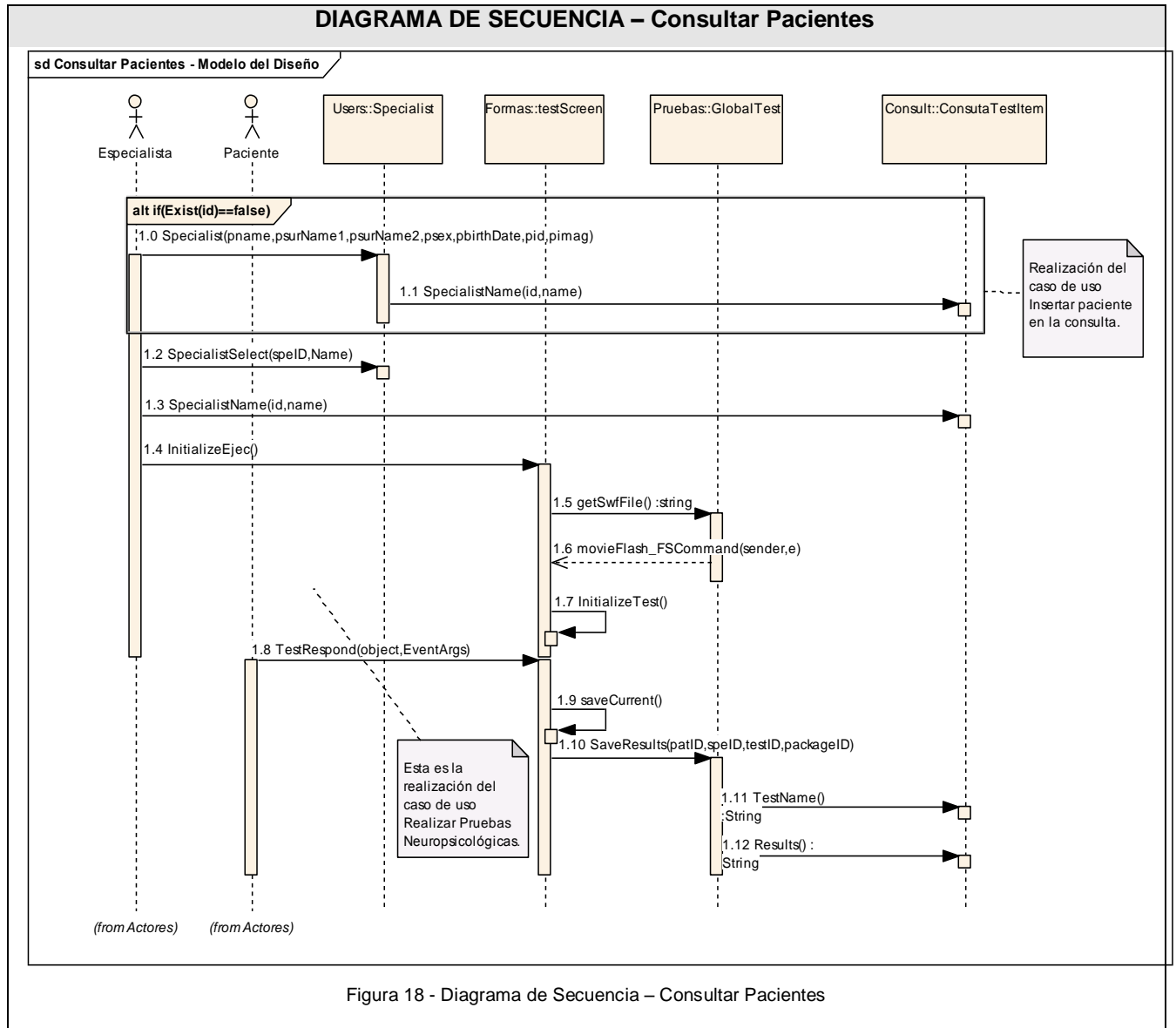


DIAGRAMA DE CLASES – Consultar Pacientes

class Consultar Pacientes - Modelo del Diseño

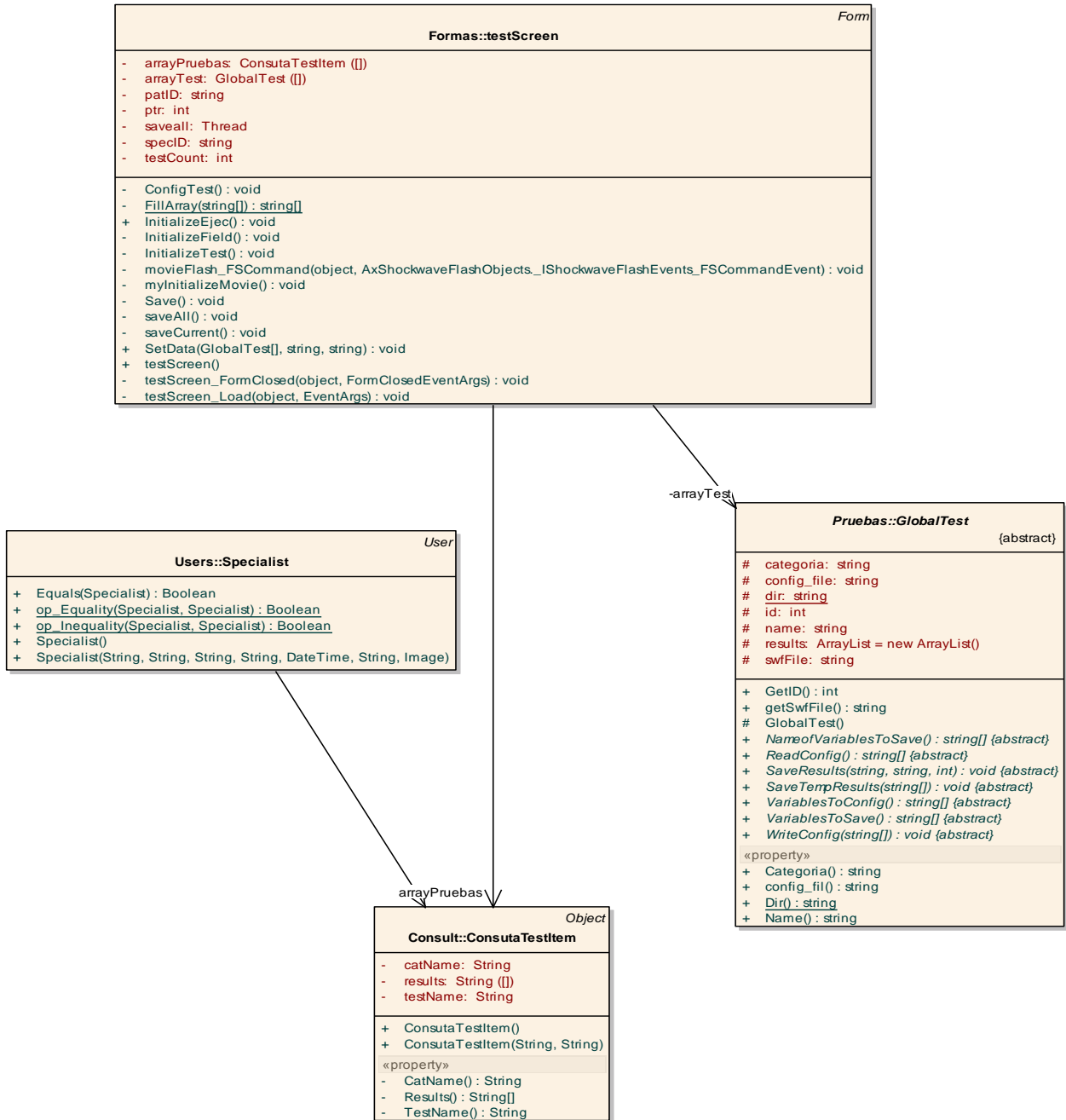


Figura 19 - Diagrama de Clases – Consultar Pacientes

A.2.2 Caso de Uso Configurar Pruebas Neuropsicológicas

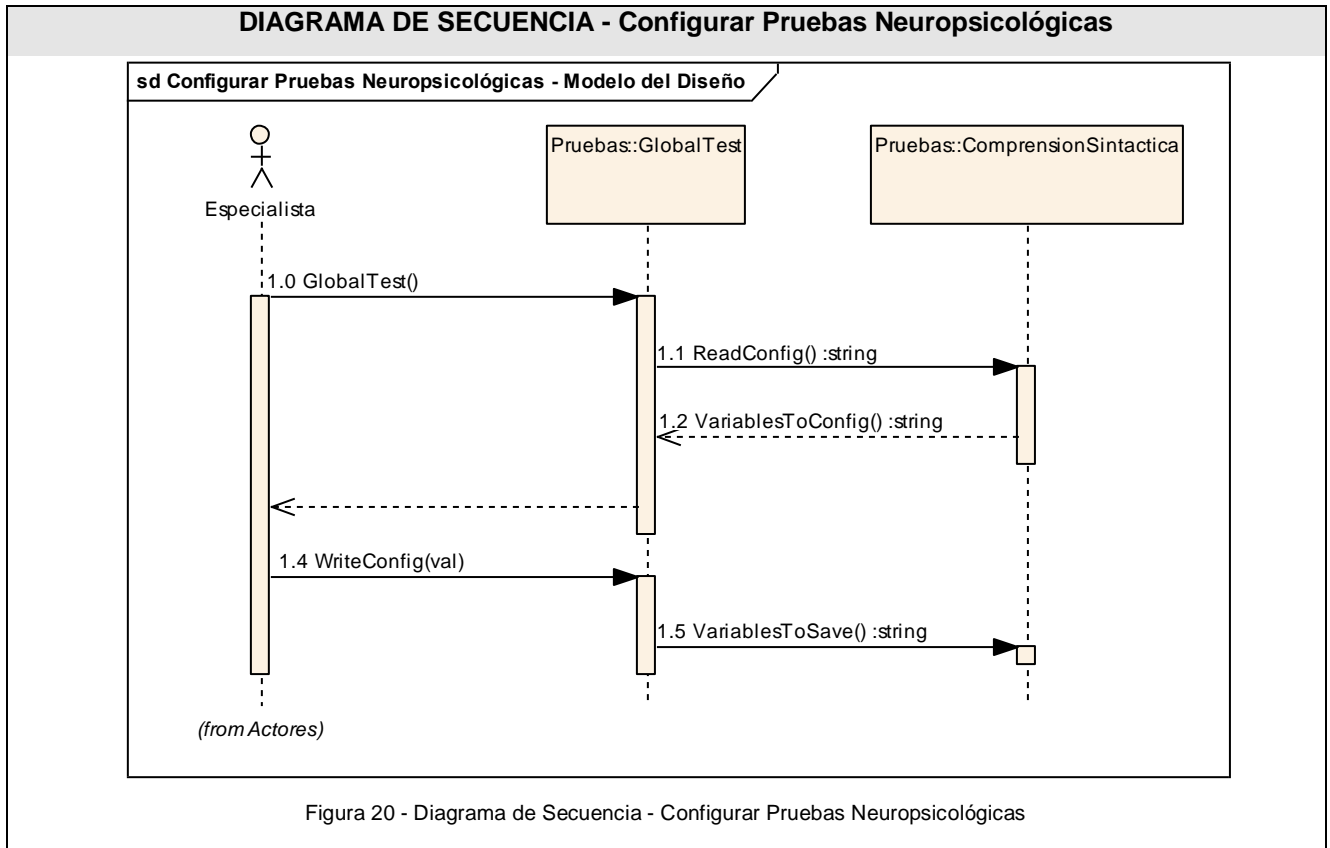


DIAGRAMA DE CLASES – Configurar Pruebas Neuropsicológicas

class Configurar Pruebas Neuropsicológicas - Modelo del Diseño

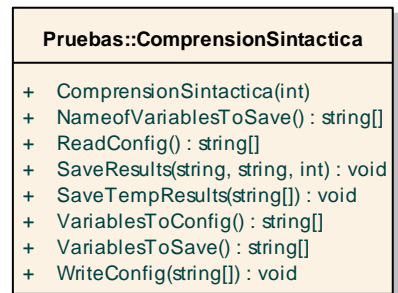
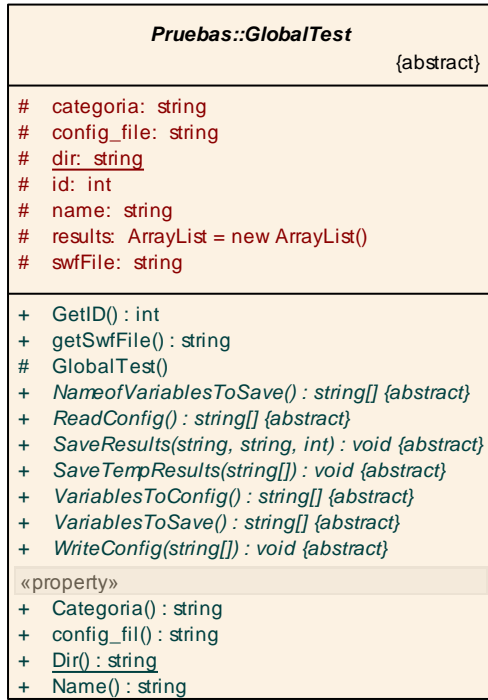


Figura 21 - Diagrama de Clases – Configurar Pruebas Neuropsicológicas

A.2.4 Caso de Uso Configurar Conexión al Repositorio.

