

**Universidad de las Ciencias Informáticas**  
**"Facultad 6"**



**Título: “Herramienta para la migración de datos de  
Microsoft Access a PostgreSQL.”**  
**Módulo Examen Clínico.**

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

**Autores:** "Francisco Muñoz Tamayo"

"Félix Roberto Aballí Morell"

**Tutor:** "Ing. Yixander Yero Tarancón"

**Co-tutor:** "Ing. Maikel Laurencio Giralt"

**Asesora:** "Ing. Alionuska Sánchez Velázquez"

"Julio 2009"



*“Expresaré mi modesta opinión mientras pueda y necesite hacerlo.”*

*¡Los vivos y los muertos lucharemos!*

*Edi Barro*

## **DECLARACIÓN DE AUTORÍA**

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Francisco Muñoz Tamayo**

\_\_\_\_\_  
Firma del Autor

**Félix Roberto Aballí Morell**

\_\_\_\_\_  
Firma del Autor

**Yixander Yero Tarancón**

\_\_\_\_\_  
Firma del Tutor

## **DATOS DE CONTACTO**

**Tutor:** “Ing. Yixander Yero Tarancón”.

Instructor Recién Graduado de la Universidad de las Ciencias Informáticas, Facultad 6.

E-mail: [yyero@uci.cu](mailto:yyero@uci.cu)

**Co-tutor:** “Ing. Maikel Laurencio Giralt”.

Instructor Recién Graduado de la Universidad de las Ciencias Informáticas, Facultad 6.

E-mail: [mлаurencio@uci.cu](mailto:mлаurencio@uci.cu)

**Asesora:** “Ing. Alionuska Sánchez Velázquez”

Instructor Recién Graduado de la Universidad de las Ciencias Informáticas, Dirección de Calidad de Software.

E-mail: [avelazquez@uci.cu](mailto:avelazquez@uci.cu)

## **AGRADECIMIENTOS**

Agradezco a la Revolución por haberme formado durante todos estos años, por darme la oportunidad de formar parte de su Proyecto Futuro permitiéndome estudiar en una Universidad de Excelencia como lo es la Universidad de las Ciencias Informáticas. Agradezco a mis padres por su amor sin límites, por su confianza en mi y sus siempre buenos consejos. A mi hermanita, por ser la mejor hermana del mundo. A toda mi familia. A mi novia, por su amor incondicional, por su entrega, por su enorme confianza, preocupación, y por todo el apoyo que me ha dado durante estos dos años. A mi compañero de tesis, por todo el empeño que puso en la realización de este trabajo. A mis tutores por toda la ayuda que nos dieron. A Alionuska, por ser tan buena amiga, por su apoyo incondicional y desinteresado; sus consejos fueron de gran ayuda en la elaboración del documento de la tesis. A todos mis amigos. A las niñas de mi grupo (Maidelis, Yasnay, Kilmeny, Lisset, Kenia) por encargarse de que nunca las olvide. A Yenisel por ser tan buen amigo y “sobrino”. A mi amigo Manuel Alejandro Buch por su amistad incondicional desde niños. A los Cuadros de la UJC de la UCI por su amistad y preocupación. A Berta por ser tan buena amiga. A Mercedes por su ayuda incondicional y por ser la pinareña más loca que he conocido. A todos los que de una manera u otra han dejado su huella en mi durante estos 5 años ¡Muchas Gracias!

**Francisco Muñoz.**

## **DEDICATORIA**

Dedico este Trabajo de Diploma a mis padres Ana Luisa Tamayo y Francisco Muñoz por haberse sacrificado tanto durante todos estos años; por todo el amor que me han dado, por sus consejos, por siempre intentar guiarme por el camino correcto, por haber predicado siempre con el ejemplo. A mi hermana Adannis Muñoz por su amor incondicional, por su preocupación en todo momento, por su esfuerzo y sacrificio durante estos años, por ser mi hermanita linda. A mi novia, por ser lo mejor que me ha pasado. A mi abuelo Diógenes por estar siempre tan presente en todo lo que hago. A mi amigo Manuel Alejandro Buch porque hoy nos estamos graduando los dos. A mi familia y amigos.

**Francisco Muñoz.**

## **RESUMEN**

Durante el proceso de generación de conocimientos a través de investigaciones, estudios o el trabajo diario se producen grandes volúmenes de información que necesitan ser guardados y protegidos. El desarrollo que se ha alcanzado en el mundo de las tecnologías, ha permitido crear nuevas técnicas y maneras para almacenar esta valiosa información, las Bases de Datos.

El Proyecto Cubano Mapeo Cerebral Humano surgió con el objetivo de crear un mapa de las diferentes funciones del cerebro que representan tanto la superficie cortical como los núcleos internos del cerebro de los cubanos.

Impulsado por el objetivo de crear herramientas cuantitativas que permitan la investigación activa y la evaluación de tratamientos, surgió el Proyecto Informático Cubano de Mapeo Cerebral Humano encargado de procesar, obtener y gestionar toda la información que generan los estudios médicos. Como parte de este proyecto informático se creó el Sistema para la Gestión de la Información de los Exámenes Clínicos encargado de automatizar el proceso de recopilación, almacenamiento y procesamiento de los datos.

La información que resulta de cada estudio es recogida en una base de datos Microsoft Access, que debido a su incorrecto diseño, sumado al elevado volumen de información que recoge y la carga de transacciones a la que está sometida, resulta ineficiente, e imposibilita la migración de los datos hacia el gestor PostgreSQL utilizando métodos y herramientas convencionales. Por lo que se propuso crear una herramienta para la migración de datos que permita solucionar las dificultades gestoras en el procesamiento de dicha información.

## **PALABRAS CLAVE**

Evaluación de tratamientos, herramientas cuantitativas, recopilación, procesamiento, almacenamiento, migración, gestión, bases de datos, superficie cortical, núcleos internos.

## TABLA DE CONTENIDOS

<b>AGRADECIMIENTOS .....</b>	<b>I</b>
<b>DEDICATORIA.....</b>	<b>II</b>
<b>RESUMEN.....</b>	<b>III</b>
<b>TABLA DE CONTENIDOS.....</b>	<b>IV</b>
<b>INTRODUCCIÓN.....</b>	<b>1</b>
PROBLEMA CIENTÍFICO: .....	5
OBJETO DE ESTUDIO:.....	5
CAMPO DE ACCIÓN .....	5
OBJETIVO GENERAL.....	5
OBJETIVOS ESPECÍFICOS .....	5
TAREAS.....	6
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....</b>	<b>8</b>
1.1 INTRODUCCIÓN .....	8
1.2 MIGRACIÓN DE LA INFORMACIÓN. ....	8
1.2.1 Definición de Migración .....	8
1.3 ¿QUÉ ES EL SOFTWARE LIBRE? .....	10
1.3.1 Software Comercial .....	10
1.3.2 Software Shareware .....	11
1.3.3 Software Freeware .....	11
1.3.4 Software de Licencia Pública General (GPL). Software Libre .....	12
1.3.5 Beneficios del Software Libre. ....	13
1.4 IMPORTANCIA DE LA MIGRACIÓN HACIA SOFTWARE LIBRE PARA CUBA.....	14
1.4.1 Beneficios del Software Libre para Cuba [5].....	15
1.5 HERRAMIENTAS DE MIGRACIÓN.....	16
1.6 SISTEMAS GESTORES DE BASES DE DATOS.....	16
1.6.1 Microsoft Access .....	17
1.6.2 MySQL .....	18
1.6.3 PostgreSQL.....	19
1.6.4 ¿Por qué usar PostgreSQL? .....	21
1.7 PROCESOS DE DESARROLLO DE SOFTWARE.....	22

1.7.1 Extreme Programming (XP).....	22
1.7.2 Proceso Unificado de Desarrollo (RUP).....	23
1.7.3 Open Unified Process (OpenUP).....	25
1.7.4 ¿Por qué usar OpenUP? .....	26
1.7.5 Lenguaje de modelado. ....	26
1.7.6 Mapeo Relacional de Objeto (ORM).....	27
1.7.5.1 Propel.....	28
1.7.6 Lenguaje de programación. ....	28
1.7.6.1 PHP 5.....	28
1.7.6.2 JavaScript.....	29
1.7.7 Herramientas CASE .....	30
1.7.7.1 Visual Paradigm for UML.....	31
1.7.7.2 DBDesigner Fork.....	32
1.7.8 Editores PHP.....	33
1.7.8.1 Zend Studio .....	33
1.7.9 Editores de páginas Web .....	34
1.7.9.1 Dreamweaver .....	34
CONCLUSIONES DEL CAPÍTULO.....	35
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....</b>	<b>36</b>
2.1 INTRODUCCIÓN .....	36
2.2 DESCRIPCIÓN DE LOS PROCESOS A AUTOMATIZAR .....	36
2.3 PROCEDIMIENTO PARA MIGRAR LA BD .....	37
2.4 PROPUESTA DE MEJORAS AL MÓDULO DE EXAMEN CLÍNICO.....	39
2.4.1 Gestión de EEG .....	39
2.4.1.1 Definición de Electroencefalograma .....	39
2.4.1.2 Electrogénesis cerebral .....	39
2.4.1.3 Electrogénesis cortical.....	40
2.5.1.4 Sincronización de la actividad celular .....	41
2.4.1.5 Captación EEG.....	41
2.4.1.6 Actividad Cerebral .....	42
2.4.1.7 Actividad Patológica del Electroencefalograma .....	43
2.4.1.8 Actividad Normal .....	43
2.4.1.9 Alteraciones de la Actividad de Base.....	43

2.4.1.10 Actividad Focal .....	43
1.5.1.11 Actividad Paroxística .....	44
2.4.1.12 Técnicas de realización del Electroencefalograma .....	44
2.4.1.13 Prescripción del Electroencefalograma.....	44
2.4.1.14 Conclusiones Parciales .....	45
2.5 CAPTURA DE LOS REQUISITOS DEL NUEVO SISTEMA.....	45
2.5.1 Atributos de una correcta Especificación de Requerimientos de Software.....	46
2.5.2 Requisitos Funcionales .....	47
2.5.3 Requisitos no Funcionales .....	48
2.5.4 Identificación de los Casos de Uso del Sistema. ....	50
2.5.4.1 ¿Cómo se llega a los Casos de Uso del Sistema a partir de los Requisitos?.....	50
2.5.4.2 Diagrama de Casos de Uso.....	51
2.5.4.3 Descripción Textual de Caso de Uso: Gestionar Electroencefalograma .....	51
2.5.4.4 Descripción textual caso de uso Gestionar Sujeto .....	55
2.5.4.5 Describir los Actores.....	63
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA .....</b>	<b>64</b>
3.1 INTRODUCCIÓN .....	64
3.2 IMPORTANCIA DEL DISEÑO DE UN SISTEMA. ....	64
3.3 DIAGRAMAS. ....	64
3.3.1 Diagramas de Clases. ....	64
3.3.2 Modelo de Objetos .....	67
3.3.3 Clases Persistentes.....	67
3.3.4 Diagramas de Interacción.....	68
3.3.4.1 Diagramas de Colaboración. ....	68
3.3.4.2 Diagramas de Secuencia.....	69
3.4 ARQUITECTURA Y PATRONES.....	78
3.4.1 Arquitectura.....	78
3.4.2 Diseño Orientado a Objetos .....	78
3.4.3 Concepto de Patrón .....	78
3.4.4 Patrón de Arquitectura.....	78
3.4.5 Patrón de Diseño.....	80
3.4.6 Patrones de diseño en el MVC .....	82
3.4.7 Patrones de asignación de responsabilidades (GRASP) .....	85

3.4.8 Arquitectura Actual del Sistema.....	88
3.5 MODELO DE DESPLIEGUE .....	88
<b>CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.....</b>	<b>90</b>
4.1 INTRODUCCIÓN .....	90
4.2 MODELO DE IMPLEMENTACIÓN .....	90
4.2.1 Diagrama de Componentes.....	90
4.3 CONSIDERACIONES FINALES .....	92
4.3.1 Prototipo de Interfaz .....	92
4.3.2 Pasos a Seguir para su utilización.....	94
<b>CONCLUSIONES.....</b>	<b>95</b>
<b>RECOMENDACIONES.....</b>	<b>96</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>97</b>
<b>BIBLIOGRAFÍA.....</b>	<b>99</b>
<b>ANEXOS .....</b>	<b>103</b>
<b>GLOSARIO.....</b>	<b>115</b>

## INTRODUCCIÓN

Cuando se concluye el programa del genoma humano, que fuera en su tiempo un desafío colosal para la humanidad y hoy una gran hazaña, se presenta entonces para los científicos otro reto de alcance similar: “el Proyecto de Mapeo Cerebral Humano”. El hecho de hacer un mapa de cuáles son las diferentes funciones del cerebro tanto de la superficie cortical como de los núcleos internos es en realidad un logro para la ciencia moderna. [1]

De tarea idílica se clasifica el proceso de recolección de información personal sobre individuos vinculados al estudio realizado por nuestros especialistas. Estos procedimientos se realizan de modo manual y posteriormente son archivados de forma digital en una base de datos.

El objetivo principal de este proyecto a nivel internacional, es crear una gigantesca base de datos que incluya toda la información del cerebro conocida hasta la fecha. Los científicos tratan de agrupar lo obtenido, organizarlo y crear así, un mapa cartográfico de las distintas regiones del cerebro a niveles tanto químicos como biológicos, así como su relación con las enfermedades. [2]

El Consorcio Internacional para el Mapeo Cerebral (ICBM, por sus siglas en inglés) fue creado en 1993 con el apoyo del Instituto Nacional de Salud de los EE.UU. (NIH, por sus siglas en inglés) con el objetivo de organizar y desarrollar este proyecto a escala mundial. [3]

Este consorcio está compuesto en su núcleo fundamental por 4 centros de investigación, ellos son:

- UCLA, University of California, Los Ángeles.
- Montreal Neurologic Institute.
- University of Texas at San Antonio.
- Institute of Medicine at Juelich & Heinrich Heine University.

Adicionalmente se han creado otros centros para la adquisición de datos en Asia y Europa, que contribuyen con esta entidad.

El ICBM ha analizado los cerebros de aproximadamente 7000 personas en 9 países. Se destacan las investigaciones realizadas a:

- 174 gemelos en EE.UU.
- 305 adultos en Canadá.
- 738 adultos en Japón.
- 504 niños de 4 a 18 años en EE.UU.

Los estudios realizados en el proyecto se basan en la adquisición de neuroimágenes por diferentes métodos, Resonancia Magnética (RM), Resonancia Magnética Funcional (fMRI, por sus siglas en inglés), Tomografía Axial Computarizada (TAC) y Tomografía por Emisión de Positrones (PET, por sus siglas en inglés).

Algunos de los resultados clínicos más destacados de estos estudios son:

- Descubrimiento del adelgazamiento progresivo de la corteza cerebral que acompaña la aparición de la esquizofrenia en la adolescencia.
- Hallazgo de cambios degenerativos precoces que anuncian la aparición de la enfermedad de Alzheimer.
- Demostración de las alteraciones cerebrales permanentes que acompaña al maltrato en niños pequeños.
- Esclarecimiento de los cambios estructurales cerebrales en niños autistas.

Cuba a pesar de ser un país bloqueado se ha integrado al proyecto debido a la experiencia de los especialistas del Centro de Neurociencias de Cuba (CNEURO, en lo adelante) en el procesamiento de neuroimágenes.

El proyecto cubano no sólo ha introducido técnicas novedosas, sino también es el primero a nivel internacional en incorporar el mapeo de la actividad eléctrica cerebral, a través del Electroencefalograma (EEG). (3)

CNEURO ha venido coordinando el Proyecto Cubano de Mapeo Cerebral Humano. Su principal objetivo es la creación de un ATLAS acerca de la estructura y el funcionamiento del cerebro de los cubanos, aportando conocimientos sobre los sustratos neurales en condiciones normales y los cambios que estos sufren ante enfermedades neuropsiquiátricas.

Para llevar a cabo el proyecto Mapeo Cerebral Humano Cubano (MCHC) se tomó como muestra poblacional inicialmente las personas del municipio La Lisa, provincia Ciudad de La Habana debido a la heterogeneidad social que según estudios estadísticos existen en este territorio, además se considera una muestra representativa de numerosos lugares del país y la cercanía al centro hospitalario donde se encuentra instalado el equipo de Resonancia Magnética. Se pretende, en un futuro, llevar el proyecto a todo el país para realizar así una mejor normalización de los resultados y obtener un atlas que sea representativo de la población cubana.

Algunos de los objetivos que el MCHC persigue lograr son los siguientes:

- Crear un Atlas Cuantitativo del desarrollo cerebral normal de los cubanos.
- Determinar posibles influencias genético-ambientales sobre el cerebro.
- Observar bases cerebrales sobre las diferencias individuales que se presentan en el cerebro teniendo en cuenta la edad, el sexo, el nivel educacional y el coeficiente de inteligencia entre otros factores.
- Detectar las diferencias en los cerebros de sujetos de los estudios especiales con relación al Atlas.
- Desarrollar métodos de pesquisa temprana de desórdenes cerebrales.
- Crear bases para ensayos terapéuticos rigurosos para el tratamiento de disfunciones cerebrales.
- Crear herramientas cuantitativas que permitan la investigación activa y la evaluación de tratamientos.
- Comparar los datos obtenidos con los del Proyecto Internacional.

En el cumplimiento de estas metas están involucradas un grupo de instituciones entre las que se encuentran: La Dirección Municipal de Salud Lisa, los policlínicos de la Lisa: Aleida Fernández y Elpidio Beróvidez, el Centro de Neurociencias de Cuba, el Hospital CIMEQ, el Centro Nacional de Genética Médica: CNGM y la Universidad de las Ciencias Informáticas.

El rango de los ciudadanos encuestados fluctúa entre los 8 y 81 años. Aunque en la actualidad la muestra se enmarca en el intervalo 15-68 años. A continuación se muestra el flujo de trabajo del proyecto MCHC.

- Selección aleatoria de una muestra de la población a partir de los listados de los carné de identidad.
- Captación de los sujetos por parte de las enfermeras comunitarias y firma del consentimiento informado.
- Selección para el estudio mediante el llenado de un cuestionario que establezca que el sujeto es sano. Obtención de la muestra de sangre para estudios genéticos y una evaluación médica completa.
- Evaluación psicocognitiva incluyendo, por primera vez en el país, la aplicación completa de los tests de inteligencia WAIS-III. Además de otras pruebas como el Examen Físico-Neurológico,

Cuestionario de Normalidad, Mini Entrevista Neuropsiquiátrica Internacional y Cuestionario de Manualidad.

- Realización de un estudio de electroencefalograma con un MEDICID de 64 ó 128 canales para estudiar la fisiología cerebral mediante la Tomografía Eléctrica Cerebral (TEC).
- Realización de un estudio de Resonancia Magnética Cerebral Anatómica con el fin de ver la integridad de estructuras cerebrales.
- Realización de un estudio de Resonancia Magnética Cerebral Funcional de la difusión del agua para evaluar conexiones Cerebrales.

Impulsado por el objetivo de crear herramientas cuantitativas que permitan la investigación activa y la evaluación de tratamientos, surgió el Proyecto Informático de Mapeo Cerebral Humano Cubano encargado de obtener, procesar y gestionar toda la información que generan los estudios médicos que forman parte del proyecto. Como parte de este proyecto informático se creó el Sistema para la Gestión de la Información de los Exámenes Clínicos.

Los Exámenes Clínicos, mediante el conjunto de subpruebas antes mencionadas (Cuestionario de Manualidad, Cuestionario de Normalidad, Examen Físico-Neurológico y Mini Entrevista Neuropsiquiátrica Internacional), tienen la responsabilidad de validar la condición sana del individuo y evaluarlo psicocognitivamente.

Este trabajo debe su objeto, a la avidez por parte de CNEURO de solucionar las dificultades gestoras en el procesamiento de esa información, o en otras palabras, humanizar el trabajo que hasta ahora han venido realizando. En este proceso la información que resulta de cada estudio es recogida en una base de datos Microsoft Access, diseñada por los propios especialistas del centro, que hasta el momento les ha solucionado el problema, pero que, debido al elevado volumen de información que recoge y a la alta carga de transacciones a la que está sometida, hoy resulta ineficiente. La misma fue sometida a un proceso de evaluación por parte de nuestro equipo de desarrollo, el cual detectó problemas como: incorrecta identificación de los campos de las tablas, ausencia de relación entre las tablas relacionadas con los estudios, los que conllevó a una alta redundancia en los datos almacenados, falta atomicidad en campos específicos, todo esto esclareciendo la ausencia de normalización en cualquiera de sus formas. Esta disyuntiva dificulta la migración de los datos hacia el gestor seleccionado utilizando métodos y herramientas convencionales. Además, a esta situación habría que agregarle el hecho de que el actual Sistema de Gestión de la Información de Exámenes

Clínicos, no cuenta en su base de datos con la tabla correspondiente a los exámenes de Electroencefalogramas, por lo que aún pudiendo realizar la migración, esta sería incompleta .

### **Problema Científico:**

De esta manera surge el problema científico a resolver: ¿Cómo solucionar el proceso de migración de datos de Microsoft Access a PostgreSQL dentro del PCMCH?

### **Objeto de Estudio:**

Como objeto de estudio dentro del problema abordado tiene: La migración de Datos de Microsoft Access a PostgreSQL.

### **Campo de Acción**

El objeto de estudio delimita su campo de acción: El Proceso de Migración de la información resultante de los Exámenes Clínicos en el PCMCH.

### **Objetivo General**

Para darle solución al problema se propone como objetivo general: Desarrollar un Sistema Informático basado en la arquitectura cliente-servidor que permita la migración de la información resultante de los Exámenes Clínicos en el PCMCH.

### **Objetivos Específicos**

Del objetivo general se derivan los siguientes objetivos específicos:

- ✓ Definir los requerimientos del sistema informático.
- ✓ Diseñar el sistema informático.
- ✓ Implementar el diseño del sistema informático.

## **Tareas**

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

- ✓ Identificación de los requerimientos funcionales y no funcionales del sistema.
- ✓ Revisión de los posibles sistemas informáticos que puedan brindar una solución semejante a la que se requiere.
- ✓ Selección de las herramientas apropiadas para el análisis, diseño e implementación del sistema informático.
- ✓ Selección de la metodología de desarrollo de software adecuada para la gestión del sistema informático.
- ✓ Realización del diseño del sistema.
- ✓ Migración de la base de datos de Microsoft Access a PostgreSQL.

De esta manera se simplificará la labor de los especialistas, que serán los usuarios finales.

El trabajo consta de: introducción, 4 capítulos, conclusiones, recomendaciones, referencias bibliográficas, bibliografía y anexos. El presente documento se divide en varios capítulos:

### **Capítulo 1:** Fundamentación Teórica.

El presente aborda las principales tendencias sobre este estudio en el mundo, lo explica en brevedad y señala el trabajo desarrollado en el país hasta la fecha. Se describen tecnologías y metodologías empleadas para solucionar el problema.

### **Capítulo 2:** Características del sistema.

En este capítulo se trata la definición del objeto de estudio del problema. Se describirán los procesos a automatizar. Se describirá la propuesta que se piensa del sistema, así como el funcionamiento. Los requerimientos son especificados (tanto funcionales como no funcionales). Se identifican los casos de uso del sistema con su descripción y relación correspondiente con los actores del sistema.

### **Capítulo 3:** Diseño del sistema

Aquí se evidencia la especificación que describe el comportamiento del sistema implementado basado en sus requisitos, a través del diseño. Se realizan los diagramas de clases del diseño y diagramas interacción más significativos. Se explica la arquitectura utilizada y los principales patrones de diseño utilizados.

### **Capítulo 4:** Implementación

En el capítulo se exponen los artefactos e información necesaria para llevar a cabo la implementación del sistema. Se realiza el modelo de despliegue y los diagrama de componentes.

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

### **1.1 Introducción**

En este capítulo se realiza una descripción general del objeto de estudio. Además se estudia el comportamiento de las tendencias actuales en el uso de aplicaciones similares en el mundo y en Cuba. Se describen también las técnicas, metodologías y herramientas utilizadas para dar solución al problema con el fin de justificar su uso para el desarrollo del proyecto.

### **1.2 Migración de la Información.**

#### **1.2.1 Definición de Migración**

Hablando en términos lingüísticos, el mayor número de definiciones de la palabra “migración”, están relacionadas a los aspectos demográficos y poblacionales existentes.

La **RAE** define:

1. f. emigración.
2. f. Acción y efecto de pasar de un país a otro para establecerse en él. Se usa hablando de las migraciones históricas que hicieron las razas o los pueblos enteros.
3. f. Viaje periódico de las aves, peces u otros animales migratorios.
4. f. Desplazamiento geográfico de individuos o grupos, generalmente por causas económicas o sociales.