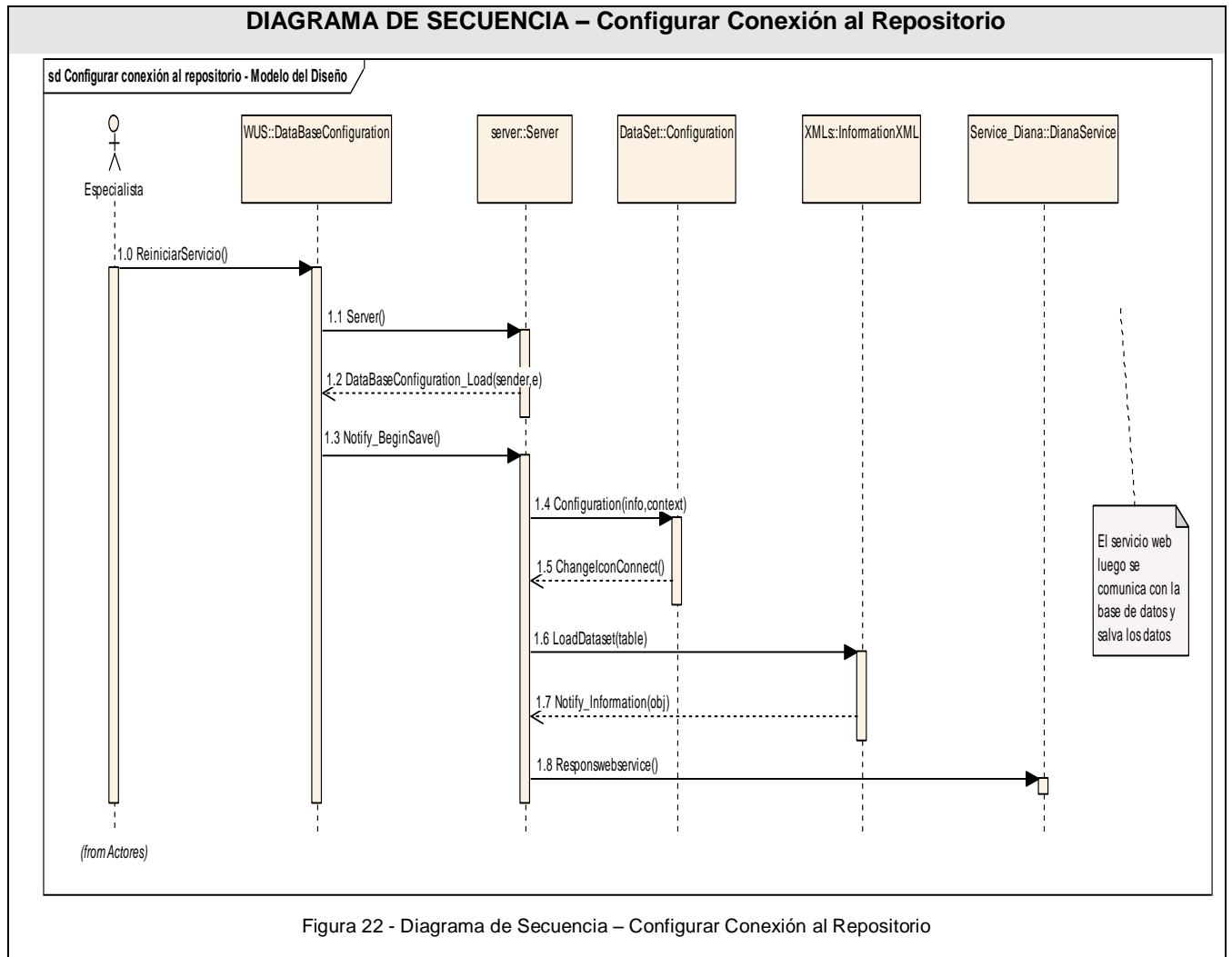


DIAGRAMA DE CLASES – Configurar Conexión al Repositorio

class Configurar conexión al repositorio - Modelo del Diseño

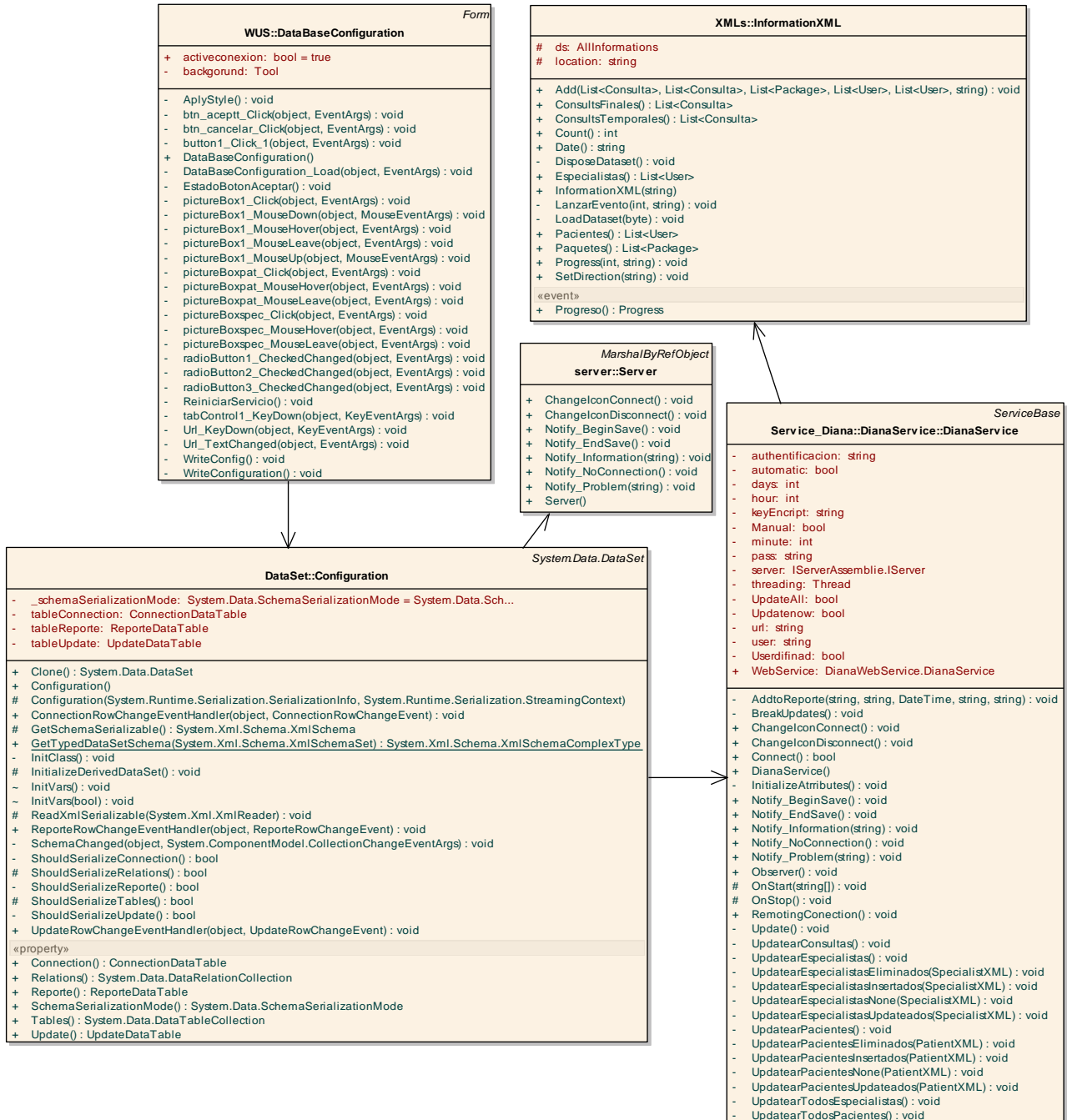


Figura 23 - Diagrama de Clases – Configurar Conexión al Repositorio

A.2.5 Caso de Uso Generar Reportes.

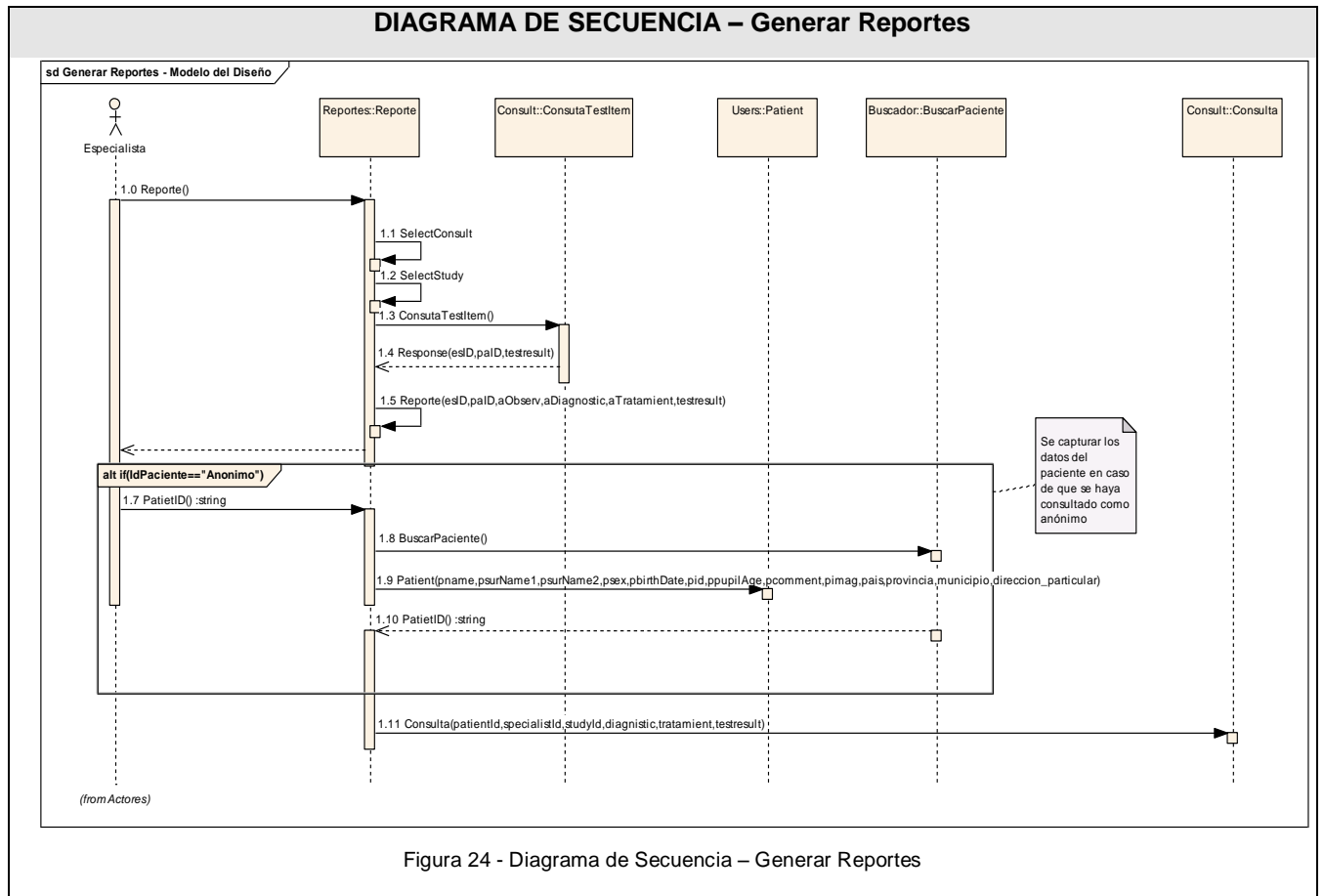


DIAGRAMA DE CLASES- Generar Reportes

class Generar Reportes - Modelo de Diseño

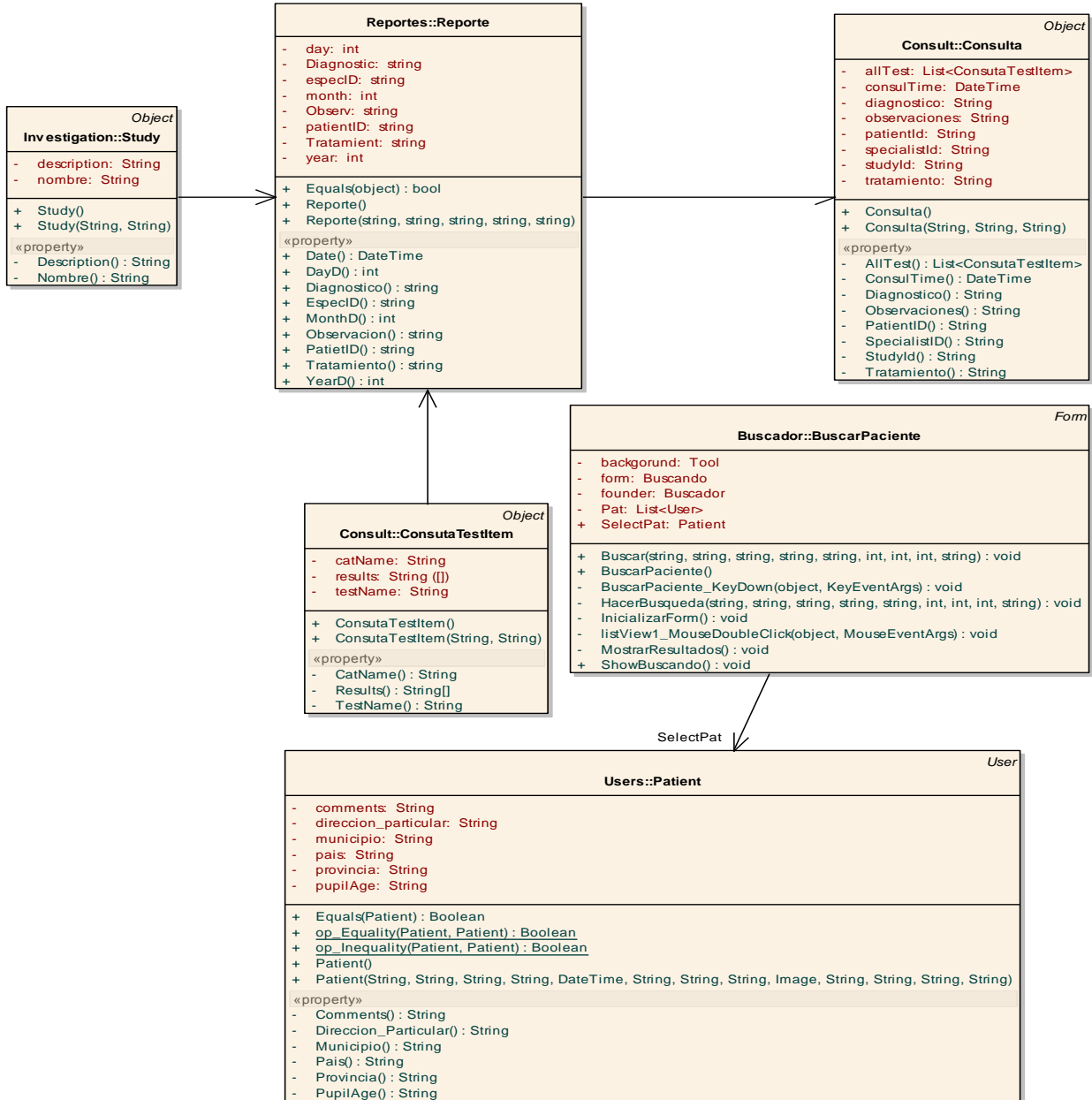


Figura 25 - Diagrama de Clases – Generar Reportes

A.2.6 Caso de Uso Almacenar Información en la base de datos

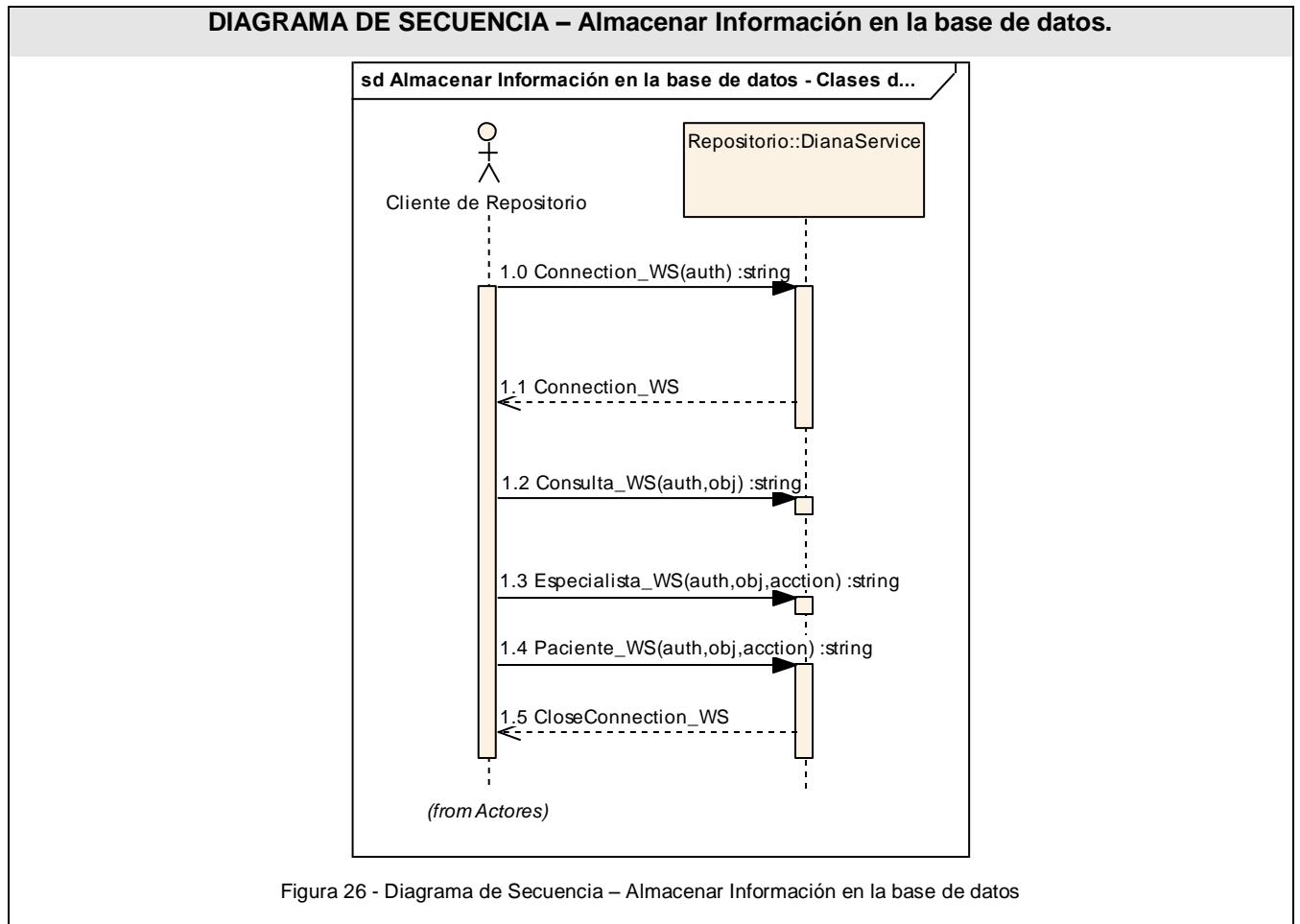


DIAGRAMA DE CLASES - Almacenar Información en la base de datos

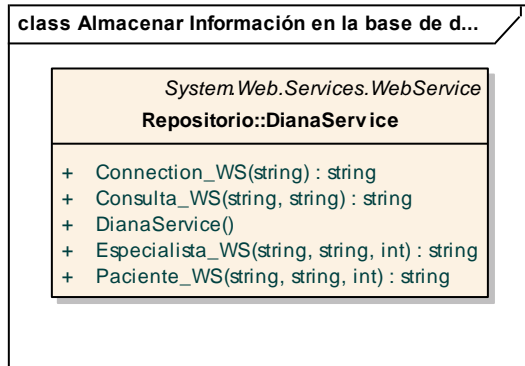


Figura 27- Diagrama de Clases – Almacenar Información en la base de datos.

ANEXO 3: DESCRIPCIÓN DE LAS CLASES DEL DISEÑO

A.3.1 Descripción de la clase DianaService.

Nombre:	DianaService.	
Tipo de clase :	Controladora.	
Atributo		Tipo
Para cada responsabilidad:		
Nombre:	Connection_WS(string auth,)	
Descripción:	Método que provee la autorización para la conexión.	
Nombre:	Consulta_WS(string auth,string obj,)	
Descripción:	Método que permite salvar la consulta completada en la base de datos.	
Nombre:	DianaService()	
Descripción:	Constructor de la clase.	
Nombre:	Especialista_WS(string auth,string obj,int accion,)	
Descripción:	Método que permite salvar los datos del especialista en la base de datos.	
Nombre:	Paciente_WS(string auth,string obj,int accion,)	

A.3.2 Descripción de la clase GlobalTest

Nombre:	GlobalTest	
Tipo de clase :	Entidad	
Atributo		Tipo
categoria	string	
config_file	string	
dir	string	
id	int	
name	string	
Resultados	ArrayList	
swfFile	string	
Para cada responsabilidad:		
Nombre:	Categoria()	
Descripción:	Método que tiene la categoría de las pruebas neuropsicológicas, memoria, procesos visos espaciales, procesos ejecutivos, lenguaje y atención.	
Nombre:	Dir()	

Descripción:	Directorio donde se encuentran las pruebas neuropsicológicas.
Nombre:	GetID()
Descripción:	Captura el id de las pruebas neuropsicológicas, este id lo utiliza para llamarlas.
Nombre:	getSwfFile()
Descripción:	Retornará la dirección en la que se encuentra el archivo .swf de la prueba. @returns
Nombre:	GlobalTest()
Descripción:	Constructor de la clase.
Nombre:	NameofVariablesToSave()