El **Diccionario de Uso del Español de América Vox © Larousse** la define como:

1. Nombre femenino.
2. Movimiento de población que consiste en dejar el lugar de residencia para establecerse en otro país o región, generalmente por causas económicas o sociales: migraciones internas; las migraciones son abundantes en épocas de grave crisis económica.
3. Viaje que las aves, los peces y otros animales emprenden cada cierto tiempo por exigencias de la alimentación o la reproducción: las distancias cubiertas por los patos en sus migraciones varían con la especie.

**El Diccionario Sinónimos y Antónimos Vox © Larousse Editorial S.L. 2007 dice:**

1. Sustantivo femenino
2. emigración, inmigración, transmigración, éxodo, presencia, llegada.
3. Toda migración supone una emigración, o salida del país de origen, y una inmigración, en el país de llegada. Transmigración y éxodo equivalen a emigración, y se usan especialmente cuando esta es colectiva.

La **Gran Enciclopedia Aragonesa** establece que:

Migración es un movimiento espacial de la población que desarraiga temporal o definitivamente a los hombres de su lugar de origen (emigración) para introducirlos en otro país o región (inmigración).

1. Teniendo en cuenta las causas de la migración se pueden distinguir dos tipos:
  - Las forzadas, que resultan impelidas desde el poder.
  - Las voluntarias, que obedecen al derecho inalienable de la persona humana a elegir el lugar más idóneo para el desarrollo de su existencia.
2. Una segunda clasificación de las migraciones, más geográfica, se refiere al alcance espacial de las mismas: migración de la montaña al valle, del secano al regadío, del campo a la ciudad, intrarregionales, interregionales e internacionales.
3. Una tercera clasificación se fundamenta en el tiempo en que dure la migración. Estas pueden ser temporales o definitivas.

Hasta este momento y luego de leer todas estas definiciones se puede decir que “migración” no es más que el desplazamiento condicionado de personas o animales de un lugar a otro. El hombre durante el proceso de evolución ha generado a lo largo de la historia un inmenso volumen de información, el cual se ha encargado de guardar de forma segura para poder consultarla en cualquier momento; el Papiro en Egipto; las Tablillas Enceradas del mundo grecorromano; el Códice, que fue el primero de los libros manuscritos, realizado con hojas de papiro dobladas y cosidas; el Pergamino hecho de piel animal convertida en hojas; y el papel con mas de 2000 años de historia.

Pero con el transcurso de los años y el avance de la tecnología unido al enorme incremento de los volúmenes de información, el hombre ha tenido la necesidad de trasladar los conocimientos que guardaba a formas de almacenamiento que le permitieran acumular mayores volúmenes de datos

en el menor espacio físico posible. El desarrollo que han alcanzado las ciencias computacionales y de los medios de almacenamiento masivos, tanto magnéticos como ópticos le han permitido lograr este objetivo. Sin embargo estos sistemas con el tiempo también necesitan ser cambiados a otros nuevos. Es por esto que también surge otro tipo de migración, que tiene lugar en la **informática**.

- Se denomina migración al proceso mediante el cual los datos y las aplicaciones de una computadora o sistema se traspasan a otro.
- Traslado de una aplicación de un ordenador a otro en condiciones de compatibilidad. Migrar es también elevar una versión de un producto software a otra de más alto nivel, o bien el movimiento de una arquitectura a otra, por ejemplo, de un sistema centralizado a otro con una estructura basada en el modelo cliente/servidor.

### **1.3 ¿Qué es el Software Libre?**

Para entender los beneficios del Software Libre, si inicia explicando que “software” son los programas que permiten que la computadora desarrolle funciones y procedimientos, por ejemplo, la manipulación de información que el hombre realiza pero en forma digital, hay software que permite llevar el manejo financiero de una empresa (contabilidad, tesorería, presupuesto, inventario, cuentas por pagar y cobrar), suite ofimáticas (procesador de palabras, hoja electrónica, presentaciones, agenda), juegos, graficadores, visores de dibujo, vídeo, música (mp3), virus, antivirus, etc. En términos generales, cuando alguien necesita adquirir un software específico existen cuatro alternativas del tipo de software:

- Software Comercial
- Software Shareware
- Software Freeware
- Software de Licencia Pública General (Software Libre)

#### **1.3.1 Software Comercial**

Es aquel por el que se debe pagar para utilizarlo; más que el software, se está comprando la licencia para poder emplearlo y se deben comprar tantas licencias como equipos a utilizar dicho software. Ejemplos de software comercial son los productos de Microsoft, como el sistema operativo Windows

y familia (9x, Me, XP), Sistema Manejador de Bases de Datos, programas para servicios de Internet y sus productos para el manejo de oficina (Suite Ofimática), para el caso Microsoft Office.

En nuestro país, los software comerciales más utilizados son los creados por Microsoft, actualmente sólo por conceptos de licencias se tienen que pagar 119.00USD por el Sistema Operativo Windows XP Profesional, y 255.00USD por la Suite de Microsoft Office 2007.

Otra de las desventajas del software comercial, es que no se puede modificar el programa (el código fuente no está disponible) y sólo el fabricante puede realizar modificaciones; los clientes dan sugerencias de nuevas características que desean agregar a los programas y ellos (los fabricantes) deciden si las hacen (en caso de considerarlas convenientes) o no.

Cuando salen nuevas versiones se debe pagar más dinero si se desea actualizar la versión del programa adquirido.

### **1.3.2 Software Shareware**

Son los programas que se pueden utilizar libremente sin ningún problema de licencia por un tiempo determinado. Normalmente utilizan este sistema las casas productoras de software para que los usuarios conozcan sus productos y si les gusta lo compren. Estas versiones de Shareware la mayoría de las veces no viene con todas las funciones activas.

Uno de los programas Shareware más usados es el Winzip, que nos ayuda a comprimir/descomprimir en distintos formatos de compresión de archivos. Winzip tiene un costo de US \$29 y el periodo de prueba es por 21 días.

El software Shareware tiene los mismos intereses que el software comercial, con la diferencia de que este se puede probar antes de adquirirlo, en caso que el software no cumpla con las expectativas del usuario final, este se debe eliminar para no tener problemas de licenciamiento con los fabricantes. De igual forma que el software comercial, el Shareware no permite modificaciones/adiciones de características nuevas por parte del usuario al programa.

### **1.3.3 Software Freeware**

Son programas que pueden ser utilizados sin ningún problema de licenciamiento, no hay que comprar licencias, pero, se sigue limitado a que las modificaciones en el software sólo pueden ser realizadas por el fabricante del producto.

Uno de los programas Freeware más usado es la suite ofimática OpenOffice de la Sun Microsystems que le hace una fuerte competencia a la suite ofimática de Microsoft, con la ventaja de no tener problemas de licenciamiento.

#### **1.3.4 Software de Licencia Pública General (GPL). Software Libre**

Software libre (en inglés free software) es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software; de modo más preciso, se refiere a cuatro libertades de los usuarios del software: la libertad de usar el programa, con cualquier propósito; de estudiar el funcionamiento del programa, y adaptarlo a las necesidades; de distribuir copias, con lo que puede ayudar a otros; de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (para la segunda y última libertad mencionadas, el acceso al código fuente es un requisito previo).

El concepto general del software libre se basa en la defensa al derecho de la libre expresión, tener criterio para pensar decirlo y defender el punto de vista buscando el bien común, son cualidades muy importantes para destacar en un ser humano, el software libre nos permite pensar y actuar según nuestro criterio ya que se puede modificar o complementar, claro está, siguiendo las reglas o lineamientos requeridos, esto es garantía del potencial del que se dispone para crear e innovar.

El software libre está rodeado de muchos mitos, la mayoría de ellos impulsados desde el mundo empresarial, que por supuesto no tiene un planteamiento imparcial sobre este tema. Lo paradójico es que aquellos a los que más beneficia económicamente son los mismos que lo atacan: empresas de informática, en particular, y de tecnología, en general, han hecho creer la idea de que desarrollar software y entregarlo de manera que los usuarios puedan modificarlo, copiarlo o distribuirlo sin cargo alguno es loable, pero en ningún modo útil para una sociedad que depende de la innovación, la mejora continuada de los productos, un mantenimiento fiable y un nivel de calidad que dé la confianza suficiente como para permitir las inversiones millonarias requeridas para poner en marcha las nuevas tecnologías. Esto lo demuestra el hecho de que el 60% de los servidores Web del mundo utilicen el Apache bajo la licencia GPL.

¿Es posible que el software libre desarrolle productos innovadores y de calidad que se mantengan y mejoren a través del tiempo? No sólo es posible, sino que el mundo tal y como se conoce hoy no

existiría sin él. En contra de lo que el sector de la economía tradicional se empeña en hacer creer, los avances revolucionarios no suelen proceder de las empresas que trafican con la tecnología, sino que surgen del flujo libre de información, que en el caso de la informática consiste en compartir los formatos, los protocolos y los códigos.

### **1.3.5 Beneficios del Software Libre.**

#### **En la Ética:**

El Software Libre tiene sus bases en una ideología que dice el software no debe tener dueños, es un asunto de libertad: la gente debería ser libre de usarlo en todas las formas que sean socialmente útiles.

De esta forma, el movimiento del Software Libre pone lo que es beneficioso para la sociedad por encima de los intereses económicos o políticos.

Entre los beneficios que percibe la sociedad se pueden mencionar:

- Tecnologías transparentes, confiables y seguras.
- Tecnologías como bien público.
- Promoción del espíritu cooperativo, en el que el principal objetivo es ayudar a su vecino.
- Transferencia del conocimiento para el desarrollo de los países Subdesarrollados.
- Contribuye con la reducción de gastos.
- También hay soporte y cursos de entrenamiento y certificación.
- Precios justos.

#### **En la Práctica**

El Software Libre ofrece a las personas la posibilidad de utilizar, estudiar, modificar, copiar y redistribuir el software. Para hacer efectivas estas libertades, el código fuente de los programas debe estar disponible.

Gracias a estas libertades se obtienen muchos beneficios prácticos:

- Ejecutar el software en el momento y forma deseada.
- Aprender de los programas existentes.
- Permite mejorarlos.
- Adaptarlos y ajustarlos a necesidades específicas.
- Reutilización.

- Permite formar negocios alrededor de la creación, distribución, soporte y capacitación de programas libres.

Y el efecto de todos estos beneficios es la formación de Comunidades enormes alrededor de proyectos de software libre, gracias a las cuales es posible acceder a desarrolladores, documentadores y probadores de todo el mundo.

Un estudio realizado sobre la distribución de Linux Red Hat indica que el sistema para el 2001 contenía 30 millones de líneas de código fuente, y que su desarrollo hubiera costado \$1000 millones y 8000 años persona, siguiendo las etapas tradicionales del desarrollo de software propietario.

#### **1.4 Importancia de la Migración hacia Software Libre para Cuba.**

Ningún Estado debería estar sometido tecnológicamente a otro. Ningún Estado debería estar sometido tecnológicamente a otro y más cuando es un país bloqueado económicamente por el gobierno de los EEUU. El software libre contribuye a la igualdad entre los pueblos al permitir el libre acceso de todos a la Sociedad del Conocimiento.

Si una empresa u organismo de un país basa su informatización en software “cerrado” (aquel cuyo código fuente le es desconocido), ¿qué garantías tiene de que los programas hagan únicamente lo que se espera de ellos? ¿Cómo tener la seguridad de que no hay procesos ocultos o defectos que menoscaben la privacidad de la información? Y si son computadoras conectadas a Internet, ¿no es aún más arriesgado no saber cómo funciona el programa de correo ó el navegador web, por ejemplo?

Para los países en vía de desarrollo es un factor limitante el excesivo costo de las licencias de los sistemas operativos que suelen utilizar la mayoría de las computadoras (Windows98/NT/2000/XP). Además, está el precio de las licencias de los programas específicos (MS Office, Corel Draw, Adobe Photoshop, SAP, etc.). También hay que tener en cuenta que cada licencia sólo puede ser utilizada en una única computadora. El precio final del software está, por tanto, en función del número de computadoras del que se dispone. Esta inversión tampoco es para toda la vida, ya que el ciclo de vida del software es muy corto.

Desde abril del 2004 el Consejo de Ministros de la República de Cuba acordó que resultaba de vital importancia que el país migrara sus sistemas de información a software libre, y aunque no se puso una fecha tope, se orientó que debía ser un proceso continuo y organizado. A partir de ese momento

se creó entonces el Grupo Ejecutivo Nacional encabezado por el Ministro de la Informática y las Comunicaciones. Como institución pionera en la aplicación de estas medidas se encuentra la Aduana General de la República, donde en la actualidad solo cuentan con Sistema Operativo Windows el 5% de la maquinas. La acompañan en este camino los Ministerios de Informática y Comunicaciones (MIC), Educación Superior (MES) y Cultura (MINCULT), así como la Empresa de Telecomunicaciones de Cuba (ETECSA) y la Universidad de las Ciencias Informáticas (UCI).

#### **1.4.1 Beneficios del Software Libre para Cuba [5]**

##### **1. Independencia Tecnológica**

El Estado deja de depender de terceros (a menudo transnacionales) para el diseño, desarrollo y mantenimiento de sus sistemas de información, retomando el control total de sus procesos, en particular de los procesos críticos y de alta importancia estratégica para el país.

##### **2. Seguridad**

La información que el Estado maneja generalmente es importante y/o confidencial, puede ser muy peligroso que esta información caiga en manos incorrectas. Por esta razón es imprescindible que el Estado pueda verificar que su software no tenga puertas de entrada traseras, voluntarias o accidentales, y que pueda cerrarlas en caso de encontrarlas; tal control sólo es posible con el software libre.

##### **3. Razones Económicas**

Ahorros importantes al liberarse del pago de licencias y especialmente por la replicación casi gratuita de aplicaciones comunes a toda la administración pública. El muy bajo costo del software permitirá la ejecución de programas de desarrollo.

Al ser excesivo el costo de las licencias de los sistemas operativos cerrados, y si se le suma el número de PC que lo usan, el gasto sería inmensamente grande. Es un lujo que Cuba no podría darse. Por ejemplo, en la UCI, donde existen seis edificios docentes, el costo superaría los 5 millones de dólares al año en una sola de esas instalaciones por concepto de licencias". [6]

##### **4. Control de la información**

El acceso al código fuente, la libertad de inspeccionar el funcionamiento del software, la libertad de decidir la manera en que almacenan los datos y la posibilidad de modificar cualquiera de estos aspectos queda en manos del Estado, lo cual le permite el control total de la información y por consiguiente el ejercicio de la Soberanía Nacional.

#### 5. Confiabilidad y estabilidad

El software libre realizado por comunidades está sometido a la inspección de un importante número de personas, este número de verificadores es mucho mayor que el del software propietario. Estas personas identifican los problemas, los resuelven, y comparten las soluciones con los demás. Por tal razón los programas libres de las comunidades gozan de gran confiabilidad y estabilidad.

#### 6. Desarrollo del País

Se genera Transferencia Tecnológica hacia los actores nacionales productores de software, acelerando el Desarrollo Endógeno y reforzando la Soberanía Nacional.

### **1.5 Herramientas de Migración.**

Después de realizado un estudio de las diferentes herramientas libres existentes en Internet que permiten realizar la migración de bases de datos MS Access a PostgreSQL y analizado las características específicas que presenta el Proyecto Cubano de Mapeo Cerebral Humano, se llega a la conclusión que hoy no existe una herramienta que partiendo de las condiciones que presenta la bases de datos MS Access que recoge la información de los Exámenes Clínicos, realice una conversión eficaz a PostgreSQL sin la pérdida de información. Por lo que se decide realizar una herramienta para la migración de datos que se adapte a las características y necesidades del proyecto, incluyendo además la gestión de electroencefalogramas como condición necesaria y suficiente para poder realizar dicha migración.

### **1.6 Sistemas Gestores de Bases de Datos**

Se define una base de datos como un conjunto de información relacionada que se encuentra agrupada o estructurada en una computadora. Un Sistema Gestor de Bases de Datos (SGBD) o DBMS (Database Management System) es una colección de programas cuyo objetivo es servir de

interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. Existen numerosos SGBD, algunos libres y otros comerciales entre los que se destacan por su uso y prestaciones:

**- Sistemas Comerciales**

- Microsoft Access, Microsoft SQL Server, MAGIC, WindowBase, Progress (DBMS), Oracle, DB2, Sybase.

**-Sistemas Libres**

- PostgreSQL, MySQL, SQLite.

### **1.6.1 Microsoft Access**

Microsoft Access es un sistema gestor de bases de datos relacionales (SGBD). Por formar parte de la Suite de Microsoft Office utiliza un Motor de Bases de Datos propio de esta compañía: Microsoft Jet. Jet es el motor de base de datos de predeterminada de Microsoft Access, así como otras aplicaciones de Microsoft, como Visual Basic, Microsoft Excel, Project y Microsoft Foundation Classes (MFC). Microsoft Access es un sistema de base de datos muy popular, permite a los usuarios controlar y crear informes de datos de forma rápida y sencilla gracias a la interfaz de usuario de Microsoft Office Fluent y funciones de diseño interactivas que no requieren conocimientos especializados de bases de datos.

Microsoft Office Fluente es la interfaz de usuario que Microsoft decidió utilizar en su Suite de Office con el objetivo de facilitar a los usuarios sacar mayor provecho de las aplicaciones que integran la misma, permitiéndoles obtener mejores resultados de una manera más rápida.

Microsoft Access brinda la posibilidad de crear tablas rápidamente sin preocuparse de su complejidad. Da la posibilidad de vincular a una base de datos tablas de otras bases de datos de Access, hojas de cálculo de Excel, sitios de Windows SharePoint Services, orígenes de datos de ODBC, bases de datos de Microsoft SQL Server y otros orígenes de datos. También puede usar estas tablas vinculadas para crear informes fácilmente y basar en ellos sus decisiones sobre un conjunto de datos más general.

Sin embargo se debe tener en cuenta que tiene limitaciones en cuanto a número de accesos simultáneos y tamaño de las tablas manejadas. Por otro lado uno de los principales componentes del Microsoft Jet es la biblioteca MSJET40.DLL, que procesa todas las solicitudes de lectura y escritura de datos en la base de Microsoft Access (archivos MDB). Se ha detectado que MSJET40.DLL en ocasiones no valida correctamente un archivo MDB malformado, ocasionando que el sistema falle por una condición de desbordamiento de búfer.

Debido a estas dificultades en sus 2 últimas versiones se sustituyó este motor por el Microsoft Data Engine (MSDE) que además proporciona almacenamiento de datos local compatible con Microsoft SQL Server 7.0. Por otra parte, tanto el volumen de datos a manejar como el número de visitantes conectados de forma simultánea a la base de datos son pequeños, posee un bajo nivel de procesamiento de consultas, la seguridad de la base de datos así como la de las transacciones que se realizan es muy pobre por lo que se recomienda utilizarla en aplicaciones donde las operaciones sean de solo lectura, no permite la administración remota de la base de datos, así como tampoco el uso de Funciones, Triggers y Procedimientos Almacenados. Además sólo está diseñado para Sistemas Operativos Windows en sus distintas versiones.

### **1.6.2 MySQL**

MySQL es un sistema de administración de bases de datos relacional (RDBMS). Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en una red corporativa. Para agregar, acceder a y procesar datos se necesita un administrador como MySQL Server. Debido a la importancia de manejar grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

#### **Características:**

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el

mayor número de operaciones disponibles.

- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

### 1.6.3 PostgreSQL

PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977. El desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo, dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group). Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre como DB2 u Oracle, entre estas se encuentran:

- DBMS Objeto-Relacional

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multiversión, soporte multiusuario, transacciones, optimización de consultas, herencia, y arrays.

- Altamente Extensible

PostgreSQL soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.

- Soporte SQL Comprensivo

PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.

- Integridad Referencial

PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

- API Flexible

La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.

- Lenguajes Procedurales

PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL ("Tool Command Language" o lenguaje de herramientas de comando) como lenguaje procedural embebido.

- Cliente/Servidor

PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

- MVCC

MVCC, o Control de Concurrencia Multiversión (Multi-Version Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios. Si alguna vez ha usado algún DBMS con capacidades SQL, tal como MySQL o Access, probablemente habrá notado que hay ocasiones en las que una lectura tiene que esperar para acceder a información de la base de datos. La espera está provocada por usuarios que están escribiendo en la base de datos. Resumiendo, el lector está bloqueado por los escritores que están actualizando registros.

- Write Ahead Logging (WAL)

La característica de PostgreSQL conocida como Write Ahead Logging incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en el hipotético caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual se puede restaurar la misma. Esto puede ser enormemente beneficioso en el caso de caída, ya que cualesquiera cambios que no fueron escritos en la base de datos pueden ser recuperados usando el dato que fue previamente registrado. Una vez el sistema ha quedado restaurado, un usuario puede continuar trabajando desde el punto en que lo dejó cuando cayó la base de datos.

#### **1.6.4 ¿Por qué usar PostgreSQL?**

El proyecto MCHC gestionará un gran volumen de información por lo que se escoge para el desarrollo del sistema al SGBD PostgreSQL por ser un poderoso sistema manejador de bases de datos, es decir, un sistema diseñado para administrar grandes cantidades de datos, el cual es considerado como una de las bases de datos de código abierto (Open Source) más avanzada del mundo. PostgreSQL se ha preocupado por ser una solución real a los complejos problemas del mundo empresarial y a la vez mantener la eficiencia al consultar los datos. Con ese fin, se han desarrollado y añadido a PostgreSQL las más interesantes y útiles características que antes sólo podían hallarse en sistemas manejadores de bases de datos comerciales como Oracle, DB2 o Sybase. Tiene además un alto rendimiento, estable y capacitado para lidiar con grandes volúmenes de datos. Estas características de PostgreSQL son las que garantizan la integridad de los datos, la velocidad de acceso y consultas a la base de datos.

El software libre ha demostrado ser un modelo válido para el desarrollo y la innovación; la adopción de sistemas y protocolos abiertos es la base en la que se asienta el más importante desarrollo tecnológico de las últimas décadas, Internet; y muchas de las críticas vertidas contra él parten de actores con unos intereses económicos que no permiten que sean imparciales. Por todo lo hablado se puede concluir que las revoluciones, al menos en informática, surgen de la capacidad de manipulación y combinación que los formatos y códigos abiertos ofrecen a los investigadores. Sólo así es posible llegar a una

tecnología cuyo potencial desarrollo no dependa únicamente de las leyes del mercado sino también de la capacidad creativa del ser humano, la verdadera impulsora de la innovación [7]

PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre. PostgreSQL se ha preocupado por ser una solución real a los complejos problemas del mundo empresarial y a la vez mantener la eficiencia al consultar los datos, un alto rendimiento, estabilidad y capacidad para lidiar con grandes volúmenes de datos, le garantizan un lugar cimero entre el resto SGBD libres y propietarios.

## **1.7 Procesos de Desarrollo de Software**

El proceso de desarrollo de software es difícil de controlar, surge entonces la necesidad de tener una metodología que garantice cumplir con los planes de producción del software y la satisfacción del cliente con el cumplimiento de los objetivos del proyecto. Los desarrolladores muchas veces cuando se enfrentan a un nuevo proyecto se preguntan ¿Qué metodología usar para un correcto proceso de desarrollo del software? La respuesta correcta se tendrá después de analizar la situación del proyecto, de saber a lo que se enfrentan y lo que deben producir. Entre las metodologías que más se utilizan a nivel mundial existen dos corrientes de importancia: los denominados métodos pesados y métodos ligeros. Entre las más reconocidas se encuentran el Proceso Unificado de Desarrollo (RUP), exponente de la metodología pesada; Extreme Programming (XP) y Open Unified Process (OpenUP) como exponentes de las metodologías ligeras o ágiles.

### **1.7.1 Extreme Programming (XP)**

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizadas para proyectos de corto plazo, equipo y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

#### **Características.**

Entre las características fundamentales de este proceso de desarrollo se tienen:

- Pruebas unitarias: esta característica está basada en realizar pruebas a los procesos principales para poder detectar posibles errores que puedan suceder.
- Refabricación: esta característica se basa en la reutilización de código lo que le permite ser más flexible al cambio.
- Programación en pares: esta característica se basa en que dos desarrolladores participen en un proyecto en una misma estación de trabajo con el propósito de lograr un mismo objetivo, la satisfacción del cliente.

### **¿Qué es lo que propone XP?**

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.

Esta metodología permite la comunicación entre los desarrolladores y el cliente, simplifica el desarrollo de módulos y posibilita una retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales. No existe documentación del proyecto lo que más se acerca a la documentación son las historias de usuario, pero al concluir el proyecto se descartan. Inclusive se recomienda hacer dos secciones, una con todas las historias de usuario que faltan desarrollar, y otra donde se archiven las concluidas, esto aproximará el estado de avance del proyecto.

### **1.7.2 Proceso Unificado de Desarrollo (RUP)**

Es un proceso bien definido, estructurado y adaptable a las características y necesidades de cada proyecto específico. RUP es un proceso iterativo e incremental que se encarga de dividir el trabajo en partes más pequeñas o en mini proyectos. Este proceso se divide en 4 fases para el desarrollo de software.