Descripción:	Retorna el nombre de las variables existentes en el *.swf que se encargan de la configuración de la prueba. @returns
Nombre:	ReadConfig()
Descripción:	Lee del archivo de configuración de la prueba. @returns
Nombre:	SaveResults(string patID,string spelD,int packageID,)
Descripción:	Salva los resultados finales de la prueba. @param ="vals"
Nombre:	SaveTempResults(string[] vals,)
Descripción:	Salvar los resultados temporalmente.
Nombre:	VariablesToConfig()
Descripción:	Retorna el nombre descriptivo de las variables existentes en el *.swf que se encargan de la configuración de la prueba. @returns
Nombre:	VariablesToSave()
Descripción:	Retorna el nombre de las variables existentes en el *.swf que contienen los resultados de la prueba. @returns
Nombre:	WriteConfig(string[] val,)
Descripción:	Escribe en el archivo de configuración de la prueba.

A.3.3 Descripción de la clase **CompresionSintactica**.

Nombre:	CompresionSintactica	
Tipo de clase :	Entidad	
	Atributo	Tipo
Para cada responsabilidad:		
Nombre:	CompresionSintactica(int ID,)	
Descripción:	Constructor de la clase.	
Nombre:	NameofVariablesToSave()	
Descripción:	Retorna el nombre de las variables existentes en el *.swf que se encargan de la configuración de la prueba. @returns	
Nombre:	ReadConfig()	
Descripción:	Lee del archivo de configuración de la prueba. @returns	
Nombre:	SaveResults(string patID,string spelD,int packageID,)	
Descripción:	Salva los resultados finales de la prueba. @param ="vals"	
Nombre:	SaveTempResults(string[] vals,)	
Descripción:	Salvar los resultados temporalmente.	
Nombre:	VariablesToConfig()	
Descripción:	Retorna el nombre descriptivo de las variables existentes en el *.swf que se encargan de la configuración de la prueba. @returns	
Nombre:	VariablesToSave()	
Descripción:	Retorna el nombre de las variables existentes en el *.swf que contienen los resultados de la prueba. @returns	
Nombre:	WriteConfig(string[] val,)	
Descripción:	Escribe en el archivo de configuración de la prueba.	