- Inicio: El Objetivo en esta etapa es determinar la visión del proyecto.

- **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción:** En esta etapa el objetivo es obtener la capacidad operacional inicial.
- **Transición:** El objetivo es llegar a obtener la liberación del proyecto.

RUP hace uso de arquitectura basada en componentes, permite el modelado visual del software y la verificación de la calidad del software, además pretende implementar las mejores prácticas en Ingeniería de Software. RUP brinda un proceso integrado que utiliza el estándar de notación UML para permitir desarrollar un proceso de forma iterativa e incremental a partir de la identificación e implementación de los casos de uso.

### **Flujos de trabajo:**

- **Modelado del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Instalación:** Produce el lanzamiento del producto realizando actividades de empaque, instalación y asistencia a los usuarios finales del software.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.

RUP es uno de los procesos más generales de los existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier proyecto.

### **1.7.3 Open Unified Process (OpenUP)**

Open Unified Process (OpenUP) es un proceso de desarrollo unificado que está basado en Rational Unified Process (RUP). Mantiene las mismas características de RUP, que contiene el desarrollo iterativo, casos de uso y escenarios de conducción de desarrollo, gestión de riesgos, y el enfoque centrado en la arquitectura.

OpenUP es una parte del Eclipse Process Framework (EPF), un proceso de código abierto desarrollado en el marco de la organización Eclipse de fuente abierta. Ofrece las mejores prácticas de una variedad de desarrollo de software líderes de pensamiento y de la más amplia comunidad de desarrollo de software que cubren un conjunto diverso de perspectivas y necesidades de desarrollo.

El OpenUP es un proceso mínimo y suficiente, lo que significa que sólo el contenido fundamental y necesario es incluido. Por lo tanto no provee lineamientos para todos los elementos que se manejan en un proyecto pero tiene los componentes básicos que pueden servir de base a procesos específicos. La mayoría de los elementos de OpenUP están declarados para fomentar el intercambio de información entre los equipos de desarrollo y mantener un entendimiento compartido del proyecto, sus objetivos, alcance y avances. Se aplica en enfoques tanto iterativo como incremental. Este proceso puede ser desarrollado para hacer frente a una extensa diversidad de tipos de proyectos.

Existe una forma más básica y fácil de manejar de OpenUP que es OpenUP/Basic, objetivos más pequeños y con sede equipos interesados en el desarrollo ágil e iterativo.

#### **Áreas de interés**

Los elementos de OpenUP dirigen la organización del trabajo en los niveles personal, de equipo y de interesados. El proyecto en general se divide en iteraciones, las cuales son planificadas en un intervalo definido de tiempo que no superan las pocas semanas. OpenUP tiene elementos que ayudan a los equipos de trabajo a enfocar los esfuerzos a través del ciclo de vida de cada iteración de tal forma que se puedan distribuir funcionalidades incrementales de una manera predecible, una versión totalmente probada y funcional al final de cada iteración.

OpenUP estructura el ciclo de vida de un proyecto en 4 fases: concepción, elaboración, construcción y transición. El ciclo de vida del proyecto provee a los interesados un mecanismo de supervisión y

dirección para controlar los fundamentos del proyecto, su ámbito, la exposición a los riesgos, el aumento de valor y otros aspectos.

#### **1.7.4 ¿Por qué usar OpenUP?**

Después de un análisis de las metodologías anteriormente expuestas se determinó que XP no es recomendable para este proyecto porque la misma exige la integración total del cliente al equipo de trabajo, condiciones con las cuales no se cuenta, ya que el cliente es un agente externo al equipo de trabajo. Debido a que el módulo Examen Clínico del Proyecto Mapeo Cerebral requiere de un tiempo relativamente corto de desarrollo se decide optar por la metodología OpenUP/Basic basada en RUP. Es una metodología ágil que se aplica a proyectos de corta duración, diseñado para pequeños equipos de trabajo, generalmente de 3-6 personas y de 3-6 meses de esfuerzo de desarrollo. El mismo preserva las características esenciales de RUP, como: estar centrado en la arquitectura y estar dirigido por Casos de Uso, y es iterativo e incremental, pero con la diferencia que las partes opcionales del RUP fueron obviadas, surgiendo como resultado un proceso muy simple que cumple con los principios fundamentales de la metodología RUP.

#### **1.7.5 Lenguaje de modelado.**

Se utilizó el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) debido a que la metodología de desarrollo seleccionada lo propone como lenguaje de notación. Además UML es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Está respaldado por el OMG (Object Management Group).

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de

software (tal como el Proceso Unificado de Rational) pero no especifica en sí mismo qué metodología o proceso usar.

### **1.7.6 Mapeo Relacional de Objeto (ORM)**

El mapeo relacional de objetos (más conocido por su nombre en inglés, Object Relational Mapping, o sus siglas ORM) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos o se puede definir como una interfaz que traduce la lógica de los objetos a la lógica relacional. En la práctica esto crea una base de datos orientada a objetos virtual, por sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).

La principal ventaja que aporta el ORM es la reutilización, permitiendo llamar a los métodos de un objeto de datos desde varias partes de la aplicación e incluso desde diferentes aplicaciones. La capa ORM también encapsula la lógica de los datos. La utilización de objetos en vez de registros y de clases en vez de tablas, tiene otra ventaja: permite añadir métodos de acceso en los objetos que no tienen relación directa con una tabla. Si se dispone por ejemplo de una tabla llamada cliente con 2 campos llamados "nombre" y "apellidos", puede que se necesite un dato llamado "NombreCompleto" que incluya y combine el nombre y los apellidos. En el mundo orientado a objetos, es tan fácil como añadir un método de acceso a la clase Cliente. Desde el punto de vista de la aplicación, no existen diferencias entre los atributos Nombre, Apellidos, NombreCompleto de la clase Cliente. Solo la propia clase es capaz de determinar si un atributo determinado se corresponde con una columna de la base de datos.

Existen soluciones que proveen un mapeo básico O/R para PHP. A continuación se identifican algunos proyectos similares que existen actualmente en varios escenarios de desarrollo para PHP:

- ALYOOP - un simple PHP5 DAO framework
- DB\_DataContainer - una solución PEAR-compliant O/R.
- DB\_DataObject - Solución PEAR oficial de mapeo O/R con generación de código PHP.
- DB\_Table - herramienta para abstraer una tabla de base de datos de Paul M. Jones.
- EasyORM – una solución de objetos persistentes únicamente para MySQL.
- Metastorage - Una de las capas más comprensivas persistentes para PHP, incluye generación de clases PHP.
- Propel - Es un servicio de objeto persistente y de consulta para PHP.

### **1.7.5.1 Propel**

Propel es un servicio de objeto persistente y de consulta, lo que significa que provee un sistema para almacenar objetos en una base de datos y un sistema para búsqueda y restauración de objetos desde una base de datos. Propel le permite realizar consultas complejas y manipulación de bases de datos sin escribir una sola cláusula SQL con un mínimo costo de realización para su aplicación en PHP. Esta herramienta le permite hacer más fácil la escritura de aplicaciones, más fácil de desplegar, y mucho más fácil para migrar si alguna vez la situación lo amerita.

Puede ser descrito como un mapeado objeto-relacional, una capa DAO, o una capa objeto persistente, optimizado para el lenguaje PHP. También genera las clases para cada tabla que exhibe algunas de las propiedades de la tabla del patrón datos de entrada. Una tabla de entrada de datos almacena todo el SQL para acceder a una sola tabla o vista: selecciones, inserciones, actualizaciones, y eliminaciones. Otro código llama los métodos para todas las interacciones con la base de datos. En Propel las clases de tabla de entrada de datos son llamadas clases Peer, mientras las clases de filas de entrada de datos son llamadas entidad o clases objeto.

El ambiente de ejecución proporciona una capa de abstracción y encapsulación de bases de datos para reglas de lógicas de negocios. Las clases Propel representan la capa modelo del tradicional patrón Modelo-Vista-Controlador, diseñado para encapsular cualquier nivel de validación de dato necesitado por su aplicación. Propel es un proyecto de software libre.

Después de un estudio de las características de Propel se decidió usar esta herramienta como puente entre el sistema y la Base de Datos ya que nos brinda una abstracción de la lógica relacional a una lógica de objeto permitiendo la interacción con las bases de datos sin importar el SGBD utilizado. Además nos permite utilizar todas las ventajas del paradigma de Programación Orientado a Objeto como son la herencia y el polimorfismo.

## **1.7.6 Lenguaje de programación.**

### **1.7.6.1 PHP 5**

PHP es un lenguaje de programación usado generalmente para la creación de contenido para sitios Web. PHP es un acrónimo recurrente que significa "PHP Hypertext Pre-processor" (inicialmente PHP Tools, o Personal Home Page Tools), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web. PHP5 es la

última versión de dicho lenguaje; esta incorpora el paradigma de la Programación Orientada a Objetos (POO) e incluye todas las ventajas que provee el nuevo Zend Engine 2. Es un lenguaje multiplataforma. Tiene capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad como MySQL, PostgreSQL, mSQL, Oracle, dbm, filepro, Hyperwave, Informix, InterBase y Sybase. Posee una amplia documentación en su página oficial, La curva de aprendizaje del lenguaje es baja. Posee una biblioteca nativa de funciones sumamente amplia. Es libre y Open Source, por lo que se presenta como una alternativa de fácil acceso para todos además no se está forzado a pagar actualizaciones anuales para tener una versión que funcione.

Se decidió por parte de la dirección del proyecto utilizar php ya que, teniendo en cuenta las características mencionadas anteriormente que lo hacen muy potente para la implementación de aplicaciones en ambiente web, el equipo de desarrollo posee un mayor conocimiento de este lenguaje. Además la entrega del sistema debía realizarse en un período relativamente corto.

#### **1.7.6.2 JavaScript**

JavaScript es un lenguaje de programación del lado del cliente ampliamente utilizado en el mundo del desarrollo web por ser muy versátil, compatible con la mayoría de los navegadores modernos, y potente, tanto para la realización de pequeñas tareas como para la gestión de complejas aplicaciones.

JavaScript es un lenguaje interpretado que, al contrario de las aplicaciones normales, que son ejecutadas por el sistema operativo, es ejecutado por el navegador para ver las páginas. Eso permite desarrollar aplicaciones de muy diverso tipo, desde generadores de HTML, comprobadores de formularios, etc. hasta programas que gestionen las capas de una página. Pueden desarrollarse incluso aplicaciones que permitan poder tener capas en una página como si fueran ventanas, y dar la sensación de estar trabajando con una aplicación con interfaz de ventanas.

Con JavaScript se pueden crear efectos especiales en las páginas web y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica.

Entre las acciones típicas que se pueden realizar en JavaScript se cuenta con dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, JavaScript permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se pueden crear páginas interactivas con programas como calculadoras, agendas, tablas de cálculo, calendarios y validar informaciones entradas por el usuario para verificar su veracidad o su forma de escribirse.

### **¿Qué se puede hacer con JavaScript?**

- Páginas dinámicas (DHTML).
- Comprobación de datos (Formularios).
- Uso de los elementos de una página web.
- Intercambiar información entre páginas web en distintas ventanas.
- Manipulación de gráficos, texto, etc.
- Comunicación con plug-ins: Flash, Java, Shockwave, etc.

### **1.7.7 Herramientas CASE**

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas son de gran ayuda en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

Las Herramientas CASE se plantean los siguientes objetivos:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.
- Mejorar el tiempo y coste de desarrollo y mantenimiento de los sistemas Informáticos.
- Mejorar la planificación de un proyecto.

- Automatizar, desarrollo del software, documentación, generación de código, pruebas de errores y gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación.
- Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

Existen herramientas Case de trabajo visuales como DBDesigner4, Rational Rose, Embarcadero ER/Studio, GNU Ferret, MagicDraw, Umbrello, ArgoUML y Visual Paradigm que permiten realizar el modelado del desarrollo de los proyectos. A continuación se explican algunas de la herramientas CASE más utilizadas.

### **1.7.7.1 Visual Paradigm for UML**

Visual Paradigm for UML una herramienta multiplataforma que permite proporcionar un ambiente modelado visual para satisfacer la tecnología del software de hoy y necesidades de comunicación. Es muy fácil de usar y presenta un ambiente gráfico agradable para el usuario. Utiliza UML como lenguaje de modelado para la construcción de los sistemas software ofreciendo soluciones de software que permiten a las organizaciones desarrollar las aplicaciones de calidad más rápido, bien y más barato. Tiene la capacidad de ejecutarse sobre diferentes sistemas operativos lo que le confiere la característica de ser multiplataforma, está disponible para Windows, Linux, MacOS X, etc. Integra diferentes funcionalidades para el desarrollo de aplicaciones como el modelado de UML, el modelado de base de datos, el modelado de requerimientos, el modelado del proceso de negocio, la interoperabilidad con otras herramientas CASE, la generación de documentación entre otros. Permite generar código en lenguajes tan utilizados como: Java, C++, PHP, C #, Delphi, Perl.

Permite incorporar los dibujos de Visio en cualquier diagrama de UML. Muchas herramientas CASE realizan solo diagramas de UML normales, Visual Paradigm posibilita modelar hardware, dominio-específico, el software, conectando una red de computadoras los componentes usando sus iconos, más allá de las anotaciones de UML normales.

Como el equipo de desarrollo trabaja sobre la plataforma GNU/Linux y no cuenta con una versión de Rational Rose que corra sobre dicha plataforma se usará para el desarrollo de la aplicación la herramienta CASE Visual Paradigm, software del que se tiene licencia en la Universidad de las Ciencias Informáticas.

### **1.7.7.2 DBDesigner Fork**

DBDesigner Fork, como su nombre lo dice, es un fork de fabFORCE DBDesigner 4 que es un sistema de diseño visual de bases de datos que integra diseño, modelado, creación y mantenimiento de estas en un único entorno para MySQL, pero el proyecto quedó descontinuado. DBDesigner Fork partió del código existente y a partir del mismo ha creado un diseñador visual de bases de datos que integra la creación de bases de datos y permite modelar y especificar las relaciones entre las entidades. Genera scripts para Oracle, SQL Server, MySQL, PostgreSQL y FireBird.

Con acabado profesional, el programa dispone de infinidad de opciones y funcionalidades, además de ser bastante intuitivo y fácil de utilizar. En su web se dice que esta solución es comparable a otras, mucho más caras, como Oracle's Designer, IBM's Rational Rose o Computer Associates's ERwin, entre otras.

DBDesigner Fork tiene características interesantes como la posibilidad de poder hacer ingeniería inversa: generar el modelo ER (Entidad-Relación) a raíz de una base de datos existente, también permite cosas como la sincronización del trabajo con servidores de bases de datos (para ir aplicando los últimos cambios que haces sobre el modelo relacional directamente a su base de datos) y dispone de dos modos principales: el Modo diseño (Design Mode) con el que crear / mantener visualmente el modelo de la base de datos; y el Modo Consulta (Query Mode), completa interfaz con la que poder trabajar con los datos de las tablas y construir consultas SQL. Además, los datos los guarda con estructura XML pudiendo trabajar con ellos a través de otras aplicaciones. Y, finalmente, dispone de un sistema de plug-ins de fácil manejo / desarrollo para expandir y personalizar el manejo de esta herramienta trayendo algunos de serie; uno de ellos, SimpleWebFront, permite generar una aplicación CRUD completa de tu diseño en PHP sin tocar una sola línea de código.

El listado de características a destacar de DBDesigner Fork es el que sigue:

- Modo diseño / Modo consulta.
- Ingeniería inversa para MySQL, Oracle, MSSQL y cualquier base de datos ODBC.
- Sincronización modelo visual / base de datos.
- Colocación automática de clave ajena.
- Soporte de entidad débil.
- Posibilidad de exportar el modelo como imagen.
- Soporte integro con MySQL (tipos de datos, etc.)
- Tipos de datos de usuario definidos.

- Posibilidad de guardar el modelo independientemente de su base de datos correspondiente.
- Control de versiones.
- Manejador SQL.
- Manejador de plug-ins.

Además es un sistema de código libre (GNU/GPL) y se pueden obtener sus ejecutables desde Internet.

Para el modelado y diseño de la Base de Datos del sistema se decidió optar por DBDesigner fork debido a que es una herramienta libre y sus características satisfacen las necesidades del equipo de desarrollo.

### **1.7.8 Editores PHP**

Los editores PHP son muy utilizados por desarrolladores de aplicaciones web para escribir programas en este lenguaje. Algunos de los editores que se encuentran para el lenguaje PHP son:

Zend Studio, PHP Designer, NuSphere, PHP Coder, DzSoft PHP Editor, Kumbia

#### **1.7.8.1 Zend Studio**

Este Entorno Integrado de Desarrollo (IDE por sus siglas en inglés) no es libre pero son muchos los desarrolladores de aplicaciones web que trabajan con Zend Studio. Es un programa de la casa Zend, uno de los mayores impulsores de PHP, orientado a desarrollar aplicaciones web para dicho lenguaje. ZendStudio es un editor de texto para páginas PHP que proporciona un buen número de ayudas desde la creación y gestión de proyectos hasta la depuración del código.

Existen versiones del programa para diversas plataformas debido a que está escrito en Java. Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración.

La Universidad de las Ciencias Informáticas posee una licencia de uso por lo que se opta por su utilización debido a las ventajas que ofrece, como un excelente completado de código y la interrelación con el lenguaje a utilizar (PHP).

### **1.7.9 Editores de páginas Web**

Un editor de páginas Web es una aplicación diseñada con el fin de facilitar la creación de documentos HTML o XHTML. Su complejidad puede variar desde la de un simple editor de texto plano, entornos WYSIWYG. Existen disimiles editores web. Dentro de los más destacados se pueden mencionar: Adobe Dreamweaver, BlueFish, Quanta Plus, SCREAM, KompoZer, Microsoft Expression Web.

#### **1.7.9.1 Dreamweaver**

Es un editor WYSIWYG de páginas Web, creadas por Adobe (anteriormente por Macromedia). Es uno de los programas más utilizado en el sector del diseño y la programación Web, por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium. Su principal competidor es Microsoft FrontPage. Tiene soporte tanto para edición de imágenes como para animación a través de su integración con otras herramientas. Dreamweaver permite al usuario utilizar la mayoría de los navegadores Web instalados en su ordenador para previsualizar las páginas Web. También dispone de herramientas de administración de sitios dirigidas a principiantes como, por ejemplo, la habilidad de encontrar y reemplazar líneas de texto y código por cualquier tipo de parámetro especificado, hasta el sitio Web completo. El panel de comportamientos también permite crear JavaScript básico sin conocimientos de código. Con la llegada de la versión MX, Macromedia incorporó herramientas de creación de contenido dinámico en Dreamweaver. En lo fundamental de las herramientas HTML WYSIWYG, también permite la conexión a Bases de Datos como MySQL y Microsoft Access, para filtrar y mostrar el contenido utilizando tecnología de script como, por ejemplo, ASP (Active Server Pages), ASP.NET, ColdFusion, JSP (JavaServer Pages), PHP sin necesidad de tener experiencia previa en programación. Es una herramienta de software propietario y el precio de la licencia de uso es de \$399.

## **Conclusiones del Capítulo.**

En el capítulo se hace un análisis de las tendencias actuales y se proponen las soluciones que más se ajustan para el desarrollo del sistema. El software libre ha demostrado ser un modelo válido para el desarrollo y la innovación. Por lo que resulta positivo y necesario el uso de dichas tecnologías en el desarrollo de este sistema de software. Además se determinan las tecnologías y herramientas que se utilizarán en el desarrollo del sistema. Como metodología para guiar un adecuado proceso de desarrollo del software se utilizará OpenUp, que utiliza como lenguaje de modelado UML y la herramienta CASE Visual Paradigm. Como IDE de desarrollo para la implementación de la aplicación Zend Studio y lenguaje de programación interpretado PHP 5, para almacenar adecuadamente la información generada se utilizará PostgreSQL como sistema gestor de bases de datos. Quanta plus para el diseño de las interfaces web y DBDesigner fork para el diseño de la Base de Datos del sistema.

## **CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA**

### **2.1 Introducción**

En este capítulo se trata la definición del objeto de estudio del problema. Se describirán los procesos a automatizar. Se describirá la propuesta que se tiene del sistema, así como su funcionamiento. Los requerimientos son especificados (tanto funcionales como no funcionales). Se identifican los casos de uso del sistema con su descripción y relación correspondiente con los actores del sistema.

### **2.2 Descripción de los Procesos a automatizar**

El Centro de Neurociencias de Cuba (CNEURO) es la institución encargada de llevar a cabo el Proyecto Cubano de Mapeo Cerebral Humano. En este proceso la información que resulta de cada estudio es recogida en una base de datos Microsoft Access, diseñada por los propios especialistas del centro, que hasta el momento les ha solucionado el problema, pero que, por el elevado volumen de información que recoge y la carga de transacciones a la que está sometida, hoy resulta ineficiente. Los especialistas del centro tienen la necesidad real de solucionar las dificultades gestoras en el procesamiento de dicha información para poder continuar con la ejecución del proyecto.

La base de datos fue sometida a un proceso de evaluación por nuestro equipo de desarrollo, el cual detectó problemas como: ausencia total de relaciones entre las tablas, lo que trae consigo redundancia en los datos almacenados, falta de atomicidad en los campos específicos, esclareciendo la ausencia de normalización. Esta disyuntiva dificulta la migración de los datos hacia el gestor seleccionado utilizando métodos y herramientas convencionales. Además, a esta situación habría que agregarle el hecho de que el actual Sistema de Gestión de la Información de Exámenes Clínicos no cuenta en su base de datos con la tabla correspondiente a los Electroencefalogramas, por lo que aún pudiendo realizar la migración, esta sería incompleta.

Resulta de vital importancia la incorporación al Sistema de Gestión de la Información de Exámenes Clínicos, la Gestión de Electroencefalogramas y por consiguiente la adición a su base de datos de la tabla correspondiente a este estudio para poder realizar una migración efectiva de los datos que se encuentran almacenados. La combinación de los resultados de los exámenes clínicos con los de los

electroencefalogramas proporciona importantes puntos de apoyo en el diagnóstico de la mayoría de los síndromes epilépticos, de ahí la importancia que tiene este proceso.

## 2.3 Procedimiento para Migrar la BD

**Nombre:** Guía para migrar a PostgreSQL la base de datos de Exámenes Clínicos del Proyecto CMCH.

**Objetivos:**

- Minimizar los problemas de lógica para evitar las dificultades gestoras en el procesamiento de la información
- Simplificar la estructura de la base de datos para un mejor acceso a la misma.
- Disminuir el consumo de espacio de almacenamiento a partir de una menor repetición de los datos almacenados.

**Pasos:**

- Preparar la BD para la migración: Se crea la tabla correspondiente al electroencefalograma para garantizar la completitud de los datos de los estudios realizados al sujeto.

**Fig.1** Página donde se insertan los datos del electroencefalograma.

eeg		
<b>+examen_sujeto_idsujeto</b>	<b>int4</b>	<b>Nullable = false</b>
<b>+examen_idexamen</b>	<b>int4</b>	<b>Nullable = false</b>
equipo_registro	varchar(0)	Nullable = true
numero_canales	int4	Nullable = true
canales_defectuosos	varchar(0)	Nullable = true
frecuencia_muestreo	int4	Nullable = true
eeg_hicut	int4	Nullable = true
eeg_locut	int4	Nullable = true
eeg_ganancia	int4	Nullable = true
eeg_normal	varchar(0)	Nullable = true
informe_clinico	varchar(0)	Nullable = true
nweeg_a	int4	Nullable = true
nweeg_b	int4	Nullable = true
nweeg_c	int4	Nullable = true
nweeg_d	int4	Nullable = true
nweeg_e	int4	Nullable = true
nweeg_f	int4	Nullable = true
eeg_path	varchar(0)	Nullable = true
fecha_insercion	date	Nullable = true

- Selección de un estudio: Se introduce el código del sujeto que se desea gestionar y se obtiene un listado de los exámenes realizados y no realizados.

**Fig.2** Formulario para buscar el estudio de acuerdo al código del sujeto.

Escoja una opción

Buscar sujeto por nombre

Buscar sujeto por código

Nombre o Código

Aceptar

**Fig.3** Formulario donde se gestionan los exámenes del sujeto.

Código del sujeto

Exámenes realizados	Exámenes sin realizar
	<input type="radio"/> Cuestionario Inicial
	<input type="radio"/> MINI Entrevista Neuropsiquiátrica Internacional
	<input type="radio"/> Cuestionario de normalidad
	<input type="radio"/> Cuestionario de manualidad
	<input type="radio"/> Examen físico-neurológico
	<input type="radio"/> Examen psicométrico
	<input checked="" type="radio"/> Electroencefalograma

- Verificar la completitud de los datos: El usuario verifica que los datos introducidos estén completos y que se correspondan con los de la planilla del sujeto.
- Validar los datos: El sistema se encarga de comprobar que los datos introducidos por el usuario sean correctos.

Para terminar el proceso de de migración se aceptan las acciones realizadas con anterioridad.

## **2.4 Propuesta de mejoras al Módulo de Examen Clínico.**

### **2.4.1 Gestión de EEG**

La Gestión de Electroencefalogramas se encargará de la inserción, visualización, modificación y eliminación de los datos de EEG correspondientes a los exámenes realizados al sujeto en estudio.

#### ***2.4.1.1 Definición de Electroencefalograma***

La Electroencefalografía es el registro y evaluación de los potenciales eléctricos generados por el cerebro y obtenidos por medio de electrodos situados sobre la superficie del cuero cabelludo. [8] Los impulsos nerviosos transmitidos por las neuronas son de naturaleza electroquímica

El electroencefalograma (EEG) es el registro de la actividad eléctrica de las neuronas del encéfalo. Dicho registro posee formas muy complejas que varían mucho con la localización de los electrodos y entre individuos. Esto es debido al gran número de interconexiones que presentan las neuronas y por la estructura no uniforme del encéfalo. [9] Los impulsos eléctricos producidos por la actividad cerebral, son generados en forma de ondas alfa (despierto y relajado), beta (concentrado), delta (dormido), theta (meditación y pensamiento creativo) y sigue sus variaciones en el transcurso del tiempo. La presencia de ondas normales en un electroencefalograma ayuda a diagnosticar epilepsias, tumores y otras alteraciones neurológicas.

#### ***2.4.1.2 Electrogénesis cerebral***

El tejido nervioso presenta como una de sus funciones básicas la capacidad de generar potenciales eléctricos que son la base de la excitabilidad del organismo. Para comprender la forma en que se generan estos potenciales es preciso un conocimiento de la estructura y las conexiones de aquellas partes del cerebro que los originan. En rigor, todo el sistema nervioso posee capacidad electrogénica. Sin embargo, para los propósitos del EEG bastará con considerar la corteza cerebral y las regiones directamente relacionadas con ella.

### **2.4.1.3 Electrogénesis cortical**

Un fragmento de tejido cortical aislado es asiento de actividad eléctrica espontánea. Esta actividad se caracteriza por salvas de ondas lentas sobre las que se superponen ritmos rápidos. Entre una salva y otra aparecen períodos de silencio eléctrico.

Estas señales son producidas como consecuencia de la actividad sináptica general de regiones discretas de tejido: los PPSE (potenciales postsinápticos excitadores) y los PPSI (potenciales postsinápticos inhibidores) se suman entre si y dan origen a potenciales lentos que son las ondas registradas. Una de estas porciones de tejido capaz de producir actividad eléctrica se llama un **GENERADOR**.

Se han puesto de manifiesto tres generadores corticales:

**Generador A:** Situado a unas 500 micras de la superficie cortical está producido por la despolarización de las dendritas apicales de las células piramidales. Su actividad produce ondas negativas en la superficie de la corteza. No tiene relación con la descarga de potenciales de acción de las células.

**Generador B:** Situado a 900 micras de profundidad está formado por las despolarizaciones de los somas de las células piramidales. Produce ondas positivas en la superficie cortical y su actividad coincide con la aparición de potenciales de acción en las células.

**Generador C:** Está situado también a 900 micras, pero su actividad determina ondas negativas en la superficie cortical y es el resultado de la hiperpolarización de las células. Su actividad coincide con una interrupción de la descarga de potenciales de acción en las células piramidales.

De forma general, una tensión positiva en la superficie cortical traduce una despolarización en las capas más profundas de la corteza. En cambio, una tensión negativa puede ser resultado, bien de una despolarización superficial, o de una hiperpolarización profunda.

#### **2.5.1.4 Sincronización de la actividad celular**

De lo dicho anteriormente, las señales corticales son consecuencia de la actividad neuronal. Sin embargo, dado que en un registro normal se recoge la actividad de muchos miles de neuronas, para poder conseguir una actividad global mínima es preciso que las neuronas vecinas se encuentren sincronizadas. Cuando así ocurre, se pueden observar ondas mayores y más lentas, cuanto mayor sea la sincronía de los generadores.

La sincronización se encuentra bajo control de estructuras subcorticales, fundamentalmente ciertos núcleos talámicos que actúan como los marcapasos sincronizadores de las actividades rítmicas corticales. Por el contrario, otras regiones más caudales que van desde el hipotálamo hasta la porción rostral del bulbo constituyen estructuras desincronizadoras.

#### **2.4.1.5 Captación EEG.**

La actividad bioeléctrica cerebral puede captarse por diversos procedimientos:

- Sobre el cuero cabelludo.
- En la base del cráneo.
- En cerebro expuesto.
- En localizaciones cerebrales profundas.

Para captar la señal se utilizan diferentes tipos de electrodos:

- **Electrodos superficiales:** Se aplican sobre el cuero cabelludo.
- **Electrodos basales:** Se aplican en la base del cráneo sin necesidad de procedimiento quirúrgico.
- **Electrodos quirúrgicos:** para su aplicación es precisa la cirugía y pueden ser corticales o intracerebrales.

El registro de la actividad bioeléctrica cerebral recibe distintos nombres según la forma de captación:

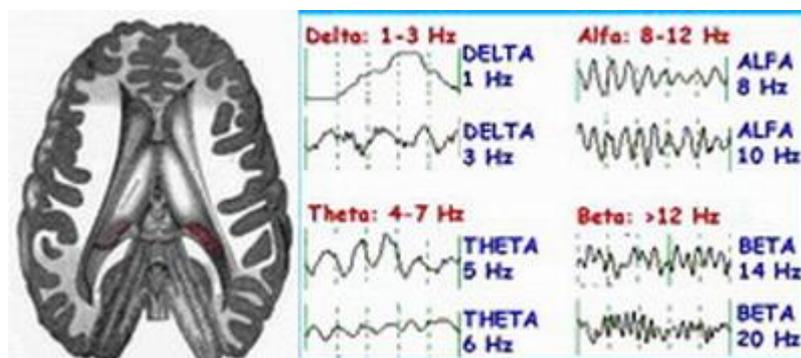
- **Electroencefalograma (EEG):** cuando se utilizan electrodos de superficie o basales.
- **Electrocorticograma (ECoG):** si se utilizan electrodos quirúrgicos en la superficie de la corteza.
- **Estéreo Electroencefalograma (E-EEG):** cuando se utilizan electrodos quirúrgicos de aplicación profunda. [9]

#### 2.4.1.6 Actividad Cerebral

Con el Electroencefalograma se registran ondas, que son producidas por la activación de las neuronas del cerebro. A estas ondas se les da el nombre de una letra griega, según su frecuencia. La frecuencia es la mayor o menor rapidez de las ondas, valora el número de ondas en un segundo, y se mide en hertzios -Hz-, por ejemplo, en una actividad de 8 Hz hay 8 ondas en un segundo. Las frecuencias del EEG se dividen en 4 grupos:

- **delta**, son las más lentas, con un ritmo de 1-3 ondas cada segundo;
- **theta**, de 4 a 7 ondas por segundo;
- **alfa**, de 8 a 12 ondas por segundo;
- **beta**, por encima de 12 ondas por segundo.

Fig. 4: Tipos de Ondas por Regiones del Cerebro donde se generan



Así, se habla de frecuencias lentas (delta y theta), frecuencia alfa y frecuencias rápidas (beta). Además de la mayor o menor rapidez -frecuencia-, interesa valorar el tamaño (amplitud) de las ondas, oscila de pocos microvoltios ( $\mu\text{V}$ ) hasta 500  $\mu\text{V}$  o 1 milivoltio (mV).

Actualmente se realiza el EEG digital, mediante ordenadores que procesan la información y permiten analizarla de distintas maneras, incluso con procesamientos matemáticos.

#### **2.4.1.7 Actividad Patológica del Electroencefalograma**

El Electroencefalograma (EEG) puede presentar una actividad normal o alteraciones y actividades patológicas.

- Actividad normal
- Alteraciones de la actividad de base
- Actividad Focal
- Actividad Paroxística

#### **2.4.1.8 Actividad Normal**

La actividad normal del electroencefalograma se describe como ondas alfa en las regiones posteriores (occipitales). Se puede comprobar estando despiertos, relajados y con los ojos cerrados, y desaparece al abrirlos o al concentrarse en una tarea.

#### **2.4.1.9 Alteraciones de la Actividad de Base**

En condiciones de estrés no se aprecia la actividad alfa, por lo que es importante estar relajado durante la exploración. En las situaciones patológicas suele encontrarse enlentecida (ondas lentas delta y theta).

#### **2.4.1.10 Actividad Focal**

Consiste en brotes de actividad de cualquier frecuencia (delta, theta, alfa, beta), generalmente ondas lentas (delta y theta), que aparecen localizadas en un área determinada del cerebro (frontal, parietal, temporal, occipital).

Además de su frecuencia y localización, la actividad focal se puede describir según su **morfología** (aguda, variable, monomorfa), su **ritmicidad** (rítmica, intermitente), **amplitud**, **propagación** y **persistencia**.

#### **1.5.1.11 Actividad Paroxística**

Consiste en la actividad que se encuentra en los casos de **epilepsia**. Corresponde a una despolarización brusca de un grupo de neuronas. Las descargas paroxísticas pueden ser **localizadas** o **generalizadas**, **rítmicas** o **arrítmicas**, y tienen un gran valor para diagnosticar los distintos tipos de epilepsia.

#### **2.4.1.12 Técnicas de realización del Electroencefalograma**

Existen diversas técnicas para obtener un Electroencefalograma (EEG).

- EEG de Vigilia.
- EEG en privación del sueño.
- Tests de latencias múltiples.
- Polisomnograma Diurno.
- Otras técnicas.
- Videomonitorización.
- Holter EEG. (ambulatorio).
- Cartografía cerebral.
- Análisis espectral.
- EEG funcional.
- Magnetoencefalografía.

#### **2.4.1.13 Prescripción del Electroencefalograma**

El EEG se prescribe para el estudio de:

- Epilepsia.

- Maduración en la infancia: prematuros, neonatos, infantil.
- Determinados casos de Cefalea.
- Deterioro cognitivo, trastornos de memoria y comportamiento, demencia.
- Meningitis, encefalitis.
- Traumatismo cráneo-encefálico.
- Anoxia cerebral (tras parada cardíaca...).
- Coma.

#### **2.4.1.14 Conclusiones Parciales**

El electroencefalograma valora la actividad eléctrica cerebral. El desarrollo de las modernas técnicas informáticas hace que actualmente sea una exploración útil y necesaria para el estudio de diversas patologías, entre las que destacan el retraso madurativo en la infancia, el deterioro cognitivo, las meningitis y encefalitis, el traumatismo craneoencefálico, la anoxia cerebral, el coma y de forma muy especial la epilepsia.

Después de conocer las características del EEG y los beneficios que reporta su utilización para determinar un gran número de enfermedades se pudiera pensar que este es el método ideal a utilizar en el Proyecto Cubano de Mapeo Cerebral Humano para realizar estudios neurofuncionales en los pacientes. Sin embargo, tiene un problema: se registra fuera de la cabeza, en el cuero cabelludo, y no tiene la precisión de la resonancia magnética ni su profundidad en tres dimensiones

Los especialistas de CNEURO han desarrollado una variante llamada **Tomografía Eléctrica Cerebral (TEC)**, consistente en combinar la información del electroencefalograma con la de la resonancia magnética anatómica. Así se obtiene una imagen en tres dimensiones de profundidad acerca de la activación de los distintos sitios neuronales, lo cual facilita la evaluación de la función cerebral

## **2.5 Captura de los requisitos del nuevo sistema.**

Una vez identificadas las funcionalidades de la herramienta se procede a realizar la captura de los requisitos del sistema.

### 2.5.1 Atributos de una correcta Especificación de Requerimientos de Software.

Davis, A (1993), propone lo que una Especificación de Requerimientos de Software (ERS) debe cumplir para ser perfecta:

1. **Correcta.** Una ERS es correcta, si y sólo si, cada requerimiento especificado se construye en el sistema.
  
2. **Sin ambigüedades.** Una ERS no tiene ambigüedades, si y sólo si, cada requerimiento especificado tiene una sola interpretación.
  
3. **Completa.** Una ERS está completa si posee las siguientes cuatro cualidades:
  - 3.1. Todo lo que se supone que el software debe hacer está incluido en la ERS.
  - 3.2. Se incluye la definición de todas las respuestas de cada entrada de datos en todas las diferentes situaciones.
  - 3.3. Todas las páginas, figuras y tablas están numeradas y son referenciadas; todos los términos y unidades de medida son explicados y todos los materiales requeridos y secciones son presentados.
  - 3.4. No incluye secciones marcadas como pendientes.
  
4. **Verificable.** Una ERS es verificable, si y sólo si, todo requerimiento en el documento es verificable. Esto es, que cada requerimiento tenga una forma finita, efectiva y no costosa tal que una persona o la misma máquina pueda verificar que el software construido empate con él.
  
5. **Consistente.** Una ERS es consistente si y sólo si cumple con los siguientes dos aspectos:
  - 5.1. Ningún requerimiento entra en conflicto con requerimientos incluidos en documentos anteriores.
  - 5.2. Ningún subconjunto de requerimientos entra en conflicto interno.
  
6. **Entendible por el cliente.** La ERS debe escribirse evitando en lo posible los formalismos. Aunque la falta de formalismos puede llevar a ambigüedades y en ocasiones es necesario incluirlos,

se debe tener en mente que quien leerá la ERS es, en la mayoría de los casos, una persona no especialista en computación.

**7. Modificable.** Una ERS es modificable si su estructura y estilo son tales que permiten realizar cambios a los requerimientos, fácil, completa y consistentemente.

**8. Trazable hacia atrás.** Una ERS es trazable hacia atrás si para cada requerimiento se sabe su origen; esto es, existe un soporte (persona o documento) que lo originó y que pertenece a las etapas de desarrollo anteriores.

**9. Trazable hacia delante.** Una ERS es trazable hacia delante si está escrita de tal forma que se facilita la referencia a cada requerimiento individual por las actividades de las etapas posteriores (diseño, prueba).

**10. Independiente del diseño.** Una ERS es independiente del diseño si no involucra arquitectura de software específica o algoritmos.

**11. Anotada.** Una ERS debe asignar a cada requerimiento tanto su necesidad relativa como su estabilidad relativa dentro del sistema. Con tales anotaciones para cada requerimiento se dará prioridad al cumplimiento de ellos respecto a los tiempos de entrega.

**12. Concisa.** Dadas dos ERS del mismo sistema en que cada una contiene los mismos elementos de calidad anteriores, la más breve es la mejor.

**13. Organizada.** Una ERS es organizada si los requerimientos contenidos en ella son fáciles de localizar. Esto involucra el hecho de que los requerimientos que están agrupados deben estar cerca entre sí.

### 2.5.2 Requisitos Funcionales

Los Requerimientos Funcionales son capacidades o condiciones que el sistema debe cumplir.

## **1. Migrar Datos.**

- 1.1 Seleccionar Sujeto.
- 1.2 Verificar Exámenes del Sujeto.
- 1.3 Validar Exámenes del Sujeto.

## **2. Gestionar Electroencefalograma**

- 2.1 Buscar EEG.
- 2.2 Mostrar EEG.
- 2.3 Insertar EEG.
- 2.4 Modificar EEG.
- 2.5 Eliminar EEG.

## **3. Gestionar Sujeto**

- 3.1 Buscar Sujeto.
- 3.2 Mostrar Sujeto.
- 3.3 Insertar Sujeto.
- 3.4 Modificar Sujeto.
- 3.5 Eliminar Sujeto.

### **2.5.3 Requisitos no Funcionales**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades se ven como las características que hacen al producto atractivo, usable, rápido o confiable.

### **Apariencia o interfaz externa**

El sistema tendrá un diseño de interfaz sencillo y fácil de entender. Las páginas de la aplicación no se cargarán con mucha información y contendrán sólo las imágenes necesarias.

### **Seguridad**

El sistema debe contar con varios niveles de acceso para permitir el trabajo organizado. En el caso del módulo Examen Clínico solo tendrá acceso el técnico y el administrador. Se debe garantizar que la información sensible sólo pueda ser vista por los usuarios con el nivel de acceso adecuado y que las funcionalidades del sistema se muestren de acuerdo al usuario que esté activo.

### **Usabilidad**

El sistema debe permitir a los usuarios un acceso fácil y rápido. Podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de una computadora y de un ambiente Web.

### **Confiabilidad**

El sistema será usado y administrado solamente por trabajadores del Centro de Neurociencias. El sistema validará la entrada de datos para evitar entradas inadecuadas.

### **Soporte**

Una vez instalado el sistema se impartirá un curso de asesoramiento y familiarización con el mismo. Además se hará un seguimiento del funcionamiento del software.

### **Portabilidad**

El sistema debe adaptarse a diferentes ambientes sin necesidad de usar otros medios que los previstos. Al sistema se debe acceder a través de navegadores desde la plataforma GNU/Linux o Microsoft Windows.

## **Hardware**

Se requiere de una PC cliente de al menos 128 MB de RAM. Se deberá contar con impresora en las computadoras clientes que interactúen con la aplicación. El servidor debe tener las siguientes características: capacidad de disco duro superior a 80.0 GB, microprocesador Pentium IV superior a 2.0 GHz y como mínimo 1.0 GB de RAM.

## **Software**

Se deberá disponer, para instalar la aplicación, del Sistema Operativo Windows 98 o superior, o cualquier distribución de Linux. Las computadoras clientes de los usuarios accederán al sistema usando uno de los siguientes navegadores: Internet Explorer o Mozilla FireFox. Para el servidor de la aplicación el sistema operativo recomendado es GNU/Linux. Se debe instalar un servidor Web Apache2. Para la Base de Datos del sistema se usará PostgreSQL.

## **Restricciones en el diseño y la implementación**

La lógica de presentación constituirá una capa independiente de la lógica de negocio, centrando su función en la interfaz de usuario y validaciones de los datos de entrada. Se usará el lenguaje de programación PHP 5 y el gestor de bases de datos PostgreSQL.

## **Legales**

Se estarán usando herramientas de software libre, licencia GNU/GPL y herramientas de las que se posea licencia.

### **2.5.4 Identificación de los Casos de Uso del Sistema.**

#### **Caso de Uso:**

Es una operación ó tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

#### **2.5.4.1 ¿Cómo se llega a los Casos de Uso del Sistema a partir de los Requisitos?**

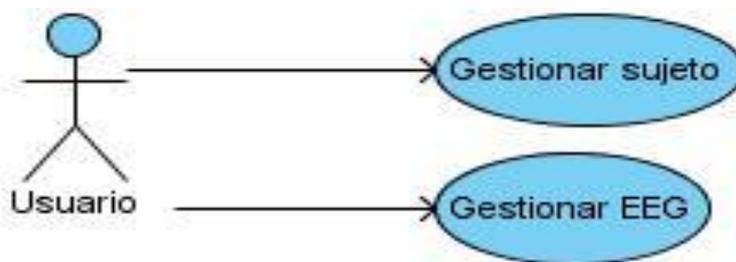
Luego de haber identificado los requerimientos funcionales y no funcionales, los actores del sistema y los posibles escenarios con los que pudieran interactuar estos actores, se procede a la identificación de los casos de uso. Para esto se agrupan en un caso de uso todos los escenarios similares que formen parte de una funcionalidad dada.

Los casos de uso explicitan los requisitos funcionales del sistema relativos a uno de los objetivos del usuario. Estos se denominan también, de manera más precisa, casos de uso con objetivo usuario.

### 2.5.4.2 Diagrama de Casos de Uso

El diagrama de casos de uso representa la forma en cómo un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

Fig.5 Diagrama de Casos de USO



### 2.5.4.3 Descripción Textual de Caso de Uso: Gestionar Electroencefalograma

Nombre del Caso de Uso	Gestionar Electroencefalograma
Actores	Usuario
Propósito	Llevar a cabo la inserción de los datos, búsqueda, modificación, visualización y eliminación de EEG.
Resumen	El caso de uso se inicia cuando el Usuario va a realizar alguna de las siguientes operaciones:

	<ul style="list-style-type: none"> <li>• Insertar datos en el Electroencefalograma.</li> <li>• Buscar Electroencefalograma.</li> </ul> <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.</p>
Referencias	2.1, 2.2, 2.3, 2.4, 2.5
Precondiciones	Que el Usuario esté autenticado en la aplicación.
Poscondiciones	El sistema inserta un nuevo Electroencefalograma o muestra el Electroencefalograma solicitado el cual podrá ser eliminado o actualizado.
Requerimientos especiales	
Flujo normal de los eventos	
Acción del actor	Respuesta del Sistema
<p>1. El Usuario, quiere realizar una de las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>• Insertar Electroencefalograma.</li> <li>• Buscar Electroencefalograma.</li> </ul>	<p>2. El sistema, en dependencia de la operación que se solicita realizar, hace lo siguiente:</p> <ul style="list-style-type: none"> <li>• Si decide Insertar Electroencefalograma, ir a la sección: "Insertar Electroencefalograma".</li> <li>• Si decide Buscar Electroencefalograma ya existente, ir a la sección: "Buscar Electroencefalograma".</li> </ul>
Sección "Insertar Electroencefalograma"	
Acción del actor	Respuesta del Sistema
<p>1. El Usuario inserta los datos correspondientes:</p> <ul style="list-style-type: none"> <li>• Código</li> <li>• Fecha de entrada del EEG</li> <li>• Fecha de registro del EEG</li> <li>• Lugar de Registro de EEG</li> <li>• Equipo de Registro de EEG</li> <li>• Técnico que Registró el EEG</li> <li>• Neurofisiólogo que Revisó el</li> </ul>	<p>2. El sistema valida que todos los campos fueron llenados correctamente.</p>

<p>EEG</p> <ul style="list-style-type: none"> <li>• Número de Canales</li> <li>• Frecuencia de Muestreo</li> <li>• EEG HiCut</li> <li>• EEG LoCut</li> <li>• EEG Ganancia</li> <li>• EEG Normal</li> <li>• Informe clínico del EEG</li> <li>• NWEEG_A</li> <li>• NWEEG_B</li> <li>• NWEEG_C</li> <li>• NWEEG_D</li> <li>• NWEEG_E</li> <li>• NWEEG_F</li> <li>• EEG Canales Malos</li> <li>• Localización EEG</li> </ul> <p>Indica insertar los datos.</p>	
	<p>3. El sistema registra el Electroencefalograma. Se adiciona al estudio del sujeto como un examen realizado.</p> <p>Ir a la acción 1 del flujo normal de eventos de la sección “Gestionar Electroencefalograma”</p>
<p>Flujo alternativo de la Sección “Insertar Electroencefalograma”</p>	
	<p>2.1 Si los campos no fueron llenados correctamente se activa un mensaje de alerta notificando el error.</p> <p>Ir a la acción 1 de la Sección “Insertar Electroencefalograma”</p>
<p>Sección “Buscar Electroencefalograma”</p>	
<p>1. El Usuario inserta el código del Electroencefalograma que desea buscar.</p>	<p>2. El sistema busca el Electroencefalograma indicado.</p>

	3. El sistema muestra los datos del Electroencefalograma indicado dando la posibilidad de modificarlo y eliminarlo.
4. El Usuario escoge una de las siguientes opciones: <ul style="list-style-type: none"> <li>• Modificar Electroencefalograma</li> <li>• Eliminar Electroencefalograma</li> </ul>	5. En dependencia de la opción que se solicitó realizar, el sistema realiza lo siguiente: <ul style="list-style-type: none"> <li>• Si indica Modificar Electroencefalograma ir a la sección “Modificar Electroencefalograma”.</li> <li>• Si indica Eliminar Electroencefalograma el sistema elimina el Electroencefalograma correspondiente. Se actualiza el estado a no realizado en el estudio del sujeto.</li> </ul>
Flujo Alternativo de Sección “Buscar Electroencefalograma”	
4.1 Si el Usuario no desea realizar ninguna de las operaciones sale de la sección.	
Sección “Modificar Electroencefalograma”	
1. El Usuario modifica los datos correspondientes a: <ul style="list-style-type: none"> <li>• Código</li> <li>• Fecha de entrada del EEG</li> <li>• Fecha de registro del EEG</li> <li>• Lugar de Registro de EEG</li> <li>• Equipo de Registro de EEG</li> <li>• Técnico que Registró el EEG</li> <li>• Neurofisiólogo que Revisó el EEG</li> <li>• Número de Canales</li> <li>• Frecuencia de Muestreo</li> <li>• EEG HiCut</li> <li>• EEG LoCut</li> </ul>	2. El sistema valida que todos los campos fueron llenados correctamente.