A.3.4 Descripción de la clase **DataBaseConfiguration**

Nombre:	DataBaseConfiguration
----------------	------------------------------

Tipo de clase :	Entidad	
	Atributo	Tipo
	backgorund	Style
Para cada responsabilidad:		
Nombre:	ApplyStyle()	
Descripción:	Se le aplica el estilo a la forma.	
Nombre:	button1_Click(Objeto de sender,EventArgs e,)	
Descripción:		
Nombre:	button1_Click_1(Objeto de sender,EventArgs e,)	
Descripción:		
Nombre:	button2_Click(Objeto de sender,EventArgs e,)	
Descripción:		
Nombre:	DataBaseConfiguration_Load(Objeto de sender,EventArgs e,)	
Descripción:	Evento que carga la configuración actual.	
Nombre:	EstadoBotonAceptar()	
Descripción:		
Nombre:	ReiniciarServicio()	
Descripción:	Cada vez que se sobrescribe el fichero de configuración se reinicia el servicio.	
Nombre:	SaveConfiguration()	
Descripción:	Permite salvar la configuración de la conexión.	
Nombre:	SaveConnection()	
Descripción:	Salva internamente el estado de la conexión.	

A.3.5 Descripción de la clase Server

Nombre:	Server	
Tipo de clase :	Entidad	
	Atributo	Tipo
Para cada responsabilidad:		
Nombre:	ChangelconConnect()	
Descripción:	Conecta el servicio con el web service.	
Nombre:	ChangelconDisconnect()	
Descripción:	Desconecta la conexión.	
Nombre:	Notify_BeginSave()	

Descripción:	Notifica que se va a empezar a salvar datos en el repositorio.
Nombre:	Notify_EndSave()
Descripción:	Notifica que se terminó de salvar.
Nombre:	Notify_NoConnection()
Descripción:	Notifica sobre la conexión fallida
Nombre:	Notify_Problem(string obj,)
Descripción:	Notifica sobre cualquier problema en el momento que se está salvando la información.
Nombre:	Server()
Descripción:	Constructor de la clase.

A.3.6 Descripción de la clase Configuration

Nombre:	Configuration	
Tipo de clase :	Entidad	
	Atributo	Tipo
	_schemaSerializationMode	System.Data.SchemaSerializationMode
	tableConnection	ConnectionDataTable
	tableReporte	ReporteDataTable
	tableUpdate	UpdateDataTable
Para cada responsabilidad:		
Nombre:	Clone()	
Descripción:		
Nombre:	Configuration()	
Descripción:		
Nombre:	Configuration(System.Runtime.Serialization.SerializationInfo info, System.Runtime.Serialization.StreamingContext context,)	
Descripción:		
Nombre:	Connection()	
Descripción:		
Nombre:	ConnectionRowChangeEventHandler(Objeto de sender, ConnectionRowChangeEvent e,)	
Descripción:		
Nombre:	GetSchemaSerializable()	
Descripción:		
Nombre:	GetTypedDataSetSchema(System.Xml.Schema.XmlSchemaSet xs,)	

Descripción:	
Nombre:	InitClass()
Descripción:	
Nombre:	InitializeDerivedDataSet()
Descripción:	
Nombre:	InitVars()
Descripción:	
Nombre:	InitVars(bool initTable,)
Descripción:	
Nombre:	ReadXmlSerializable(System.Xml.XmlReader reader,)
Descripción :	
Nombre:	Relations()
Descripción:	
Nombre:	Reporte()
Descripción:	
Nombre:	UpdateRowChangeEventHandler(Objeto de sender,UpdateRowChangeEvent e,)

A.3.7 Descripción de la clase InformationXML

Nombre:	InformationXML	
Tipo de clase :	Entidad	
	Atributo	Tipo
	ds	AllInformations
	location	string
Para cada responsabilidad:		
Nombre:	Add(List<Consulta> consultstemp,List<Consulta> consultsfinal,List<Paquete> paquetes,List<User> pacientes,List<User> especilistas,string date,)	
Descripción:	Adiciona la consulta en el XML	
Nombre:	ConsultsFinales()	
Descripción:	Método que me permiten organizar las consultas en este caso son las finales.	
Nombre:	Count()	
Descripción:	Cuenta la cantidad de elementos.	
Nombre:	DisposeDataset()	
Descripción:	Método que controla el dataset.	

Nombre:	Especialistas()
Descripción:	Método que me permite gestionar los especialistas.
Nombre:	InformationXML(string direction,)
Descripción:	Constructor de la clase
Nombre:	LanzarEvento(int progress,string imformation,)
Descripción:	
Nombre:	LoadDataset(byte table,)
Descripción:	Carga el dataset y llena sus tablas.
Nombre:	Pacientes()
Descripción:	Método que me permite gestionar los pacientes
Nombre:	Paquetes()
Descripción:	Método que me permite gestionar los paquetes.
Nombre:	Progress(int progress,string description,)
Descripción:	
Nombre:	SetDirection(string dir,)
Descripción :	Introduce el directorio donde se encuentra la información.

A.3.8 Descripción de la clase Specialist

Nombre:	Specialist	
Tipo de clase :	Entidad	
	Atributo	Tipo
	Para cada responsabilidad:	
Nombre:	Equals(Specialist obj,)	
Nombre:	op_Equality(Specialist sp1,Specialist sp2,)	
Descripción:		
Nombre:	op_Inequality(Specialist sp1,Specialist sp2,)	
Descripción:		
Nombre:	Specialist()	
Descripción:	Constructor de la clase.	
Nombre:	Especialistas()	
Descripción:	Método que me permite gestionar los especialistas.	
Nombre:	Specialist(String pname,String psurName1,String psurName2,String psex,DateTime pbirthDate,String pid,Image pimag,)	

Descripción:	Constructor de la clase inicializado con los datos del especialista.
--------------	--

A.8.9 Descripción de la clase ConsutaTestItem

Nombre:	ConsutaTestItem	
Tipo de clase :	Entidad	
	Atributo	Tipo
	catName	String
	Resultados	String
	testName	String
Para cada responsabilidad:		
Nombre:	CatName()	
Descripción:	Cuenta las consultas por el nombre.	
Nombre:	ConsutaTestItem()	
Descripción:	Constructor de la clase sin parámetros.	
Nombre:	ConsutaTestItem(String pcatName,String ptestName,)	
Descripción:	Constructor de la clase con parámetros.	
Nombre:	Resultados()	
Descripción:	Resultados de las pruebas neuropsicológicas realizadas en la consulta.	
Nombre:	TestName()	
Descripción:	Nombres de las pruebas neuropsicológicas.	

A.8.10 Descripción de la clase Reporte

Nombre:	Reporte	
Tipo de clase :	Entidad	
	Atributo	Tipo
	day	int
	Diagnostic	string
	especID	string
	month	int
	Observ	string
	patientID	string
	Tratamient	string
	year	int
Para cada responsabilidad:		

Nombre:	Date()
Descripción:	Datos generales del reporte.
Nombre:	DayD()
Descripción:	Día en que se completó la consulta.
Nombre:	Diagnostico()
Descripción:	Método que permite escribir y salvar el diagnóstico.
Nombre:	EspecID()
Descripción:	Id del especialista que la completó.
Nombre:	MonthD()
Descripción:	Mes en que se completó la consulta.
Nombre:	Observacion()
Descripción:	Método que permite escribir y salvar la Observación
Nombre:	PatientID()
Descripción:	Id del paciente que fue consultado
Nombre:	Reporte()
Descripción:	Constructor de la clase reporte
Nombre:	Reporte(string esID,string paID,string aObserv,string aDiagnostic,string aTratamient,)
Descripción:	Constructor de la clase con parámetros
Nombre:	Tratamiento()
Descripción:	Método que permite escribir y salvar el Tratamiento.
Nombre:	YearD()
Descripción:	Año en que fue completada la consulta

A.8.11 Descripción de la clase Patient

Nombre:	Patient	
Tipo de clase :	Entidad	
	Atributo	Tipo
	comments	String
	direccion_particular	String
	municipio	String
	pais	String
	provincia	String
	pupilAge	String
	comments	String

direccion_particular	String
Para cada responsabilidad:	
Nombre:	Direccion_Particular()
Descripción:	Dirección particular del paciente.
Nombre:	Equals(Patient obj,)
Descripción:	
Nombre:	Municipio()
Descripción:	Municipio al cual pertenece el paciente.
Nombre:	op_Equality(Patient pt1,Patient pt2,)
Descripción:	
Nombre:	op_Inequality(Patient pt1,Patient pt2,)
Descripción:	
Nombre:	Pais()
Descripción:	País de procedencia del paciente.
Nombre:	Patient()
Descripción:	Constructor de la clase vacío.
Nombre:	Patient(String pname,String psurName1,String psurName2,String psex,DateTime pbirthDate,String pid,String ppupilAge,String pcomment,Image pimag,String pais,String provincia,String municipio,String direccion_particular,)
Descripción:	Constructor de la clase con parámetros.
Nombre:	Provincia()
Descripción:	Provincia de procedencia del paciente.
Nombre:	PupilAge()
Descripción:	Edad del paciente, calcula la edad mediante la fecha de nacimiento.