<ul style="list-style-type: none"> <li>• EEG Ganancia</li> <li>• EEG Normal</li> <li>• Informe clínico del EEG</li> <li>• NWEEG_A</li> <li>• NWEEG_B</li> <li>• NWEEG_C</li> <li>• NWEEG_D</li> <li>• NWEEG_E</li> <li>• NWEEG_F</li> <li>• EEG Canales Malos</li> <li>• Localización EEG</li> </ul> <p>Indica insertar los datos.</p>	
	<p>3. El sistema actualiza los datos del Electroencefalograma. Ir a la</p>
<p>Flujo Alternativo de Sección "Modificar Electroencefalograma"</p>	
	<p>2.1 Si los campos no fueron llenados correctamente se activa un mensaje de alerta notificando el error. Ir a la acción 1 de la Sección "Modificar Electroencefalograma"</p>
Prototipo	Consultar Expediente de Proyecto

**2.5.4.4 Descripción textual caso de uso Gestionar Sujeto**

Nombre del Caso de Uso	Gestionar Sujeto
Actores	Usuario
Propósito	Llevar a cabo una serie de operaciones con los sujetos presentes en el estudio como son: insertar, buscar, visualizar, un sujeto, insertarlo o modificarlo.
Resumen	<p>El caso de uso se inicia cuando el Usuario va a realizar alguna de las siguientes operaciones:</p> <p>2. Insertar Sujeto.</p>

	<p>3.            <b>Buscar Sujeto.</b></p> <p>El sistema le muestra la interfaz correspondiente según su solicitud y ejecuta las acciones necesarias. El caso de uso finaliza cuando se emite el resultado de la operación solicitada.</p>
Referencias	3.1, 3.2, 3.3, 3.4, 3.5
Precondiciones	Que el Usuario esté autenticado en la aplicación.
Poscondiciones	El sistema inserta un nuevo sujeto o muestra los datos de un sujeto solicitado, al cual se le podrán actualizar sus datos.
Requerimientos especiales	-
<b>Flujo normal de los eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
<p>1. El Usuario, quiere realizar una de las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>• Insertar Sujeto.</li> <li>• Buscar Sujeto.</li> </ul>	<p>2. El sistema, en dependencia de la operación que se solicita realizar, hace lo siguiente:</p> <ul style="list-style-type: none"> <li>• Si decide Insertar los datos referentes al Sujeto, ir a la sección: "Insertar Sujeto".</li> <li>• Si decide Buscar un Sujeto ya existente, ir a la sección: "Buscar Sujeto".</li> </ul>
<b>Sección "Insertar Sujeto"</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema muestra la interfaz correspondiente a la inserción de un sujeto.
<p>2. Inserta los datos correspondientes:</p> <p>Datos del Sujeto</p> <ul style="list-style-type: none"> <li>• Código</li> <li>• Nombre</li> <li>• Apellido(1)</li> <li>• Apellido (2)</li> <li>• CI</li> </ul> <p>Datos de Nacimiento</p> <ul style="list-style-type: none"> <li>• Fecha</li> <li>• Edad</li> </ul>	3. El sistema valida que todos los campos fueron llenados correctamente.

<ul style="list-style-type: none"> <li>• Provincia</li> <li>• Municipio</li> </ul> <p>Dirección Actual</p> <ul style="list-style-type: none"> <li>• Dirección</li> <li>• Provincia</li> <li>• Municipio</li> </ul> <ul style="list-style-type: none"> <li>• Policlínico</li> <li>• Teléfono</li> <li>• Sexo</li> <li>• Manualidad(Derecho, Zurdo, Ambidextro)</li> <li>• Color de Piel(Blanca, Negra, Mulata, Otras)</li> <li>• Peso (lbs.)</li> <li>• Talla (cm)</li> <li>• Temperatura (°C)</li> <li>• Nivel de Escolaridad Vencido</li> <li>• Ingreso Per Cápita (suma total de los salarios dividido por número de personas en la vivienda)</li> <li>• Fecha Última extracción de sangre</li> <li>• Status Social</li> <li>• Comentario</li> </ul> <p>Indica insertar los datos.</p>	
	4. El sistema verifica existencia del sujeto.
	5. El sistema registra al sujeto dando la posibilidad de insertar sus exámenes y ver sus datos personales.

<p>6. El Usuario marca una de las siguientes opciones</p> <p>Exámenes no Realizados</p> <ul style="list-style-type: none"> <li>• Examen psicométrico</li> <li>• Cuestionario de normalidad (Datos Personales)</li> <li>• MINI Entrevista Neuropsiquiátrica Internacional</li> <li>• Cuestionario de normalidad (Datos Clínicos)</li> <li>• Cuestionario de manualidad</li> <li>• Examen Físico-Neurológico</li> <li>• Electroencefalograma</li> </ul> <p>Mostrar Datos del Sujeto</p>	<p>7. El sistema en correspondencia con la operación escogida por el Usuario realiza lo siguiente:</p> <p>Si selecciona en los Examen sin realizar</p> <ul style="list-style-type: none"> <li>• Examen psicométrico: Ir al Caso de Uso “Gestionar Examen psicométrico”, a la Sección “Insertar Examen psicométrico”.</li> <li>• Cuestionario de normalidad (Datos Personales): Ir al Caso de Uso “Gestionar Cuestionario de normalidad”, a la Sección “Insertar Cuestionario de normalidad”.</li> <li>• MINI Entrevista Neuropsiquiátrica Internacional: Ir al Caso de Uso “Gestionar MINI Entrevista Neuropsiquiátrica Internacional” a la Sección “Insertar MINI Entrevista Neuropsiquiátrica Internacional”.</li> <li>• Cuestionario de normalidad (Datos Clínicos): Ir al Caso de Uso “Gestionar Cuestionario de normalidad”, a la Sección “Insertar Cuestionario de normalidad”.</li> <li>• Cuestionario de manualidad: Ir al Caso de Uso “Gestionar Cuestionario de manualidad” a la Sección “Insertar Cuestionario de Manualidad”.</li> <li>• Examen Físico-Neurológico: Ir al Caso de Uso “Gestionar Examen Físico-Neurológico” a la Sección “Insertar Examen Físico-Neurológico”.</li> </ul> <p>Si selecciona Mostrar Datos del Sujeto Ir a la Sección “Visualizar Datos del Sujeto”.</p>
<p>Flujo alterno de la Sección “Insertar Sujeto”</p>	
<p>2.1 Si el Usuario no desea insertar el sujeto sale de la sección.</p>	<p>4.1 Si el campo de código está sin valor o no está correcto se activa un mensaje de alerta notificando el error y se ejecuta a la acción 2 de la Sección “Insertar Sujeto”.</p>
<p>6.1 Si el Usuario no desea realizar ninguna de las operaciones sale de la sección.</p>	<p>5.1 Si el sujeto existe el sistema muestra un mensaje de alerta. Se ejecuta la acción 2 de la Sección “Insertar Sujeto”.</p>

Prototipo	Consultar Expediente de Proyecto
Sección "Buscar Sujeto"	
	1. El sistema muestra la interfaz correspondiente a la búsqueda del sujeto.
2. El Usuario selecciona el tipo de búsqueda e inserta el dato a buscar. <ul style="list-style-type: none"> <li>Código del sujeto.</li> </ul>	3. El sistema verifica que el parámetro seleccionado tenga valor y que esté correcto.
	4. El sistema realiza la búsqueda del sujeto que cumpla con el parámetro indicado.
	5. El sistema muestra el sujeto dando la posibilidad de ver los exámenes realizados, los exámenes no realizados y sus datos personales.
6. El Usuario marca una de las siguientes opciones <ul style="list-style-type: none"> <li>Exámenes Realizados</li> <li>Examen psicométrico</li> <li>Cuestionario de normalidad(Datos Personales)</li> <li>MINI Entrevista Neuropsiquiátrica Internacional</li> <li>Cuestionario de normalidad(Datos Clínicos)</li> <li>Cuestionario de manualidad</li> <li>Examen Físico-Neurológico</li> </ul> Exámenes no Realizados <ul style="list-style-type: none"> <li>Examen psicométrico</li> <li>Cuestionario de normalidad(Datos Personales)</li> <li>MINI Entrevista</li> </ul>	7. El sistema en correspondencia con la operación escogida por el Usuario realiza lo siguiente: <p>Si selecciona en los exámenes realizados.</p> <ul style="list-style-type: none"> <li>Examen psicométrico: Ir al Caso de Uso "Gestionar Examen psicométrico" a la Sección "Buscar Examen psicométrico"</li> <li>Cuestionario de normalidad(Datos Personales): Ir al Caso de Uso "Gestionar Cuestionario de normalidad" a la Sección "Buscar Cuestionario de normalidad"</li> <li>MINI Entrevista Neuropsiquiátrica Internacional: Ir al Caso de Uso "Gestionar MINI Entrevista Neuropsiquiátrica Internacional" a la Sección "Buscar MINI Entrevista Neuropsiquiátrica Internacional"</li> <li>Cuestionario de normalidad (Datos Clínicos): Ir al Caso de Uso "Gestionar Cuestionario de normalidad" a la Sección "Buscar Cuestionario de normalidad"</li> <li>Cuestionario de manualidad: Ir al Caso de Uso "Gestionar Cuestionario de manualidad" a la Sección</li> </ul>

<p>Neuropsiquiátrica Internacional</p> <ul style="list-style-type: none"> <li>• Cuestionario de normalidad(Datos Clínicos)</li> <li>• Cuestionario de manualidad</li> <li>• Examen Físico-Neurológico</li> </ul> <p>Mostrar Datos del Sujeto</p>	<p>“Buscar Cuestionario de Manualidad”.</p> <ul style="list-style-type: none"> <li>• Examen Físico-Neurológico: Ir al Caso de Uso “Gestionar Examen Físico-Neurológico” a la Sección “Buscar Examen Físico-Neurológico”.</li> </ul> <p>Si selecciona en los Exámenes sin realizar</p> <ul style="list-style-type: none"> <li>• Examen psicométrico: Ir al Caso de Uso “Gestionar Examen psicométrico”, a la Sección “Insertar Examen psicométrico”.</li> <li>• Cuestionario de normalidad (Datos Personales): Ir al Caso de Uso “Gestionar Cuestionario de normalidad“, a la Sección “Insertar Cuestionario de normalidad”.</li> <li>• MINI Entrevista Neuropsiquiátrica Internacional: Ir al Caso de Uso “Gestionar MINI Entrevista Neuropsiquiátrica Internacional” a la Sección “Insertar MINI Entrevista Neuropsiquiátrica Internacional”.</li> <li>• Cuestionario de normalidad (Datos Clínicos): Ir al Caso de Uso “Gestionar Cuestionario de normalidad“, a la Sección “Insertar Cuestionario de normalidad”.</li> <li>• Cuestionario de manualidad: Ir al Caso de Uso “Gestionar Cuestionario de manualidad” a la Sección “Insertar Cuestionario de Manualidad”.</li> <li>• Examen Físico-Neurológico: Ir al Caso de Uso “Gestionar Examen Físico-Neurológico” a la Sección “Insertar Examen Físico-Neurológico”.</li> </ul> <p>Si selecciona Mostrar Datos del Sujeto se ejecuta la Sección “Visualizar Datos del Sujeto”</p>
<p>Flujo Alterno 1 de la Sección “Buscar Sujeto”</p>	
<p>2.1 El Usuario selecciona el tipo de búsqueda e inserta el dato a buscar.</p> <ul style="list-style-type: none"> <li>• Nombre del sujeto.</li> </ul>	<p>2.2 El sistema verifica que el parámetro seleccionado tenga valor y que esté correcto.</p>

	2.3 El sistema realiza la búsqueda del sujeto que cumpla con el parámetro indicado.
	2.4 El sistema muestra el o los sujeto encontrados dando la posibilidad de seleccionar un sujeto para ver los exámenes realizados, los exámenes no realizados y sus datos personales.
2.5 El Usuario selecciona el sujeto e indica ver los exámenes realizados, los exámenes no realizados y sus datos personales.	2.5 Se ejecuta la acción 5 de la Sección “Buscar Sujeto”.
	4.1 Si el campo de código está sin valor o no está correcto se activa un mensaje de alerta notificando el error y se ejecuta a la acción 2 de la Sección “Buscar Sujeto”.
	5.1 Si el sujeto no existe el sistema muestra un mensaje de alerta. Se ejecuta la acción 2 de la Sección “Buscar Sujeto”.
Flujo Alterno 2 de la Sección “Buscar Sujeto”	
	2.3.1 Si el campo de código está sin valor o no está correcto se activa un mensaje de alerta notificando el error y se ejecuta a la acción 2 de la Sección “Buscar Sujeto”.
	2.4.1 Si el sujeto no existe el sistema muestra un mensaje de alerta. Se ejecuta la acción 2 de la Sección “Buscar Sujeto”.
Prototipo	Consultar Expediente de Proyecto
Sección “Visualizar Datos del Sujeto”	
	1. El sistema muestra los datos del sujeto dando la posibilidad de que estos puedan ser modificados.
2. El Usuario realiza los cambios necesarios y/o actualizaciones pertinentes.	3. El sistema valida que todos los campos fueron llenados correctamente
<ul style="list-style-type: none"> <li>• Código</li> <li>• Nombre</li> <li>• 1er Apellido</li> </ul>	

<ul style="list-style-type: none"> <li>• 2do Apellido</li> <li>• CI</li> </ul> <p>Datos de nacimiento</p> <ul style="list-style-type: none"> <li>• Fecha</li> <li>• Edad actual</li> <li>• Provincia</li> <li>• Municipio</li> </ul> <p>Dirección actual</p> <ul style="list-style-type: none"> <li>• Dirección</li> <li>• Provincia</li> <li>• Municipio</li> <li>• Policlínico</li> <li>• Teléfono</li> </ul> <p>Datos personales</p> <ul style="list-style-type: none"> <li>• Sexo</li> <li>• Manualidad Referida</li> <li>• Color de la piel</li> <li>• Peso</li> <li>• Talla</li> <li>• Temperatura</li> <li>• Nivel de escolaridad vencido</li> <li>• Ingreso per cápita</li> <li>• Fecha de última extracción de sangre</li> <li>• Estatus Social</li> <li>• Comentario</li> </ul>	
	4. El sistema registra los cambios realizados en los datos del sujeto.
Flujo Alternativo de la Sección "Visualizar Datos del Sujeto"	

2.1 El Usuario no desea modificar los datos del sujeto y sale de la sección.	4.1 Si los campos necesarios no tienen valor o existen campos con errores se muestra un mensaje de alerta. Se ejecuta la acción 2 de la Sección “Visualizar Datos del Sujeto”.
Prototipo	Consultar Expediente de Proyecto

#### **2.5.4.5 Describir los Actores**

<b>Actores</b>	<b>Justificación</b>
Usuario	Es el encargado de realizar las operaciones de inserción, búsqueda, actualización y eliminación de la información del sujeto y de los exámenes correspondientes al mismo.

## **CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA**

### **3.1 Introducción**

Aquí se evidencia la especificación que describe el comportamiento del sistema implementado basado en sus requisitos, a través del diseño. Se realizan los diagramas de clases del diseño y diagramas interacción más significativos. Se explica la arquitectura utilizada y los principales patrones de diseño utilizados.

### **3.2 Importancia del diseño de un sistema.**

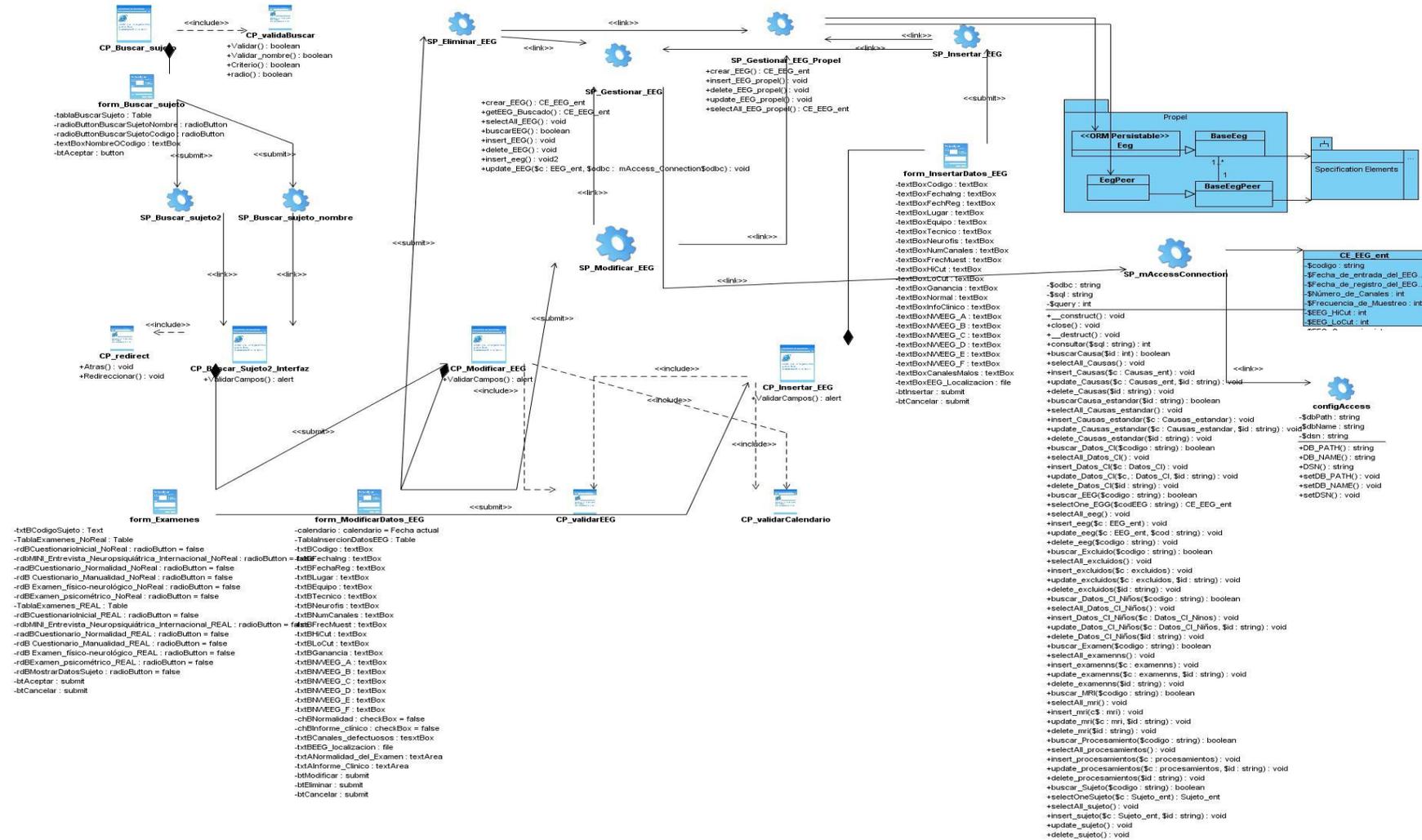
Para poder realizar un sistema que soporte todos los requerimientos especificados, tanto funcionales como no funcionales, se necesita modelarlo, lo cual se hace a través del diseño. El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar (23). Abarca los diagramas de clases y las realizaciones de los casos de uso del diseño a través de los diagramas de interacción. En el modelamiento del diseño se encontrará la forma que tendrá el sistema incluyendo la arquitectura, lo cual contribuirá a crear un punto de partida para las actividades del flujo de implementación.

### **3.3 Diagramas.**

#### **3.3.1 Diagramas de Clases.**

Un Diagrama de Clases es un diagrama que muestra un conjunto de interfaces, colaboraciones y sus relaciones. Gráficamente, un diagrama de clases es una colección de nodos y arcos. Un diagrama de clases del diseño representa las clases del diseño y sus objetos, así como los subsistemas del diseño.

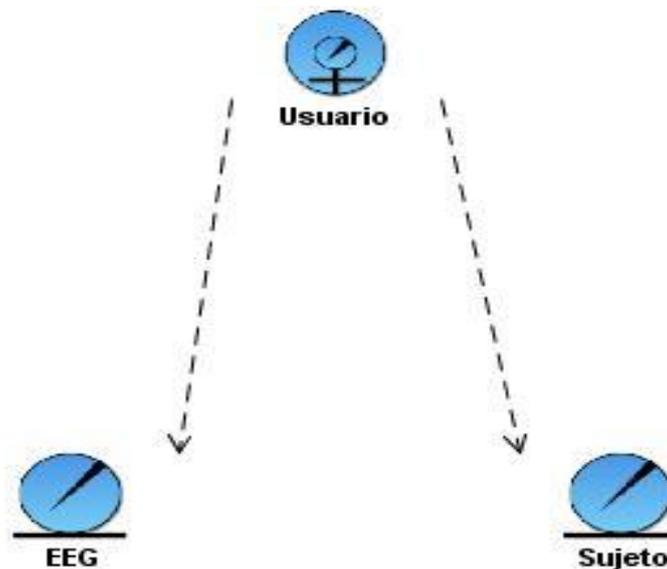
Fig.6 Diagrama de Clases del Diseño: CU Gestionar EEG.





### 3.3.2 Modelo de Objetos

Fig.8 Diagrama Objetos



### 3.3.3 Clases Persistentes

Se señalan como persistentes las clases EEG y Sujeto. Debido a que el sistema incluye un proceso de mejoras al modulo anterior no se diseña una Base de Datos sino que solo se diseña la tabla correspondiente a EEG, la clase Sujeto ya fue diseñada con anterioridad en el sistema previo.

Fig.9 Tabla EEG.

<b>&lt;&lt;ORM Persistable&gt;&gt; Eeg2</b>
-examen_sujeto_idsujeto : int
-examen_idexamen : int
-equipo_registro : String
-numero_canales : Integer
-canales_defectuosos : String
-frecuencia_muestreo : Integer
-eeg_hicut : Integer
-eeg_locut : Integer
-eeg_ganancia : Integer
-eeg_normal : String
-informe_clinico : String = "0"
-nweeg_a : Integer
-nweeg_b : Integer = 0
-nweeg_c : Integer = 0
-nweeg_d : Integer = 0
-nweeg_e : Integer = 0
-nweeg_f : Integer = 0
-eeg_path : String
-fecha_insercion : Date

### **3.3.4 Diagramas de Interacción.**

Se trata de un término genérico que se aplica a varios tipos de diagramas que hacen hincapié en las interacciones entre objetos. Los diagramas de actividades están íntimamente relacionados.

El patrón de interacción entre objetos se muestra en un diagrama de interacción. Los diagramas de interacción tienen diferentes formas, basadas todas ellas en una misma información subyacente pero resaltando cada una un punto de vista de la misma: diagramas de secuencia, diagramas de colaboración.

Un diagrama de secuencia muestra una interacción que está organizada como una secuencia temporal. En particular, muestra los objetos que participan en la interacción mediante sus líneas de vida y mediante los mensajes que intercambian, organizados en forma de una secuencia temporal. Un diagrama de secuencia no muestra los enlaces existentes entre objetos. Los diagramas de secuencia tienen distintos formatos, adecuados para propósitos diferentes.

Un diagrama de colaboración muestra una interacción organizada en torno a los objetos que efectúan operaciones. Es parecido a un diagrama de objetos que muestra los objetos y los enlaces existentes entre ellos que se necesitan para implementar una operación de nivel más elevado.

#### **3.3.4.1 Diagramas de Colaboración.**

Una colaboración modela los objetos y los enlaces significativos dentro de una interacción. Los objetos y los enlaces son significativos solamente en el contexto proporcionado por la interacción. Un rol describe un objeto, y un rol en la asociación describe un enlace dentro de una colaboración.

Un diagrama de colaboración muestra los roles en la interacción en una disposición geométrica. Los mensajes se muestran como flechas, ligadas a las líneas de la relación, que conectan a los roles. La secuencia de mensajes, se indica con los números secuenciales que preceden a las descripciones del mensaje.

Un uso de un diagrama de colaboración es mostrar la implementación de una operación. La colaboración muestra los parámetros y las variables locales de la operación, así como asociaciones más permanentes. Cuando se implementa el comportamiento, la secuencia de los mensajes corresponde a la estructura de llamadas anidadas y el paso de señales del programa.

### **3.3.4.2 Diagramas de Secuencia.**

Un diagrama de secuencia representa una interacción como un gráfico bidimensional. La dimensión vertical es el eje de tiempo, que avanza hacia abajo de la página. La dimensión horizontal muestra los roles de clasificador que representan objetos individuales en la colaboración. Cada rol de clasificador se representa mediante una columna vertical-línea de vida. 'Durante el tiempo que existe un objeto, el rol se muestra por una línea discontinua. Durante el tiempo que dura una activación de un procedimiento en el objeto, la línea de vida se dibuja como una línea doble.

Se muestra un mensaje como una flecha desde la línea de vida de un objeto a la del otro. Las flechas se organizan en el diagrama en orden cronológico hacia abajo.

**Fig.10 - Diagrama de Secuencia: CU Insertar EEG para MS Access**

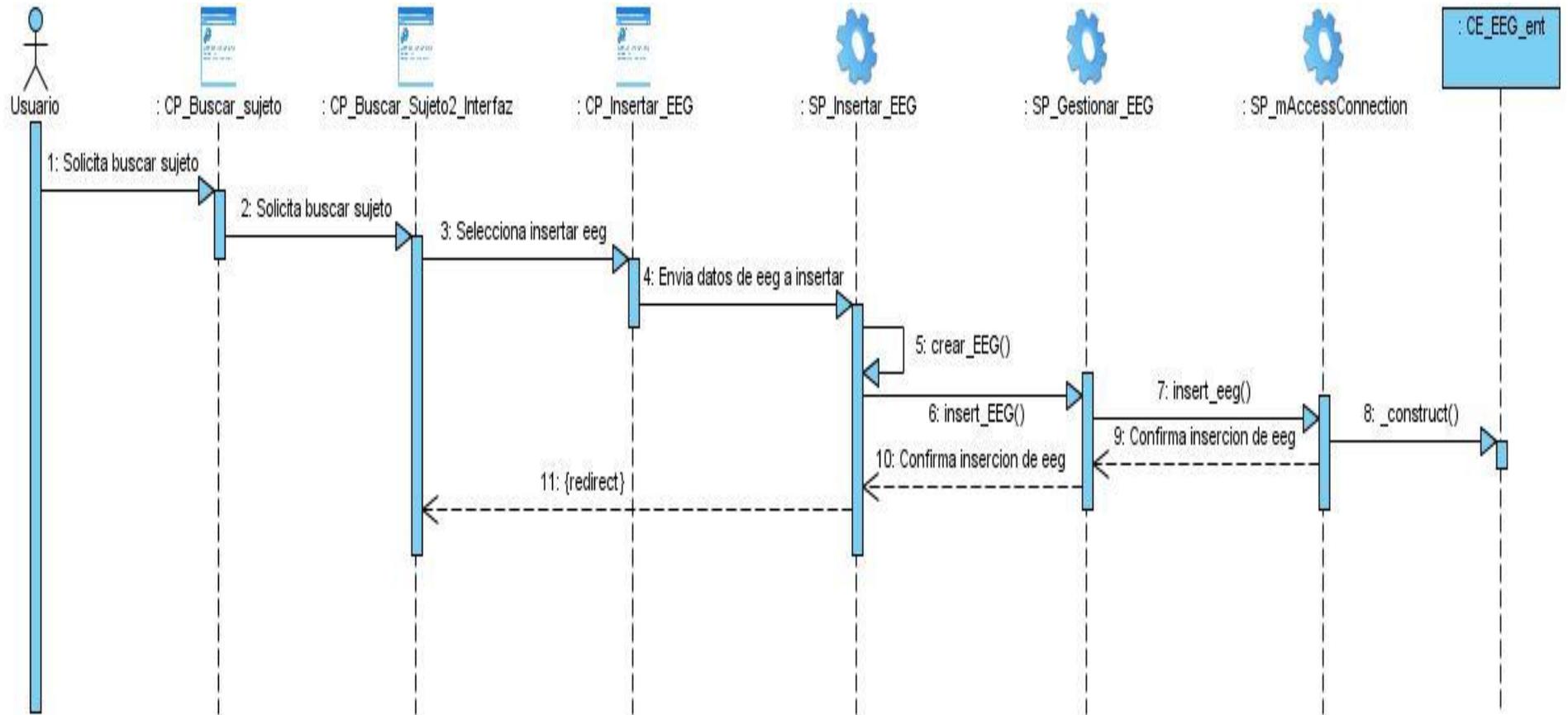


Fig.11 - Diagrama de Secuencia: CU Insertar EEG para Propel

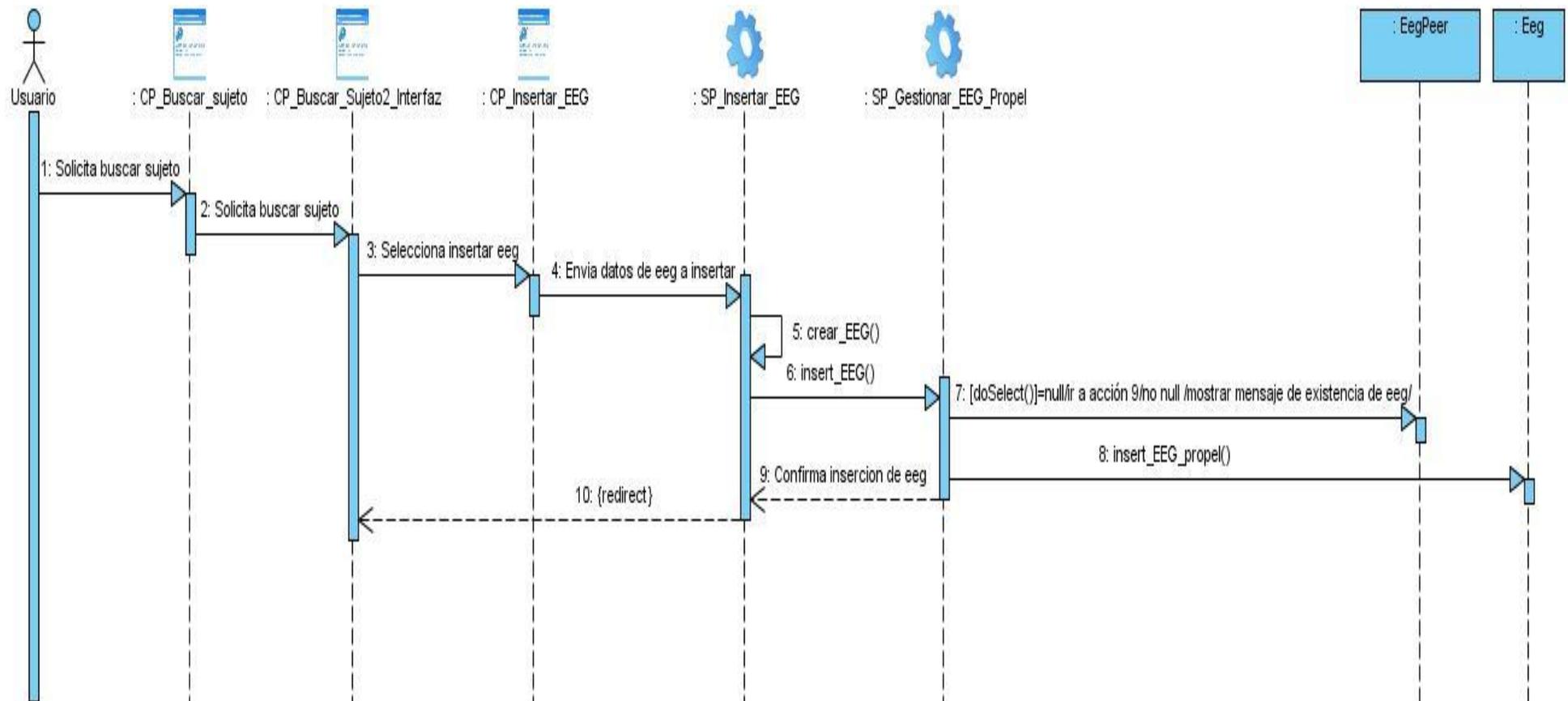


Fig. 12 - Diagrama de Secuencia: CU Modificar EEG para MS Access

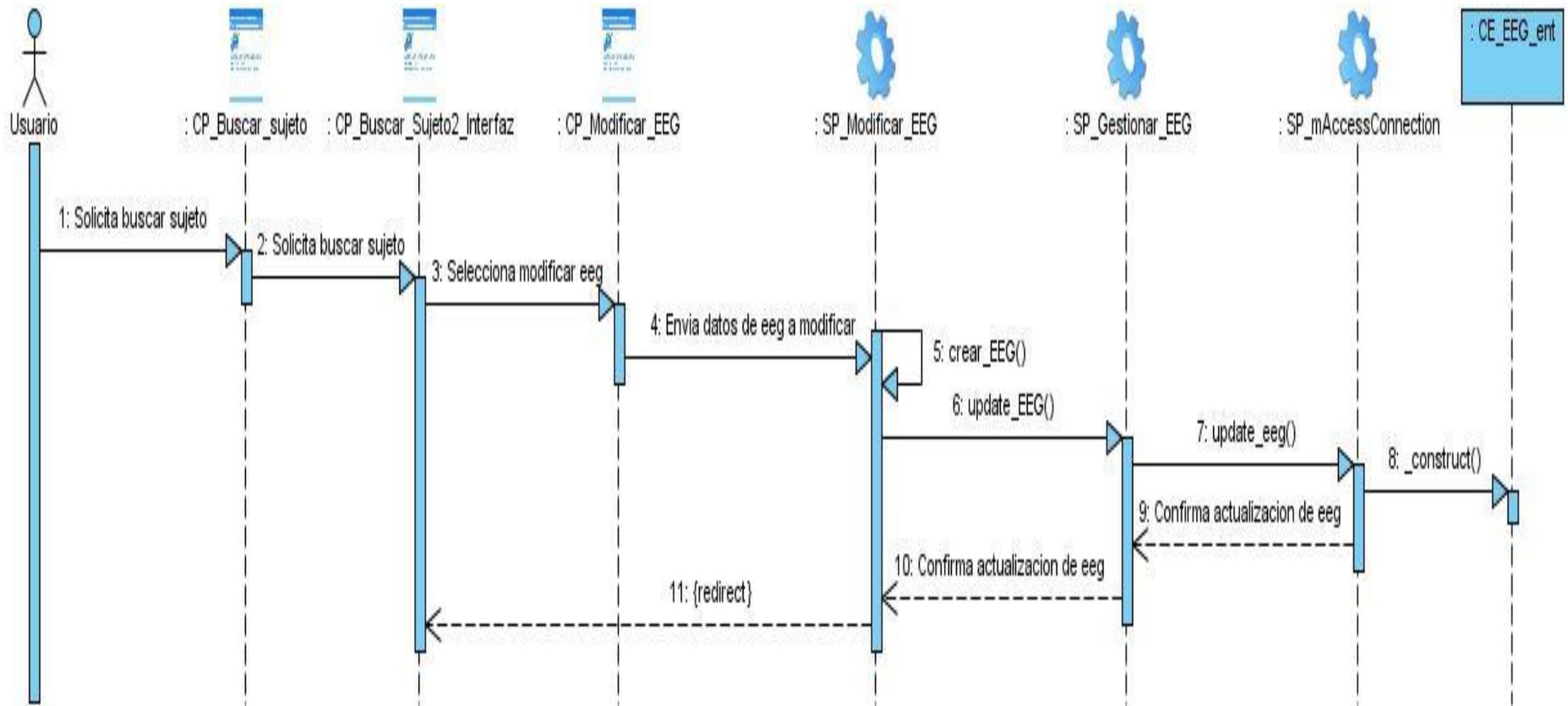


Fig.13 - Diagrama de Secuencia: CU Modificar EEG para Propel

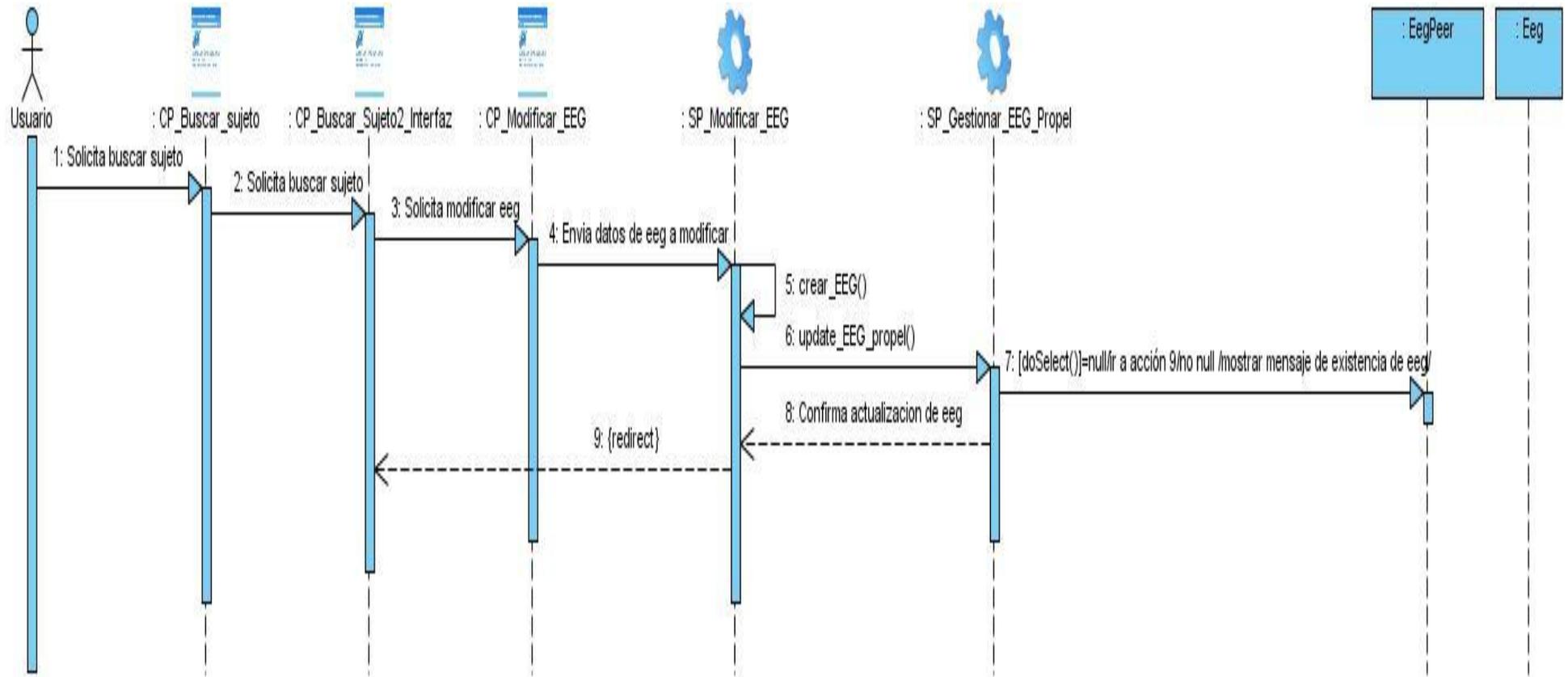


Fig. 14 - Diagrama de Secuencia: CU Eliminar EEG para MS Access.

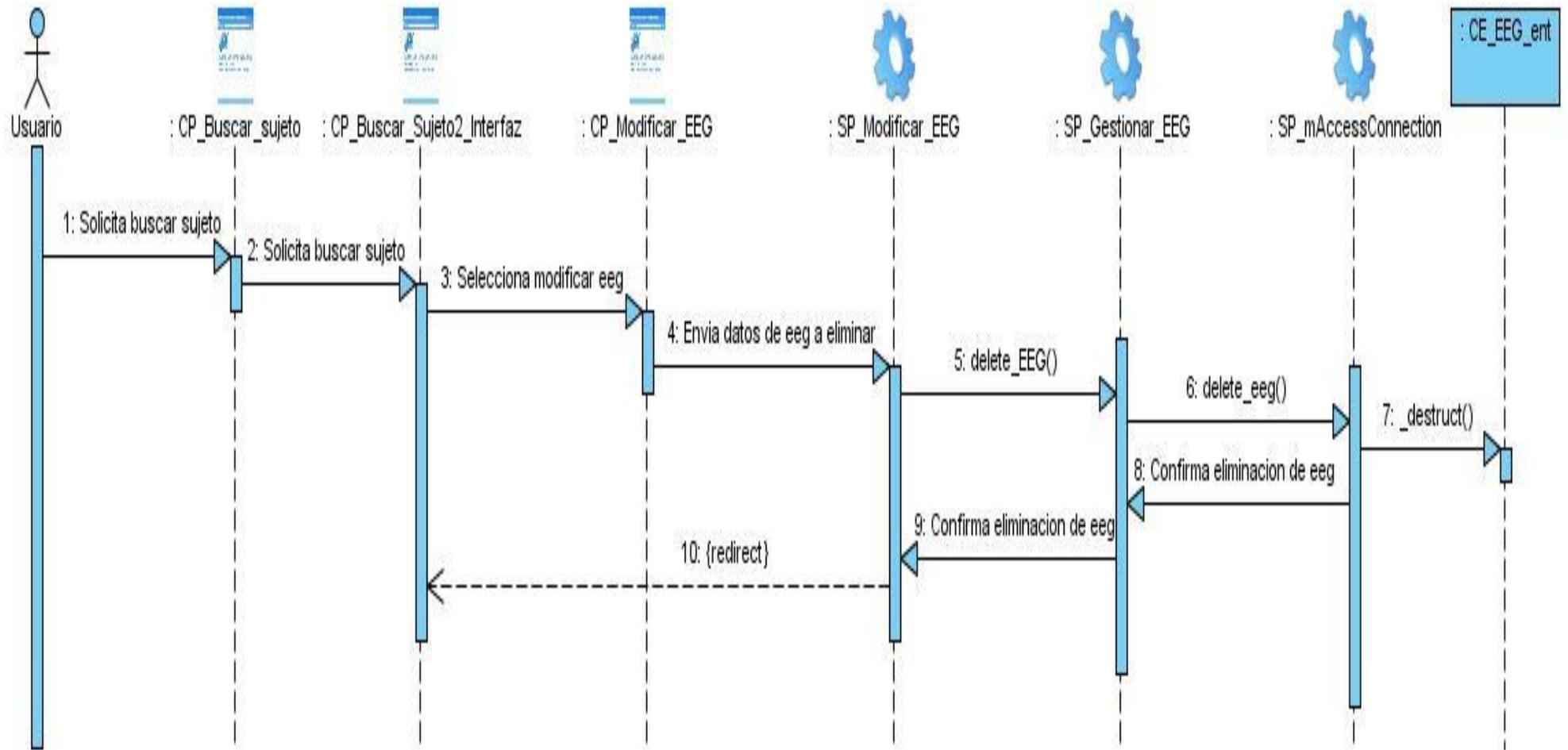
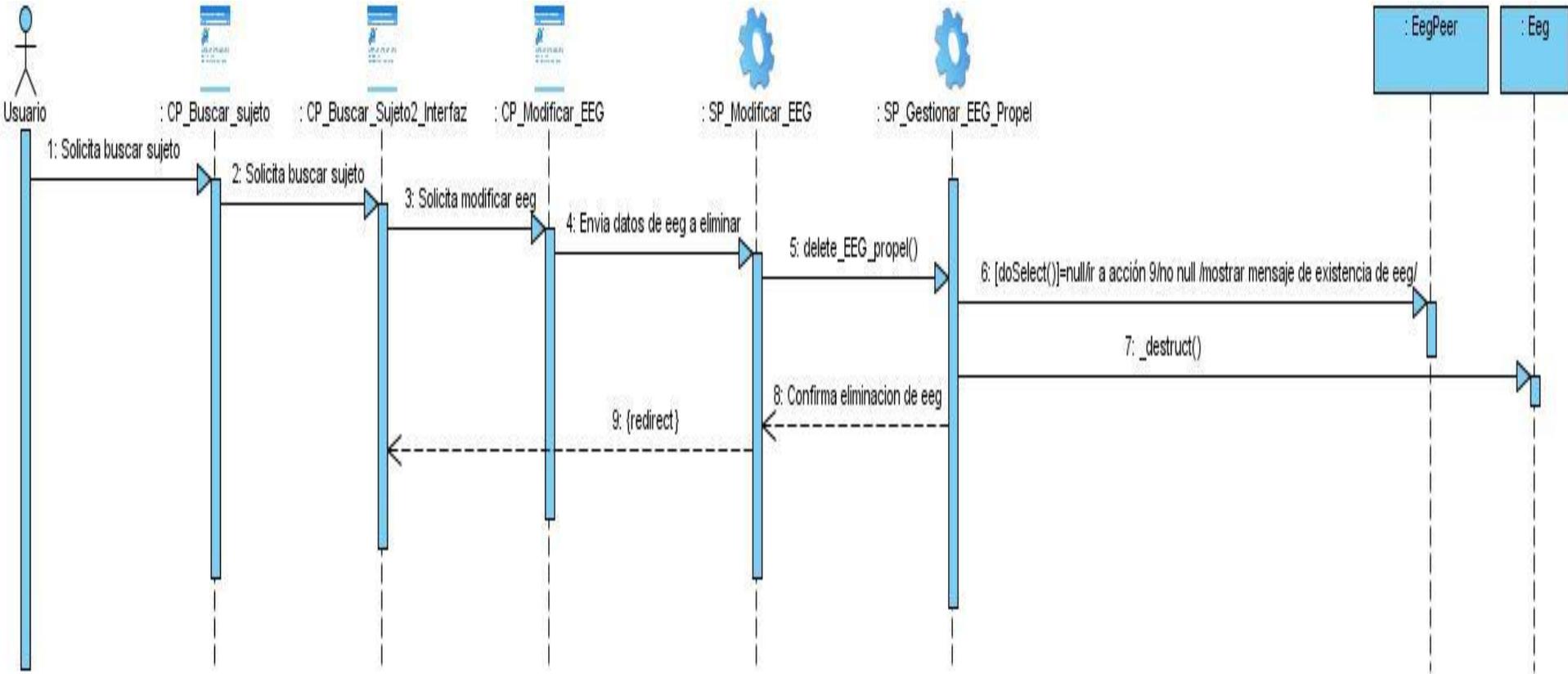