A.8.12 Descripción de la clase BuscarPaciente

Nombre:	BuscarPaciente	
Tipo de clase :	Entidad	
	Atributo	Tipo
	backgorund	Tool
	form	Buscando
	founder	Buscador
	Pat	List<User>
	SelectPat	Patient
Para cada responsabilidad:		

Nombre:	Buscar(string name,string surname1,string surname2,string sex,string id,int dia,int mes,int anno,string escolaridad,)
Descripción:	Buscar Paciente desde el Cliente Diana, pacientes que se encuentren almacenados en el sistema en los XML.
Nombre:	BuscarPaciente()
Descripción:	Constructor de la clase.
Nombre:	BuscarPaciente_KeyDown(Objeto de sender,KeyEventArgs e,)
Descripción:	
Nombre:	HacerBusqueda(string name,string surname1,string surname2,string sex,string id,int dia,int mes,int anno,string escolaridad,)
Descripción:	
Nombre:	InicializarForm()
Descripción:	Inicializar forma donde va a mostrarse los pacientes.
Nombre:	listView1_MouseDoubleClick(Objeto de sender,MouseEventArgs e,)
Descripción:	
Nombre:	MostrarResultados()
Descripción:	Muestra todos los resultados de la búsqueda.
Nombre:	ShowBuscando()
Descripción:	Muestra estado de la búsqueda.

A.8.13 Descripción de la clase Consulta

Nombre:	Consulta	
Tipo de clase :	Entidad	
	Atributo	Tipo
	allTest	List<ConsutaTestItem>
	consulTime	DateTime
	diagnostico	String
	observaciones	String
	patientId	String
	specialistId	String
	studyId	String
	tratamiento	String
Para cada responsabilidad:		
Nombre:	AllTest()	
Descripción:	Todas las pruebas aplicadas en la consulta.	
Nombre:	Consulta()	

Descripción:	Constructor de la clase sin parámetros.
Nombre:	Consulta(String patientId,String specialistId,String studyId,)
Descripción:	Constructor de la clase con parámetros
Nombre:	ConsultTime()
Descripción:	Tiempo en el que se realizó la consulta.
Nombre:	Diagnostico()
Descripción:	Salva el diagnóstico en la consulta.
Nombre:	Observaciones()
Descripción:	Salva la observación en la consulta.
Nombre:	PatientID()
Descripción:	Tiene el paciente de la consulta por el ID.
Nombre:	SpecialistID()
Descripción:	Tiene el especialista de la consulta por el ID.
Nombre:	StudyId()
Descripción:	Tiene el estudio neuropsicológico en el que fue realizada la consulta por el ID.
Nombre:	Tratamiento()
Descripción:	Salva el tratamiento en la consulta.

ANEXO 4: MODELO DE DATOS

A.4.1 Diagramas Entidad - Relación

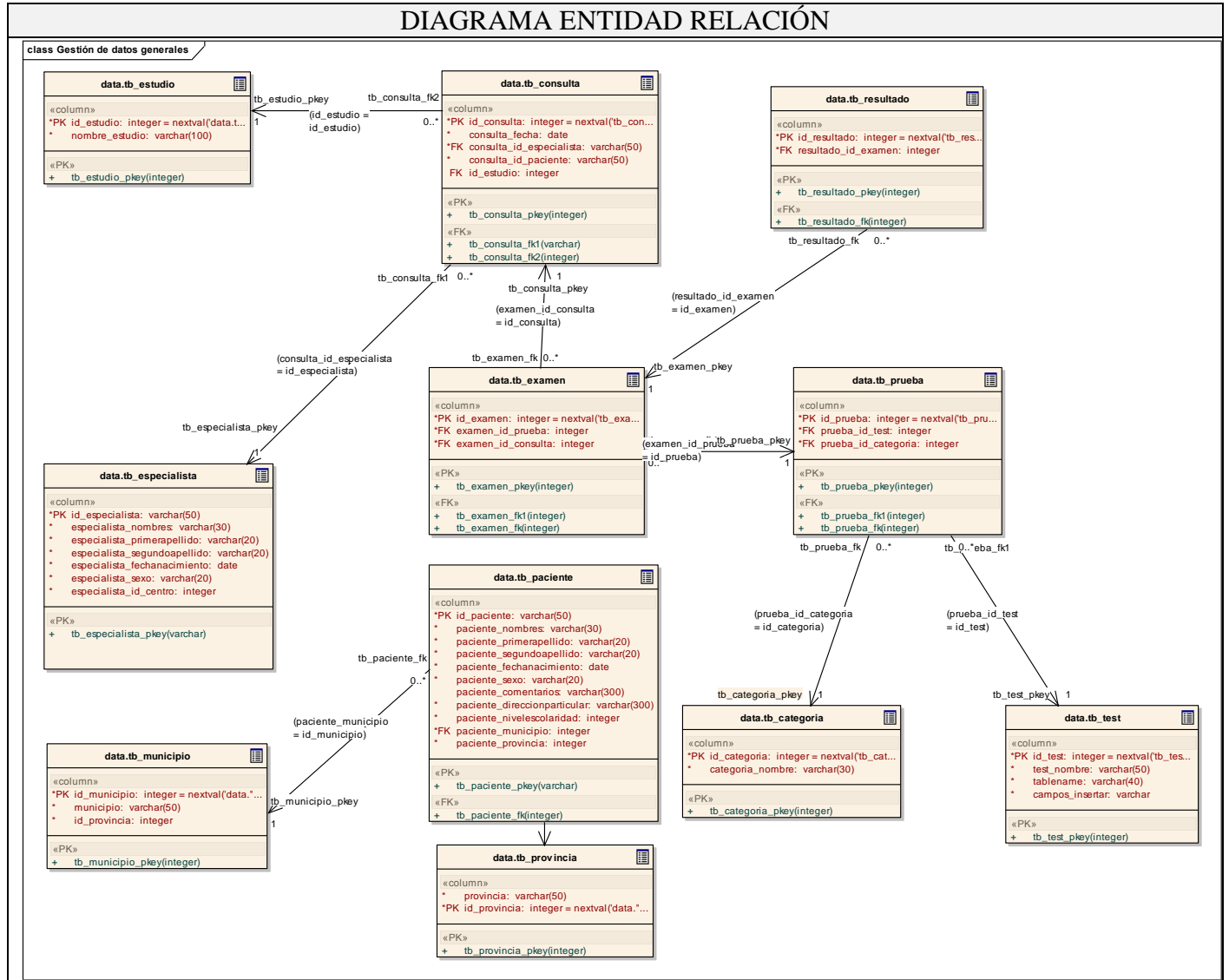


Figura 28. Diagrama Entidad Relación, gestión de datos generales

DIAGRAMA ENTIDAD RELACIÓN

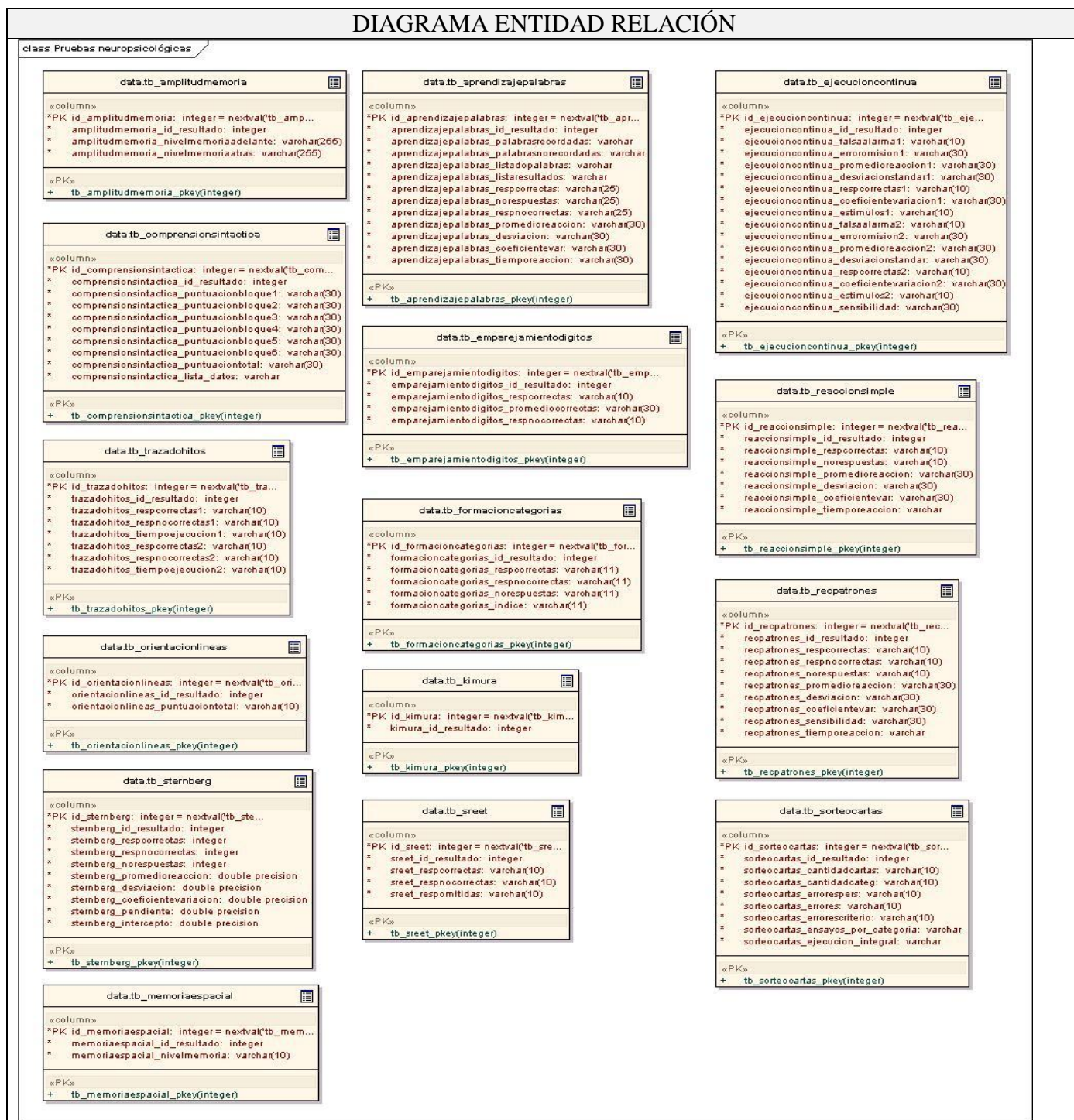


Figura 29. Diagrama Entidad – Relación, Pruebas Neuropsicológicas

A.4.2 Descripción de las tablas de la base de datos.

A.4.2.1 Gestión de datos generales.

Nombre: tb_categoria		
Descripción: La tabla está relacionada con la tabla prueba, va a contener la categoría a la cual pertenece la prueba neuropsicológica.		
Atributo	Tipo	Descripción
id_categoria	integer	
categoria_nombre	varchar	

Nombre: tb_consulta		
Descripción: La tabla está relacionada con la de examen, va a contener los datos de la consulta neuropsicológica.		
Atributo	Tipo	Descripción
id_consulta	integer	
consulta_fecha	date	
consulta_id_especialista	varchar	
consulta_id_paciente	varchar	
id_estudio	integer	

Nombre: tb_especialista		
Descripción: La tabla está relacionada con la de consulta, va a contener los datos del especialista de cada consulta.		
Atributo	Tipo	Descripción
id_especialista	varchar	
especialista_nombres	varchar	
especialista_primerapellido	varchar	
especialista_segundoapellido	varchar	
especialista_fechanacimiento	date	
especialistasexo	varchar	
especialista_id_centro	integer	

Nombre: tb_estudio		
Descripción: La tabla está relacionada con la de consulta va a contener el estudio que va a estar asociado con la misma.		
Atributo	Tipo	Descripción
id_estudio	integer	
nombre_estudio	varchar	

Nombre: tb_municipio		
Descripción: La tabla está relacionada con la de paciente, se especifica el municipio donde radica, se relaciona con la tabla consulta.		
Atributo	Tipo	Descripción

id_municipio	integer	
municipio	varchar	
id_provincia	integer	

Nombre: tb_paciente		
Descripción: Tabla donde van a estar contenidos los datos de los pacientes consultados por la aplicación.		
Atributo	Tipo	Descripción
id_paciente	varchar	
paciente_nombres	varchar	
paciente_primerapellido	varchar	
paciente_segundoapellido	varchar	
paciente_fecha_nacimiento	date	
paciente_sexo	varchar	
paciente_comentarios	varchar	
paciente_direccionparticular	varchar	
paciente_nivelescolaridad	integer	
paciente_municipio	integer	
paciente_provincia	integer	

Nombre: tb_provincia		
Descripción: Tabla que está relacionada con la del paciente, contiene los datos de la provincia a la cual pertenece.		
Atributo	Tipo	Descripción
provincia	varchar	
id_provincia	integer	

Nombre: tb_resultado		
Descripción: Tabla que está relacionada con la de examen, contiene los datos de los resultados de los exámenes.		
Atributo	Tipo	Descripción
id_resultado	integer	
resultado_id_examen	integer	

Nombre: tb_prueba		
Descripción: Tabla que está relacionada con la de examen, contiene los datos de las pruebas neuropsicológicas que han sido realizadas en cada examen, estos datos son los generales de las pruebas como: categoría y nombre.		
Atributo	Tipo	Descripción
id_prueba	integer	
prueba_id_test	integer	
prueba_id_categoria	integer	

Nombre: tb_test		
Descripción: Tabla que está relacionada con la de prueba, contiene los datos de la prueba		

de forma específica.		
Atributo	Tipo	Descripción
id_test	integer	
test_nombre	varchar	
tablename	varchar	
campos_insertar	varchar	

A.4.2.2 Pruebas neuropsicológicas

Nombre: tb_amplitudmemoria		
Descripción: Tabla que contiene las variables de configuración de cada prueba neuropsicológica.		
Atributo	Tipo	Descripción
id_amplitudmemoria	integer	
amplitudmemoria_id_resultado	integer	
amplitudmemoria_nivelmemoriaadelante	varchar	
amplitudmemoria_nivelmemoriaatras	varchar	

Nombre: tb_aprendizajepalabras		
Descripción: Tabla que contiene las variables de configuración de cada prueba neuropsicológica.		
Atributo	Tipo	Descripción
id_aprendizajepalabras	integer	
aprendizajepalabras_id_resultado	integer	
aprendizajepalabras_palabrasrecordadas	varchar	
aprendizajepalabras_palabrasnorecordadas	varchar	
aprendizajepalabras_listadopalabras	varchar	
aprendizajepalabras_listaresultados	varchar	
aprendizajepalabras_respcorrectas	varchar	
aprendizajepalabras_norespuestas	varchar	
aprendizajepalabras_respnocorrectas	varchar	
aprendizajepalabras_promedioreaccion	varchar	
aprendizajepalabras_desviacion	varchar	
aprendizajepalabras_coeficientevar	varchar	

aprendizajepalabras_tiem poreaccion	varchar	
--	---------	--

Nombre: tb_compressionsintactica		
Descripción: Tabla que contiene las variables de configuración de cada prueba neuropsicológica.		
Atributo	Tipo	Descripción
id_compressionsintactica	integer	
compressionsintactica_id _resultado	integer	
compressionsintactica_p untuacionbloque1	varchar	
compressionsintactica_p untuacionbloque2	varchar	
compressionsintactica_p untuacionbloque3	varchar	
compressionsintactica_p untuacionbloque4	varchar	
compressionsintactica_p untuacionbloque5	varchar	
compressionsintactica_p untuacionbloque6	varchar	
compressionsintactica_p untuaciontotal	varchar	
compressionsintactica_lis ta_datos	varchar	

Nombre: tb_ejecucioncontinua		
Descripción: Tabla que contiene las variables de configuración de cada prueba neuropsicológica.		
Atributo	Tipo	Descripción
id_ejecucioncontinua	integer	
ejecucioncontinua_id_res ultado	integer	
ejecucioncontinua_falsaal arma1	varchar	
ejecucioncontinua_erroro mision1	varchar	
ejecucioncontinua_prome dioreaccion1	varchar	
ejecucioncontinua_desvia cionstandar1	varchar	
ejecucioncontinua_respc orrectas1	varchar	
ejecucioncontinua_coefici entevariacion1	varchar	
ejecucioncontinua_estimu los1	varchar	
ejecucioncontinua_falsaal arma2	varchar	

ejecucioncontinua_erroremision2	varchar	
ejecucioncontinua_promedioreaccion2	varchar	
ejecucioncontinua_desviacionstandar	varchar	
ejecucioncontinua_respcorrectas2	varchar	
ejecucioncontinua_coeficientevariacion2	varchar	
ejecucioncontinua_estimulos2	varchar	
ejecucioncontinua_sensibilidad	varchar	

Nombre: tb_emparejamiento digitos

Descripción: Tabla que contiene las variables de configuración de cada prueba neuropsicológica.

Atributo	Tipo	Descripción
id_emparejamiento digitos	integer	
emparejamiento digitos_id_resultado	integer	
emparejamiento digitos_resp correctas	varchar	
emparejamiento digitos_promedior correctas	varchar	
emparejamiento digitos_respnocorrectas	varchar	

Nombre: tb_formacion categorías

Descripción: Tabla que contiene las variables de configuración de cada prueba neuropsicológica.

Atributo	Tipo	Descripción
id_formacion categorías	integer	
formacion categorías_id_resultado	integer	
formacion categorías_resp correctas	varchar	
formacion categorías_respnocorrectas	varchar	
formacion categorías_norespuestas	varchar	
formacion categorías_indice	varchar	

Nombre: tb_kimura

Descripción: Tabla que contiene las variables de configuración de cada prueba neuropsicológica.