Fig. 15 - Diagrama de Secuencia: CU Eliminar EEG para Propel



**Fig. 16 - Diagrama de Secuencia: CU Insertar Sujeto para MS Access**

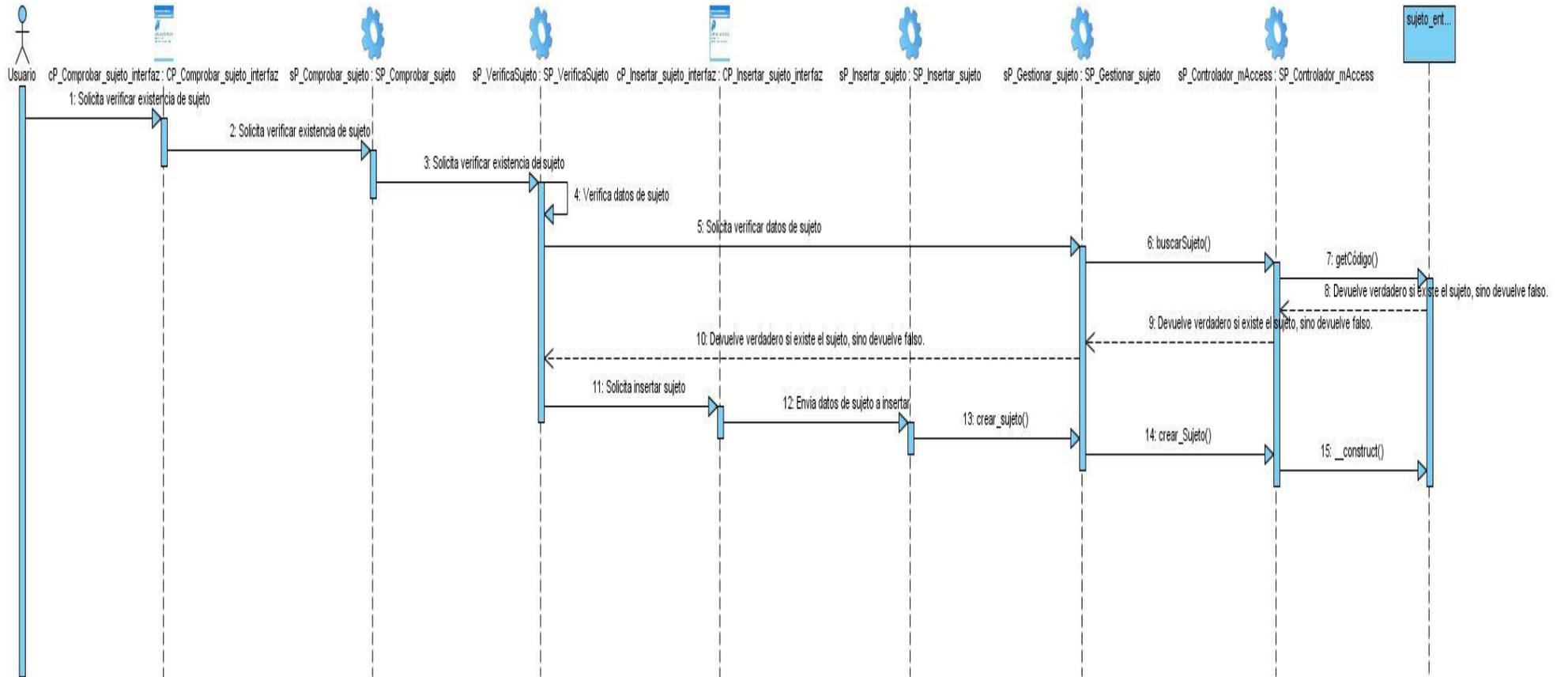
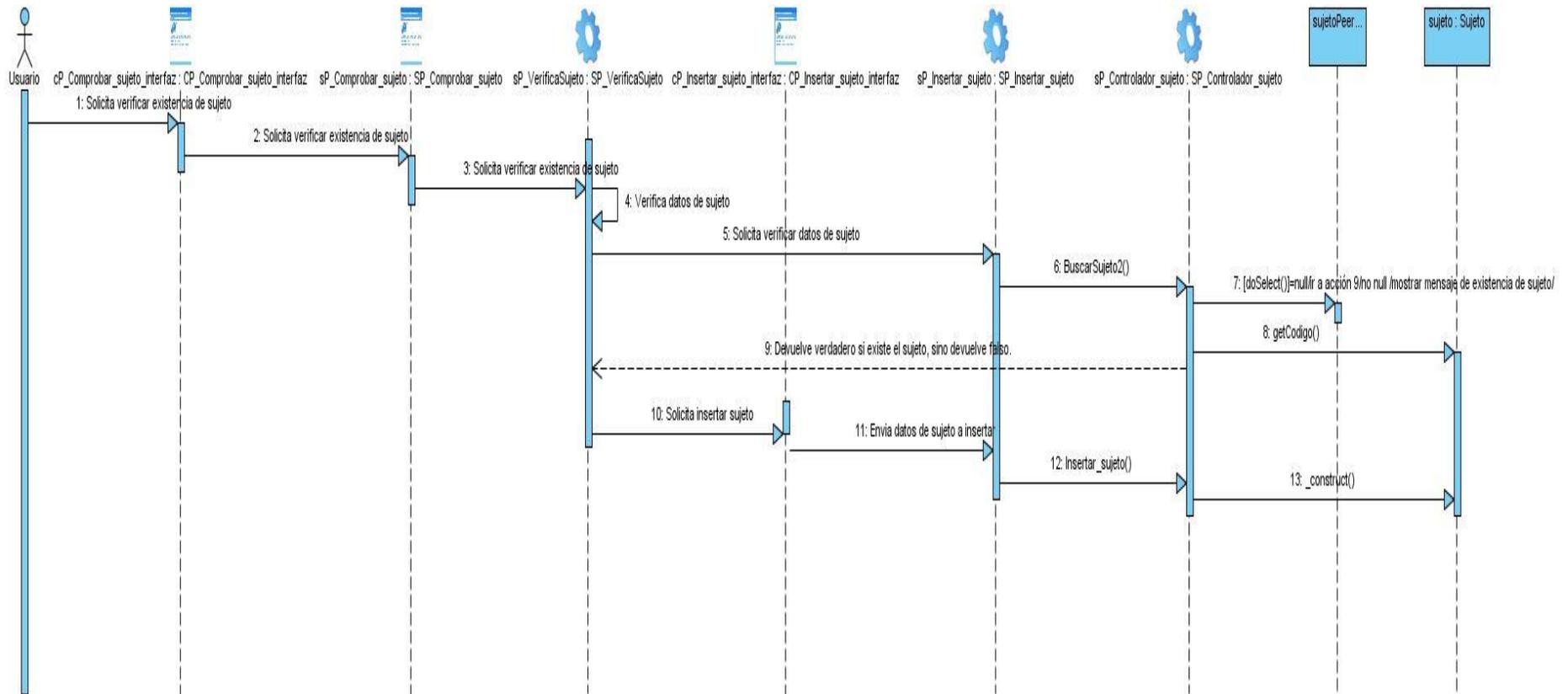


Fig. 17 - Diagrama de Secuencia: CU Insertar Sujeto para Propel



## **3.4 Arquitectura y Patrones.**

### **3.4.1 Arquitectura**

La Arquitectura proporciona una visión global del sistema a construir, permitiendo al ingeniero examinarlo como un todo. Describe la estructura y organización de los componentes de software, sus propiedades y la manera en que se comunican entre ellos. La arquitectura destaca decisiones de diseño tempranas que tendrán un profundo impacto que todo el trabajo de ingeniería de software posterior, y proporciona el mecanismo para evaluar los beneficios de las estructuras de sistema alternativas.

### **3.4.2 Diseño Orientado a Objetos**

El diseño de software orientado a objetos requiere la definición de una Arquitectura de Software multicapa, la especificación de subsistemas que realizan funciones necesarias y proveen soporte a la infraestructura, una descripción de objetos (clases), que son los bloques de construcción del sistema, y una descripción de los mecanismos de comunicación, que permiten que los datos fluyan entre las capas, subsistemas y objetos.

### **3.4.3 Concepto de Patrón**

"Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma".[10]

### **3.4.4 Patrón de Arquitectura**

Los patrones de arquitectura expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos. Ayudan a especificar la estructura fundamental de una aplicación. Cada actividad de desarrollo es gobernada por esta estructura; por

ejemplo, el diseño detallado de los subsistemas, la comunicación y colaboración entre diferentes partes del sistema, etc. Cada patrón de arquitectura ayuda a conseguir una propiedad específica en el sistema global; por ejemplo, la adaptabilidad de la interfaz de usuario. Los patrones que dan soporte a características similares se agrupan en una misma categoría. Un patrón de arquitectura puede contener varios patrones de diseño. (26)

Los patrones de arquitectura describen un problema particular y recurrente de diseño, que aparece en contextos de diseño específico, y presenta un esquema genérico demostrado con éxito para su solución. El objetivo principal que persiguen los patrones es crear un lenguaje común entre los Diseñadores y Desarrolladores para comunicar experiencias sobre los problemas y sus soluciones.

Durante el diseño del sistema se utilizó el patrón Modelo-Vista-Controlador o Separación Modelo-Vista como también se le conoce. Este patrón de arquitectura separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

**Modelo:** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.

**Vista:** Muestra la información al usuario. Obtiene los datos del modelo. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.

**Controlador:** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista. El usuario interactúa con el sistema a través de los controladores.

Las Vistas y los Controladores conforman la interfaz de usuario. Un mecanismo de propagación de cambios asegura la consistencia entre la interfaz y el modelo. La separación del modelo de los componentes vista y del controlador permite tener múltiples vistas del mismo modelo. Si el usuario cambia el modelo a través del controlador de una vista, todas las otras vistas dependientes deben reflejar los cambios. Por lo tanto, el modelo notifica a todas las vistas siempre que sus datos cambien. Las vistas, en cambio, recuperan los nuevos datos del modelo y actualizan la información que muestran al usuario.

Algunos de sus principales beneficios son:

- Menor acoplamiento
  - Desacopla las vistas de los modelos.

- Desacopla los modelos de la forma en que se muestran e ingresan los datos.
- Mayor cohesión
  - Cada elemento del patrón está altamente especializado en su tarea (la vista en mostrar datos al usuario, el controlador en las entradas y el modelo en su objetivo de negocio).
- Las vistas proveen mayor flexibilidad y agilidad.
  - Se puede crear múltiples vistas de un modelo.
  - Se puede crear, añadir, modificar y eliminar nuevas vistas dinámicamente.
  - Las vistas pueden anidarse.
  - Se puede cambiar el modo en que una vista responde al usuario sin cambiar su representación visual.
  - Se puede sincronizar las vistas.
  - Las vistas pueden concentrarse en diferentes aspectos del modelo.
- Mayor facilidad para el desarrollo de clientes ricos en múltiples dispositivos y canales.
  - Una vista para cada dispositivo que puede variar según sus capacidades.
  - Una vista para la Web y otra para aplicaciones de escritorio.
- Más claridad de diseño.
- Facilita el mantenimiento.
- Mayor escalabilidad.

### **3.4.5 Patrón de Diseño**

“Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software.”

Un patrón de diseño describe una estructura recurrente de componentes que se comunican para resolver un problema general de diseño en un contexto particular. Nomina, abstrae e identifica los aspectos clave de una estructura de diseño común, lo que los hace útiles para crear un diseño

orientado a objetos reusable. Identifica las clases e instancias participantes, sus roles y colaboraciones y la distribución de responsabilidades.

En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Los patrones de diseño pretenden:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.
- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

Se clasifican en:

- **Patrones Creacionales:** Inicialización y configuración de objetos.
- **Patrones Estructurales:** Separan la interfaz de la implementación. Se ocupan de cómo las clases y objetos se agrupan, para formar estructuras más grandes.
- **Patrones de Comportamiento:** Más que describir objetos o clases, describen la comunicación entre ellos.

### 3.4.6 Patrones de diseño en el MVC

Un patrón de arquitectura puede contener varios patrones de diseño. A modo de ejemplo, se cita el patrón de arquitectura Modelo-Vista-Controlador (analizado en el apartado anterior) que contiene (o puede contener) los siguientes patrones de diseño:

#### Estructurales:

- **Composite View:** para la creación de vistas compuestas. Este patrón permite crear una jerarquía de vistas y tratar a cada vista compuesta igual que una a una vista normal.

Fig. 18 – Interfaz de la Aplicación

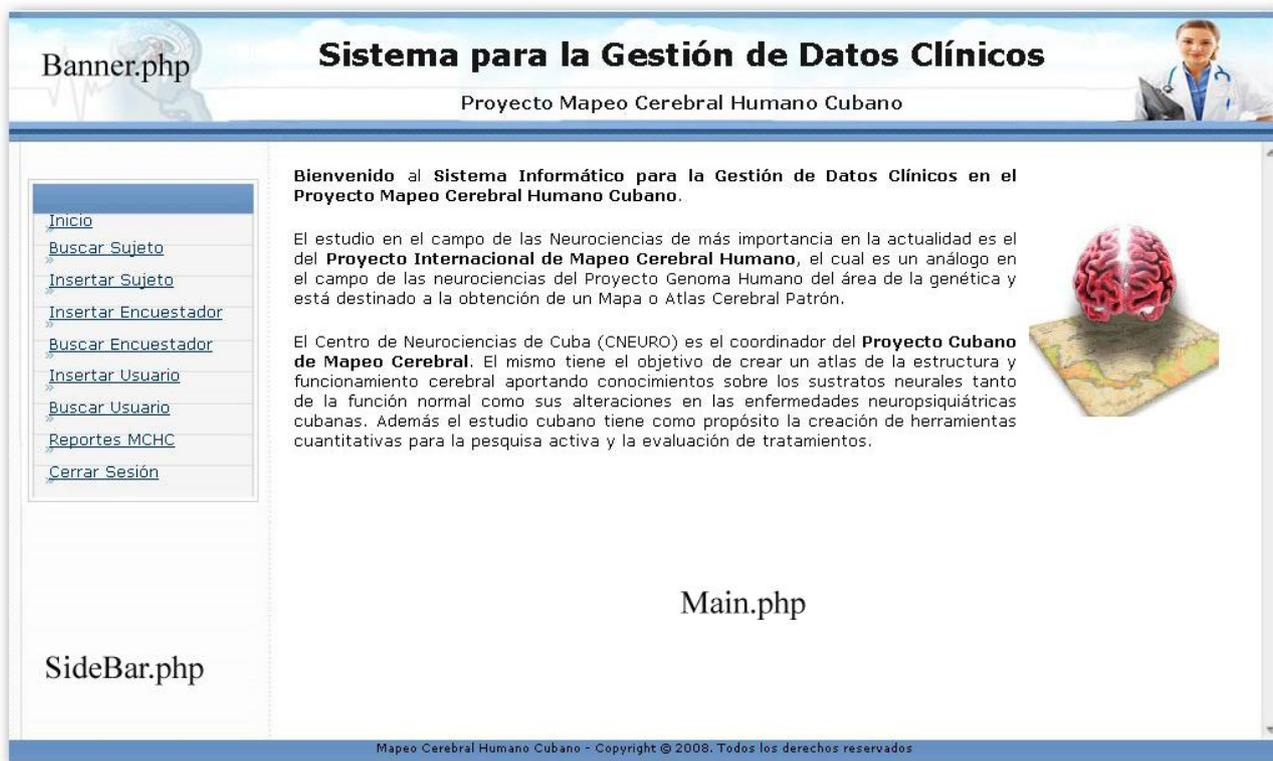
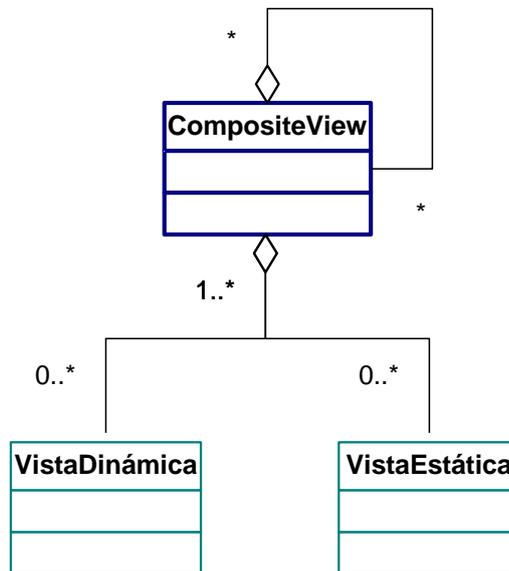


Fig. 19 – Arquitectura del patrón Composite View.



Con uso de este patrón de diseño se mejora la modularidad y la reutilización, mejora la flexibilidad, el mantenimiento, y la manejabilidad.

- **Decorator:** Para añadir capacidades adicionales a una vista (por ejemplo, scroll).
- **Facade + Adapter:** Una única interfaz de acceso para múltiples base de datos.

Fig. 20 – Clase de Acceso a Datos a ambas Bases de Datos.

```

43 /***** Access *****/
44 $ctrl_EEG_access = new Gestionar_EEG ( );
45
46 $eeg = $ctrl_EEG_access->crear_EEG( $fechIngres, $fechReg, $numCanales, $frecMuest
47 , $loCut, $ganancia, $nweeg_a, $nweeg_b, $nweeg_c, $nweeg_d, $nweeg_e, $nweeg_f, $
48 $lugar, $equipo, $tecnico, $neurofis, $normalidad, $canalDefect, $eegLocal, $info
49
50 /***** Access *****/
51
52
53 $ctrl_EEG_propel = new Gestionar_EEG_Propel ( );
54 $ctrl_EEG_propel->update_EEG_propel($eeg, $neurofis, $tecnico, $lugar);
    
```

- **Flyweight:** Gestión de vistas y uso de clases (CSS).

Fig. 21 – Uso de CSS.

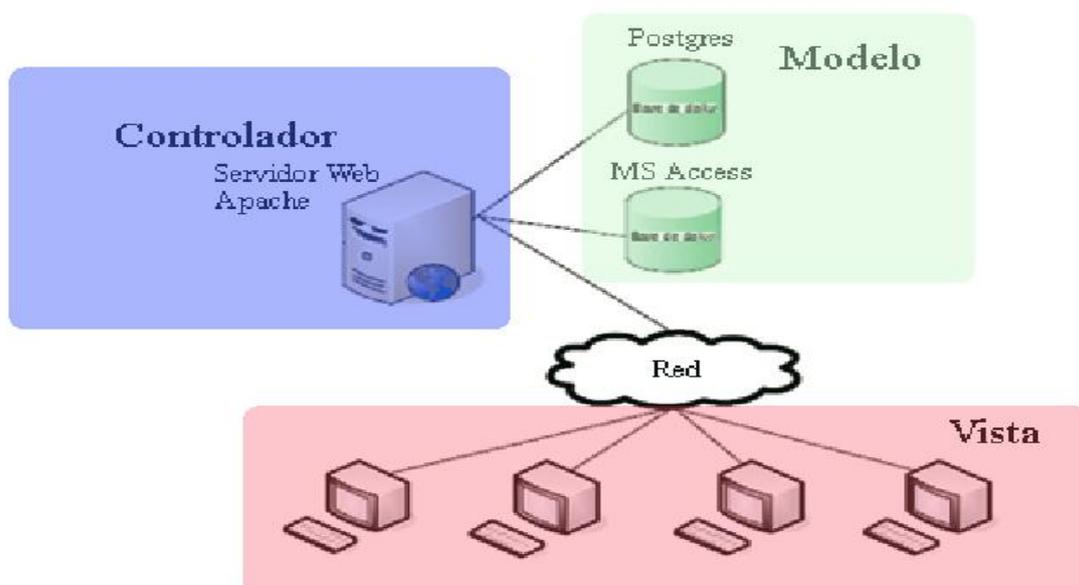
```

54 <style type="text/css">
55 .ex {
56     font-weight: bold;
57     background: #fed;
58     color: #080
59 }
60
61 .help {
62     color: #080;
63     font-style: italic;
64 }
65
66 body {
67     background: #F7F7F7;
68     font: 10pt tahoma, verdana, sans-serif;
69 }
70
71 table {
72     font: 13px verdana, tahoma, sans-serif;
73 }
74
75 a {
76     color: #00f;
77 }
78
79 a:visited {
80     color: #00f;

```

- **Proxy:** Para distribuir la arquitectura (Modelo y Vista-Controlador) en diferentes emplazamientos.

Fig. 22 – Distribución de la aplicación de acuerdo al patrón Proxy.





## **Experto**

Consiste en asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

Beneficios:

Con la aplicación de este patrón se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento.

El comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clase "sencillas" y más cohesivas que son fáciles de comprender y de mantener. Así se brinda soporte a una alta cohesión.

## **Creador**

“Consiste en asignarle a la clase B la responsabilidad de crear una instancia de clase A en uno de los siguientes casos:

B agrega objetos de A

B contiene objetos de A

B registra las instancias de los objetos A

B utiliza específicamente los objetos A

B tiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado (así que B es experto respecto a la creación de A).”

El diseño, bien asignado, soporta bajo acoplamiento, mayor claridad, encapsulamiento y reutilización.

Beneficios:

Se brinda soporte a un bajo acoplamiento, lo cual supone menos dependencia respecto al mantenimiento y mejores oportunidades de reutilización.

## **Bajo acoplamiento**

Consiste en asignar responsabilidad para mantener bajo acoplamiento. Es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

**Beneficios:**

No se afectan por cambios de otros componentes fáciles de entender por separado y fáciles de reutilizar.

### **Alta cohesión**

Consiste en asignar una responsabilidad de modo que la alta cohesión siga siendo alta. Grady y Booch señala que se da una alta cohesión funcional cuando los elementos de un componente (clase, por ejemplo) "colaboran para producir algún comportamiento bien definido".

**Beneficios:**

Mejoran la claridad y la facilidad con que se entiende el diseño. Se simplifica el mantenimiento y las mejoras en funcionalidad. A menudo se genera un bajo acoplamiento. La ventaja de una gran funcionalidad soporta una mayor capacidad de reutilización, porque una clase muy cohesiva puede destinarse a un propósito muy específico.

### **Controlador**

Consiste en asignar las responsabilidades del manejo de un mensaje de los eventos de un sistema a una clase que represente una de las siguientes opciones:

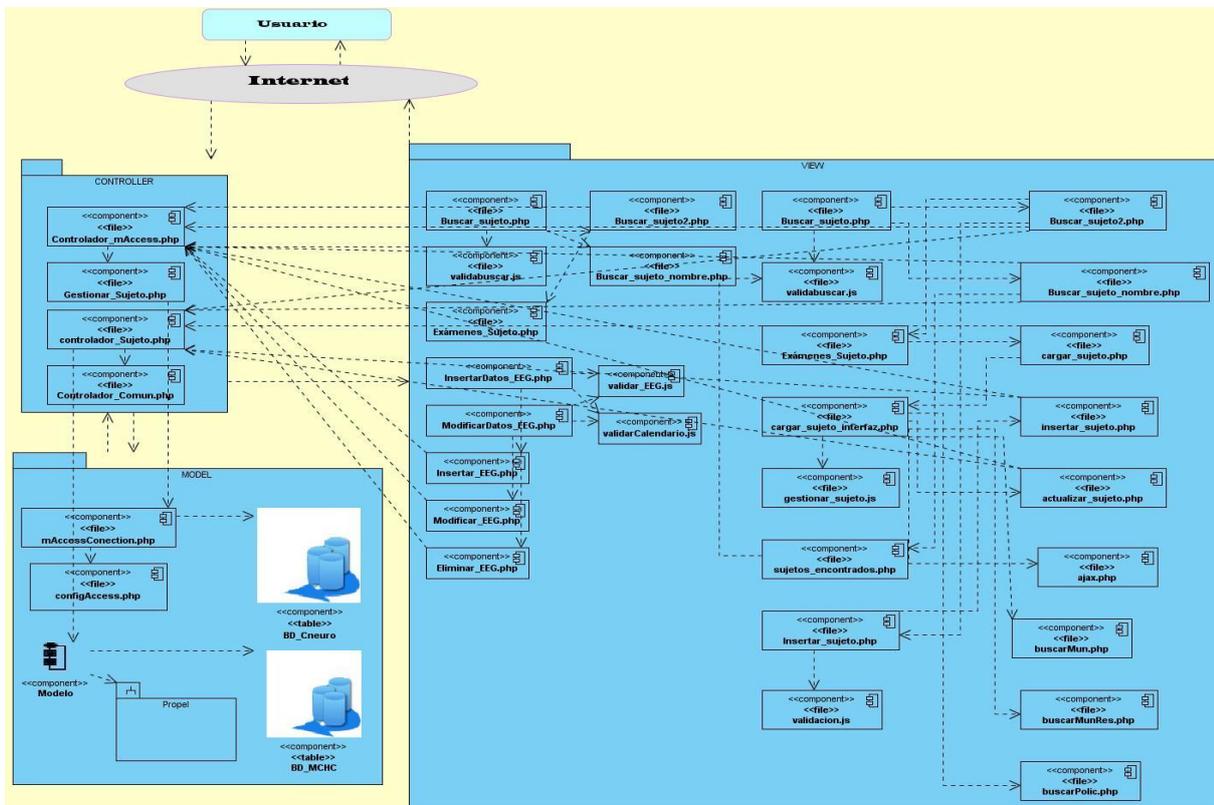
- El "sistema" global (controlador de fachada).
- La empresa u organización global (controlador de fachada).
- Algo en el mundo real que es activo (por ejemplo, el papel de una persona) y que pueda participar en la tarea (controlador de tareas).
- Un manejador artificial de todos los eventos del sistema de un caso de uso, generalmente denominados "manejador<NombreCasodeUso>" (controlador de casos de uso).

**Beneficios:**

Mayor potencial de los componentes reutilizables. Garantiza que la empresa o los procesos de dominio sean manejados por la capa de los objetos del dominio y no por la interfaz.

### 3.4.8 Arquitectura Actual del Sistema

Fig. 24 – Componentes ubicados según el Patrón MVC.

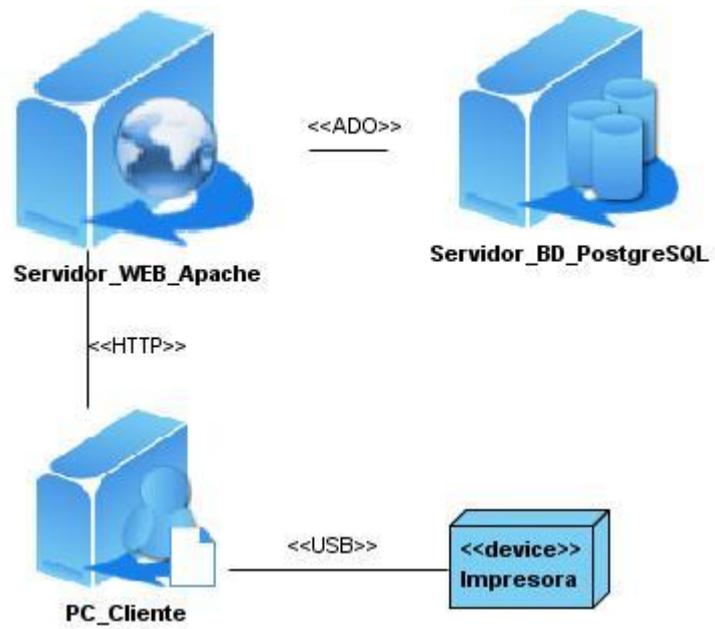


### 3.5 Modelo de Despliegue

Es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo.

La aplicación estará distribuida en los servidores de CNEURO, de la siguiente manera: Un servidor web para publicar el sistema, otro servidor para la base de datos, como gestor de base datos se tendrá PostgreSQL y Apache 2 para el servidor de las páginas web, a la aplicación se tendrá acceso desde cualquier máquina cliente del centro. En la figura se presenta el diagrama de despliegue.

Fig. 25 – Diagrama de Despliegue.



## **CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA**

### **4.1 Introducción**

En el capítulo se referirán detalles de la fase de implementación de la aplicación para dar solución a los requisitos especificados, desarrollándose el artefacto Modelo de componentes, en congruencia con la solución de la fase del diseño.

### **4.2 Modelo de Implementación**

El modelo de implementación describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, etc.