Atributo	Tipo	Descripción
----------	------	-------------

id_kimura	integer	
kimura_id_resultado	integer	

Nombre: tb_memoriaespacial		
Descripción: Tabla que contiene las variables de configuración de cada prueba neuropsicológica.		
Atributo	Tipo	Descripción
id_memoriaespacial	integer	
memoriaespacial_id_resultado	integer	
memoriaespacial_nivelmemoria	varchar	

Nombre: tb_orientacionlineas		
Descripción: Tabla que contiene las variables de configuración de cada prueba neuropsicológica.		
Atributo	Tipo	Descripción
id_orientacionlineas	integer	
orientacionlineas_id_resultado	integer	
orientacionlineas_puntuaciontotal	varchar	

Nombre: tb_reaccionsimple		
Descripción: Tabla que contiene las variables de configuración de cada prueba neuropsicológica.		
Atributo	Tipo	Descripción
id_reaccionsimple	integer	
reaccionsimple_id_resultado	integer	
reaccionsimple_respuestas	varchar	
reaccionsimple_norespuestas	varchar	
reaccionsimple_promedio_reaccion	varchar	
reaccionsimple_desviacion	varchar	
reaccionsimple_coeficientevar	varchar	
reaccionsimple_tiempo_reaccion	varchar	

Nombre: tb_recpatrones		
Descripción: Tabla que contiene las variables de configuración de cada prueba neuropsicológica.		
Atributo	Tipo	Descripción
id_recpatrones	integer	
recpatrones_id_resultado	integer	

repatrones_respcorrectas	varchar	
repatrones_respnocorrectas	varchar	
repatrones_norespuestas	varchar	
repatrones_promediorreaccion	varchar	
repatrones_desviacion	varchar	
repatrones_coeficientev	varchar	
repatrones_sensibilidad	varchar	
repatrones_tiemporeaccion	varchar	

Nombre: tb_sorteocartas

Descripción: Tabla que contiene las variables de configuración de cada prueba neuropsicológica.

Atributo	Tipo	Descripción
id_sorteocartas	integer	
sorteocartas_id_resultado	integer	
sorteocartas_cantidadcartas	varchar	
sorteocartas_cantidadcategorías	varchar	
sorteocartas_errorespers	varchar	
sorteocartas_errores	varchar	
sorteocartas_errorescritorio	varchar	
sorteocartas_ensayos_por_categoria	varchar	
sorteocartas_ejecucion_integral	varchar	

Nombre: tb_sreet

Descripción: Tabla que contiene las variables de configuración de cada prueba neuropsicológica.

Atributo	Tipo	Descripción
id_sreet	integer	
sreet_id_resultado	integer	
sreet_respcorrectas	varchar	
sreet_respnocorrectas	varchar	
sreet_respomitidas	varchar	

Nombre: tb_sternberg

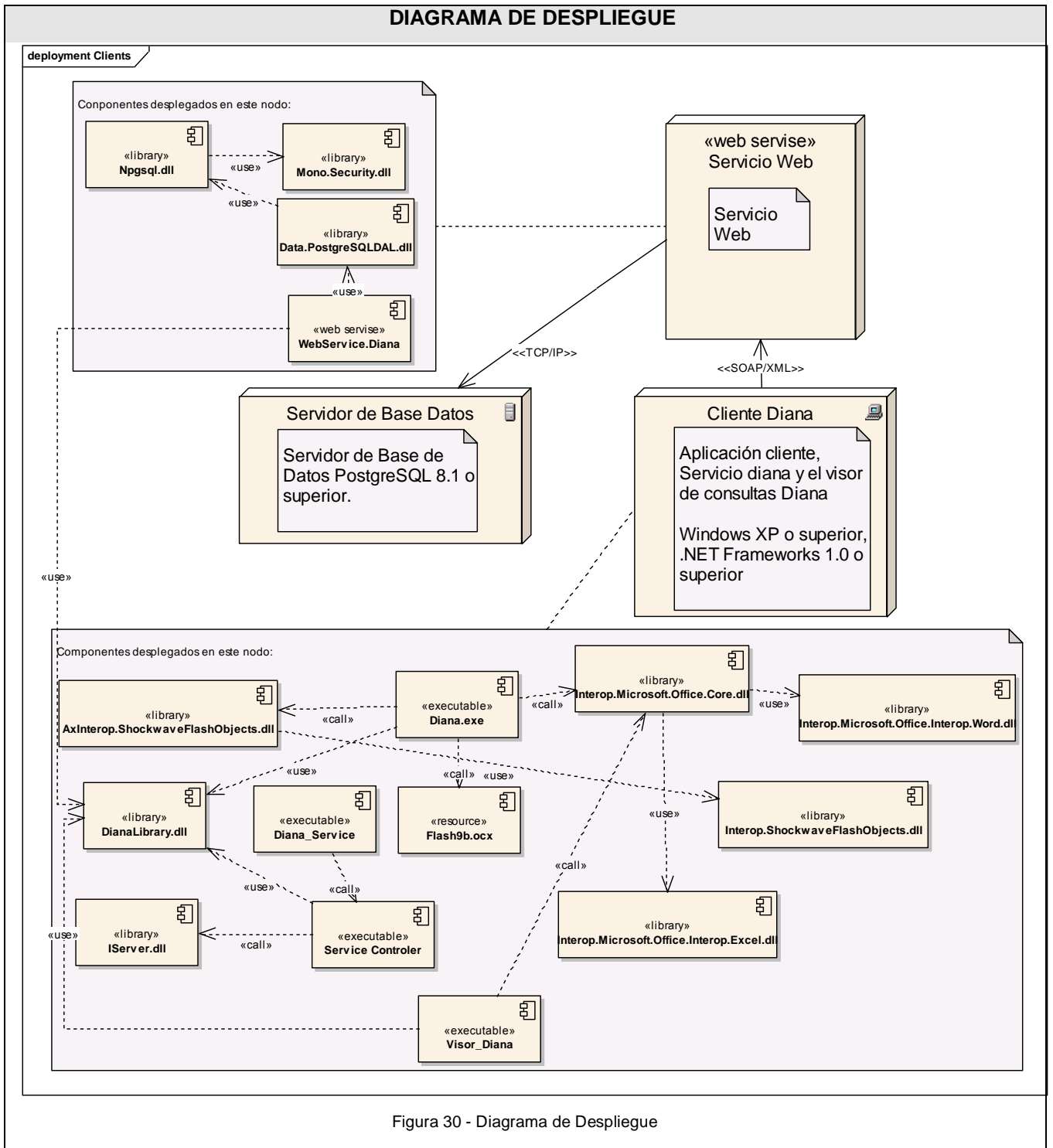
Descripción: Tabla que contiene las variables de configuración de cada prueba neuropsicológica.

Atributo	Tipo	Descripción
id_sternberg	integer	

sternberg_id_resultado	integer	
sternberg_respcorrectas	integer	
sternberg_respnocorrectas	integer	
sternberg_norespuestas	integer	
sternberg_promedioreaccion	double precision	
sternberg_desviacion	double precision	
sternberg_coeficientevariacion	double precision	
sternberg_pendiente	double precision	
sternberg_intercepto	double precision	

Nombre: tb trazadohitos		
Descripción: Tabla que contiene las variables de configuración de cada prueba neuropsicológica.		
Atributo	Tipo	Descripción
id_trazadohitos	integer	
trazadohitos_id_resultado	integer	
trazadohitos_respcorrectas1	varchar	
trazadohitos_respnocorrectas1	varchar	
trazadohitos_tiempoejecucion1	varchar	
trazadohitos_respcorrectas2	varchar	
trazadohitos_respnocorrectas2	varchar	
trazadohitos_tiempoejecucion2	varchar	

ANEXO 6: MODELO DE DESPLIEGUE DEL SISTEMA



GLOSARIO DE TÉRMINOS

Término	Grupo	Definición
Diana	Técnico	Sistema para el diagnóstico neuropsicológico automatizado.
Repositorio	Técnico	Localización física (directorio) de las consultas neuropsicológicas completadas en el dispositivo de almacenamiento.
Web Service	Técnico	Protocolos y estándares que sirven para intercambiar datos entre aplicaciones.
Apollo	Técnico	Tecnología de desarrollo que permite crear aplicaciones para escritorio usando otras tecnologías Flash, Adobe Flex, HTML, JavaScript y Ajax.
Neuropsiquiatría	Negocio	Rama de la medicina que trata de la relación entre el sistema nervioso y enfermedades psiquiátricas.
NEURONIC	Negocio	Comercializadora del Centro de Neurociencias de Cuba.
WYSIWYG	Técnico	Lo que tú ves es lo que recibes, aplicación que permite ver el documento en el momento de procesamiento como saldrá así impreso.
Concurrent Versions System (CVS)	Técnico	Aplicación informática que implementa un sistema de control de versiones.
Subversion	Técnico	Aplicación informática que implementa un sistema de control de versiones, es libre.
Visual SourceSafe (SCC)	Técnico	Aplicación informática que implementa un sistema de control de versiones.
TFS	Técnico	Aplicación informática que implementa un sistema de control de versiones.
XMI o XML Metadata Interchange	Técnico	Es una especificación para el Intercambio de Diagramas.

(XML de Intercambio de Metadatos)		
OMG	Técnico	Es un consorcio dedicado al cuidado y el establecimiento de diversos estándares de tecnologías orientadas a objetos, tales como UML, XMI, CORBA, etc.
Model-Driven Architecture o MDA	Técnico	Es un acercamiento al diseño de software, propuesto y esponsorizado por el <i>Object Management Group</i> .
CIREN	Negocio	El Ciren, de Cuba, hospital de primer mundo.(Centro Internacional de Restauración Neurológica).
IDE	Técnico	Condiciones de programación integradas dentro de una aplicación.

“Hombre es algo más que ser torpemente vivo: es entender una misión, ennoblecerla y cumplirla.”

José Martí.