#### ***4.2.1 Diagrama de Componentes***

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño. El diagrama de componentes describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros.

Fig. 26 – Diagrama de Componentes CU: Gestionar Sujeto.

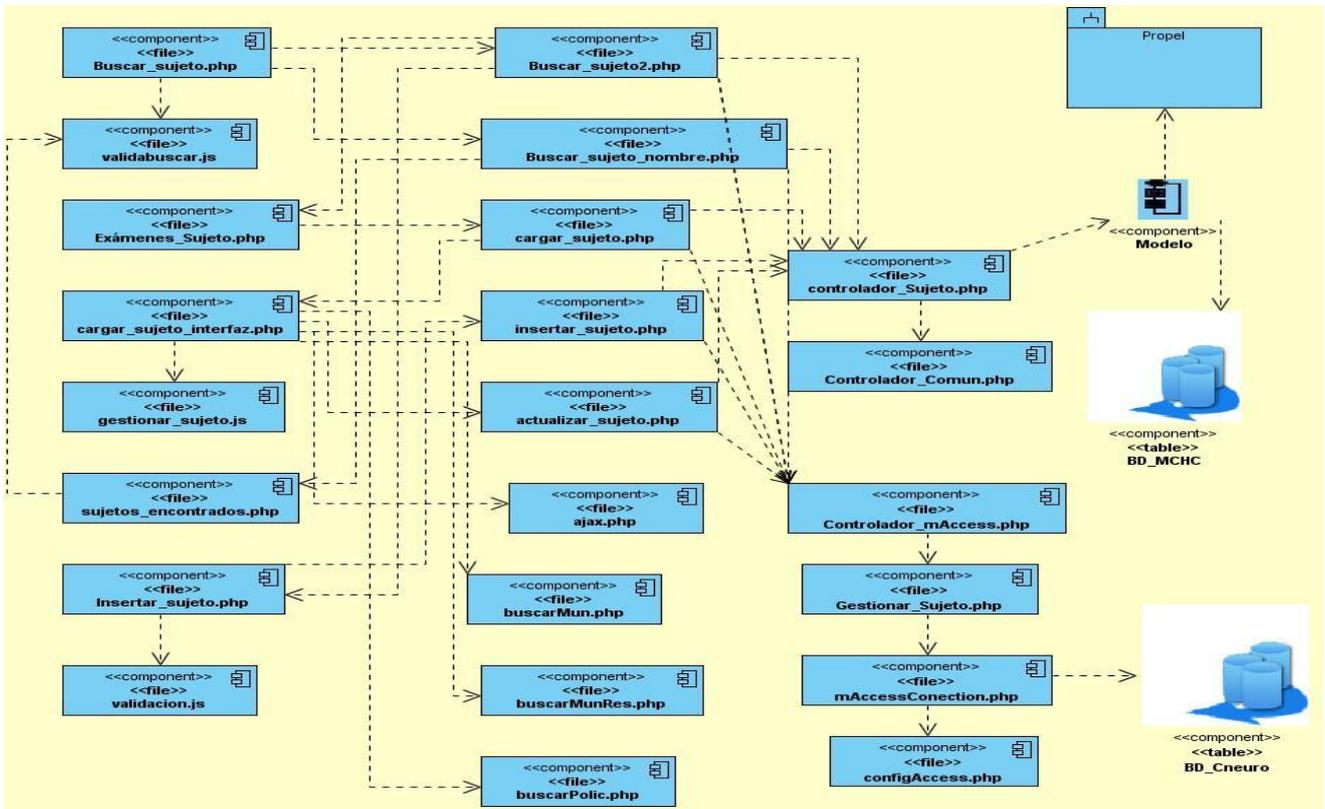
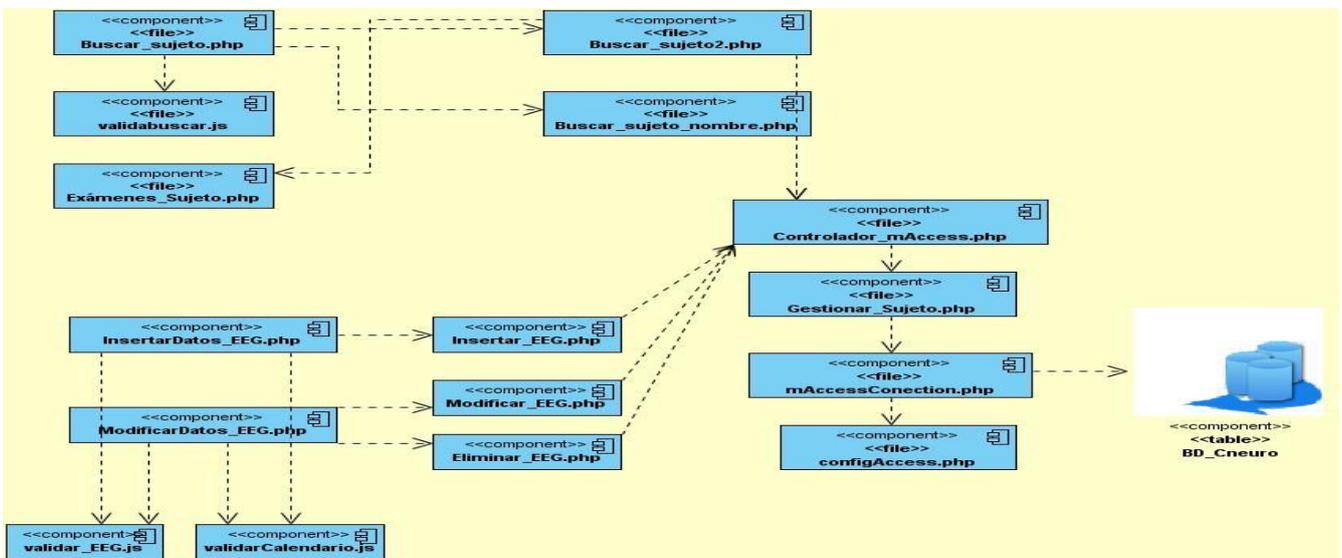


Fig.27 Diagrama de Componentes: CU Gestionar EEG



## 4.3 Consideraciones finales

### 4.3.1 Prototipo de Interfaz

Fig.28 Página Principal de la Aplicación

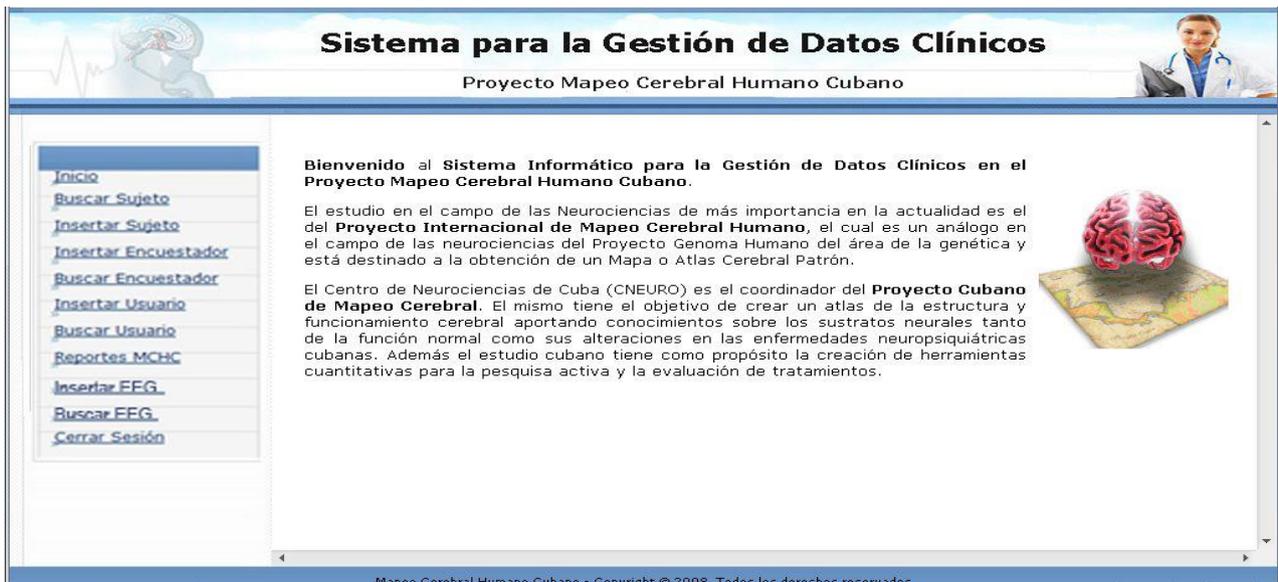


Fig.29 Página Buscar Sujeto

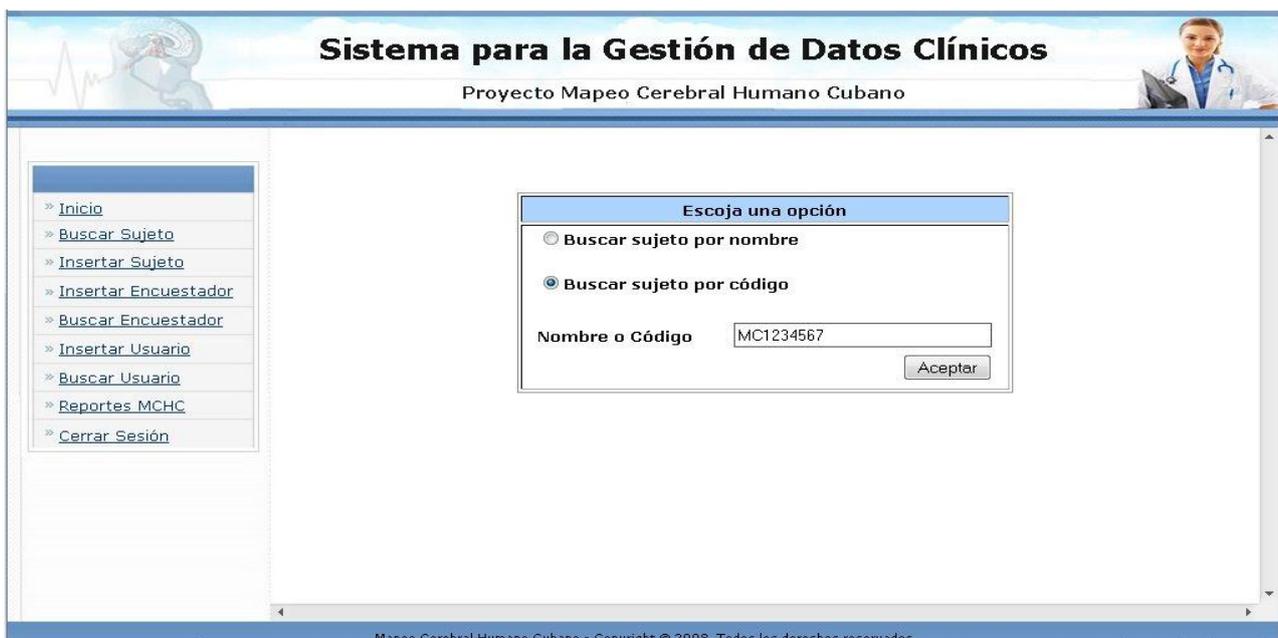


Fig.30 Página donde se muestran los exámenes realizados y no realizados.

**Sistema para la Gestión de Datos Clínicos**  
Proyecto Mapeo Cerebral Humano Cubano

El sujeto actualmente está en el estudio

Código del sujeto

Exámenes realizados	Exámenes sin realizar
	<input type="radio"/> Cuestionario Inicial
	<input type="radio"/> MINI Entrevista Neuropsiquiátrica Internacional
	<input type="radio"/> Cuestionario de normalidad
	<input type="radio"/> Cuestionario de manualidad
	<input type="radio"/> Examen físico-neurológico
	<input type="radio"/> Examen psicométrico
	<input checked="" type="radio"/> Electroencefalograma

Mostrar datos del sujeto

Mapeo Cerebral Humano Cubano - Copyright © 2008. Todos los derechos reservados.

Fig.31 Página donde se insertan los datos del electroencefalograma.

**Sistema para la Gestión de Datos Clínicos**  
Proyecto Mapeo Cerebral Humano Cubano

Código	<input type="text"/>	(cód. EEG)
Fecha ingreso	<input type="text"/>	"dd/mm/aaaa"
Fecha registro	<input type="text"/>	"dd/mm/aaaa"
Lugar	<input type="text"/>	(nombre)
Equipo	<input type="text"/>	(nombre)
Técnico	<input type="text"/>	(nombre)
Neurofisiólogo	<input type="text"/>	(nombre)
Número canales	<input type="text"/>	(#)
Frecuenc. muestreo	<input type="text"/>	(#)
HICut	<input type="text"/>	(#)
LoCut	<input type="text"/>	(#)
Ganancia	<input type="text"/>	(#)
NWEEG_A	<input type="text"/>	(#)
NWEEG_B	<input type="text"/>	(#)
NWEEG_C	<input type="text"/>	(#)
NWEEG_D	<input type="text"/>	(#)
NWEEG_E	<input type="text"/>	(#)
NWEEG_F	<input type="text"/>	(#)
Normalidad	<input type="checkbox"/> Comentarios	
Informe clínico	<input type="checkbox"/> Comentarios	
Canales defectuosos	<input type="text"/>	(#)
EEG localización	<input type="text"/>	<input type="button" value="Examinar"/>

Normalidad del Examen

Informe clínico

Mapeo Cerebral Humano Cubano - Copyright © 2008. Todos los derechos reservados.

Fig.32 Página que permite modificar o eliminar los datos referentes al electroencefalograma.

Código	MC0000001	(cód. EEG)
Fecha ingreso	27/12/2005	"dd/mm/aaaa"
Fecha registro	27/12/2005	"dd/mm/aaaa"
Lugar	San Agustín	(nombre)
Equipo	Much4_1	(nombre)
Técnico	Celso Valdivia	(nombre)
Neurofisiólogo	Trinidad Vínes	(nombre)
Número canales	56	(#)
Frecuenc. muestreo	200	(#)
HiCut	50	(#)
LoCut	0,5	(#)
Ganancia	20000	(#)
NWEEG_A	80	(#)
NWEEG_B	30	(#)
NWEEG_C	4	(#)
NWEEG_D	1	(#)
NWEEG_E	4	(#)
NWEEG_F	10	(#)
Normalidad	<input checked="" type="checkbox"/> Comentarios	
Informe clínico	<input checked="" type="checkbox"/> Comentarios	
Canales defectuosos	12, 91	(#)
EEG localización	eeg / MC0000001	Examinar...

#### 4.3.2 Pasos a Seguir para su utilización

1. Introducir el Código del Sujeto que pertenece al estudio que se desea gestionar.
2. Seleccionar Electroencefalograma.
3. Si el electroencefalograma no se ha insertado con anterioridad, se insertan los datos.
4. Si el electroencefalograma ya ha sido insertado se da la posibilidad de eliminarlo y de modificar los datos del mismo por resultados mas actualizados.
5. En cualquiera de los casos anteriores luego de realizar alguna acción se actualiza el estado del examen.

## CONCLUSIONES

1. Se analizó el proceso de gestión de la información del Examen Clínico logrando una mejor comprensión del mismo.
2. Se identificaron las funcionalidades a cumplir por el sistema para la gestión de la información referente a los electroencefalogramas dentro de los Examen Clínicos.
3. Se diseñaron las clases complementarias al sistema de gestión de la información del Examen Clínico que permitieron implementar la tabla que almacenará los datos generados por los electroencefalogramas y además definir como se implementará el sistema.
4. Con la utilización del procedimiento propuesto será posible entender el negocio del cliente y obtener todos los requisitos tanto implícitos como explícitos del sistema.
5. Se implementaron las funcionalidades que permiten la gestión de electroencefalogramas y garantizan la migración de los datos almacenados en la medida que se gestionan los exámenes de los sujetos en estudio, asegurando la estabilidad y eficiencia de la base de datos, lo que aumenta considerablemente las capacidades de procesamiento y gestión de la información contenida.
6. El sistema constituye una mejora importante al Sistema de Gestión de la Información de Exámenes Clínicos anteriormente realizado.

## **RECOMENDACIONES**

- ✓ Se recomienda que se continúe actualizando el sistema con las nuevas funcionalidades que van surgiendo por parte del CNEURO.
- ✓ Se propone como plataforma de desarrollo de una nueva versión del sistema en Java para que esté integrado a la plataforma de desarrollo del proyecto.

## REFERENCIAS BIBLIOGRÁFICAS

1. **Llanez, Clara Carmen.** Mapeo Cerebral Humano: Un reto del siglo XXI. *Mapeo Cerebral Humano: Un reto del siglo XXI*. [En línea] 19 de Septiembre de 2005. [Citado el: 15 de Marzo de 2009.] <http://www.voltairenet.org/article128202.html> p. 1
2. **Llanez, Clara Carmen.** Mapeo Cerebral Humano: Un reto del siglo XXI. *Mapeo Cerebral Humano: Un reto del siglo XXI*. [En línea] 19 de Septiembre de 2005. [Citado el: 15 de Marzo de 2009.] <http://www.voltairenet.org/article128202.html> p. 1
3. ICBM International Consortium for Brain Mapping. *ICBM International Consortium for Brain Mapping*. [En línea] 2005. [Citado el: 15 de Marzo de 2009.] <http://www.loni.ucla.edu/ICBM/> p.1
4. **Avendaño, Bárbara.** Trazos sobre una carta inconclusa. *Trazos sobre una carta inconclusa*. [En línea] 24 de enero de 2008. [Citado el: 15 de abril de 2009.] <http://www.bohemia.cu/2008/01/23/cienciatecnologia/1-mapa.html> p. 4
5. **Centro Nacional de Tecnologías de Información(CNTI), Institución adscrita al Ministerio del Poder Popular para las Telecomunicaciones y la Informática. República Bolivariana de Venezuela.** Beneficios del Software Libre. *Beneficios del Software Libre*. [En línea] 2008. [Citado el: 20 de 3 de 2009.] [http://www.cnti.gob.ve/index.php?option=com\\_content&task=view&id=94](http://www.cnti.gob.ve/index.php?option=com_content&task=view&id=94) p.1
6. **González, Maricarmen.** Cuba avanza en la migración al software libre. *Cuba avanza en la migración al software libre*. [En línea] 19 de marzo de 2009. [Citado el: 21 de marzo de 2009.] [http://www.cnti.gob.ve/index.php?option=com\\_content&view=article&id=1782:cuba-avanza-en-la-migracion-al-software-libre-&catid=50:software-libre&Itemid=112](http://www.cnti.gob.ve/index.php?option=com_content&view=article&id=1782:cuba-avanza-en-la-migracion-al-software-libre-&catid=50:software-libre&Itemid=112) p.1
7. **Gómez Castaño, Julio César: Ingeniero de Sistemas, Universidad de Manizales.** BENEFICIOS DEL SOFTWARE LIBRE. *BENEFICIOS DEL SOFTWARE LIBRE*. [En línea] 5 de octubre de 2002. [Citado el: 20 de marzo de 2009.] <http://www.umanizales.edu.co/programs/ingenieria/ventana/ventanaespecial/04.pdf> p. 1

8. **Barea Navarro, Rafael . Instrumentación Biomédica. Departamento Electrónica. Universidad Alcalá.** Electroencefalografía. *Electroencefalografía*. [En línea] [Citado el: 1 de abril de 2009.]

<http://usuarios.lycos.es/legajius/Dir/Protocolos/Eeg/electroencefalografia.pdf> p.1

9. **Barea Navarro, Rafael . Instrumentación Biomédica. Departamento Electrónica. Universidad Alcalá.** Electroencefalografía. *Electroencefalografía*. [En línea] [Citado el: 1 de abril de 2009.]

<http://usuarios.lycos.es/legajius/Dir/Protocolos/Eeg/electroencefalografia.pdf> p.1

9. **Barea Navarro, Rafael . Instrumentación Biomédica. Departamento Electrónica. Universidad Alcalá.** Electroencefalografía. *Electroencefalografía*. [En línea] [Citado el: 1 de abril de 2009.]

<http://usuarios.lycos.es/legajius/Dir/Protocolos/Eeg/electroencefalografia.pdf> p.9

10. **Welicki, León.** Patrones y Antipatrones: una Introducción - Parte I . *Patrones y Antipatrones: una Introducción - Parte I*. [En línea] 2009. [Citado el: 20 de mayo de 2009.] [http://msdn.microsoft.com/es-](http://msdn.microsoft.com/es-es/library/bb972242.aspx)

[es/library/bb972242.aspx](http://msdn.microsoft.com/es-es/library/bb972242.aspx) p.1

11. —. Patrones y Antipatrones: una Introducción - Parte II. *Patrones y Antipatrones: una Introducción - Parte II*. [En línea] 2009. [Citado el: 20 de mayo de 2009.] [http://msdn.microsoft.com/es-](http://msdn.microsoft.com/es-es/library/bb972251.aspx#mainSection)

[es/library/bb972251.aspx#mainSection](http://msdn.microsoft.com/es-es/library/bb972251.aspx#mainSection) p. 1

## BIBLIOGRAFÍA

**Aragonesa, Gran Enciclopedia. 2000.** Migración, tipos de Migración. *Migración, tipos de Migración*. [En línea] 2000. [Citado el: 22 de marzo de 2009.] [http://www.encyclopedia-aragonesa.com/voz.asp?voz\\_id=8796#](http://www.encyclopedia-aragonesa.com/voz.asp?voz_id=8796#)

**Avendaño, Bárbara. 2008.** Trazos sobre una carta inconclusa. *Trazos sobre una carta inconclusa*. [En línea] 24 de enero de 2008. [Citado el: 15 de abril de 2009.] <http://www.bohemia.cu/2008/01/23/cienciatecnologia/1-mapa.html>

**Barea Navarro, Rafael . Instrumentación Biomédica. Departamento Electrónica. Universidad Alcalá.** Electroencefalografía. *Electroencefalografía*. [En línea] [Citado el: 1 de abril de 2009.] <http://usuarios.lycos.es/legajius/Dir/Protocolos/Eeg/electroencefalografia.pdf>

**Centro Nacional de Tecnologías de Información(CNTI), Institución adscrita al Ministerio del Poder Popular para las Telecomunicaciones y la Informática. República Bolivariana de Venezuela. 2008.** Beneficios del Software Libre. *Beneficios del Software Libre*. [En línea] 2008. [Citado el: 20 de 3 de 2009.] [http://www.cnti.gob.ve/index.php?option=com\\_content&task=view&id=94](http://www.cnti.gob.ve/index.php?option=com_content&task=view&id=94)

**Definicion.de. 2008.** Concepto de migración. *Concepto de migración*. [En línea] 2008. [Citado el: 19 de marzo de 2009.] <http://definicion.de/migracion/>

**eaprende.com. 2008.** Gestor de Base de Datos: MySQL, PostgreSQL, SQLite. *Gestor de Base de Datos: MySQL, PostgreSQL, SQLite*. [En línea] 2008. [Citado el: 17 de 3 de 2009.] <http://www.eaprende.com/gestor-de-basededatos-mysql-postgresql-sqlite.html>

**Gómez Castaño, Julio César: Ingeniero de Sistemas, Universidad de Manizales. 2002.** BENEFICIOS DEL SOFTWARE LIBRE. *BENEFICIOS DEL SOFTWARE LIBRE*. [En línea] 5 de octubre de 2002. [Citado el: 20 de marzo de 2009.] <http://www.umanizales.edu.co/programs/ingenieria/ventana/ventanaespecial/04.pdf>

**González, Maricarmen. 2009.** Cuba avanza en la migración al software libre. *Cuba avanza en la migración al software libre*. [En línea] 19 de marzo de 2009. [Citado el: 21 de marzo de 2009.]

[http://www.cnti.gob.ve/index.php?option=com\\_content&view=article&id=1782:cuba-avanza-en-la-migracion-al-software-libre-&catid=50:software-libre&Itemid=112](http://www.cnti.gob.ve/index.php?option=com_content&view=article&id=1782:cuba-avanza-en-la-migracion-al-software-libre-&catid=50:software-libre&Itemid=112)

**Health System, University of Virginia. 2007.** La Radiología. Tomografía por Emisión de Positrones. *La Radiología. Tomografía por Emisión de Positrones*. [En línea] 2007. [Citado el: 1 de junio de 2009.] [http://www.healthsystem.virginia.edu/uvahealth/adult\\_radiology\\_sp/pet.cfm](http://www.healthsystem.virginia.edu/uvahealth/adult_radiology_sp/pet.cfm)

**Larousse, Diccionario de Uso del Español de América Vox ©.** migración. *migración*. [En línea] [Citado el: 22 de marzo de 2009.] <http://www.diccionarios.com/consultas.php?palabra=migracion&diccionario=definicion>

**Larousse, Diccionario Sinónimos y Antónimos Vox. 2007.** Migración. [En línea] 2007. [Citado el: 22 de marzo de 2009.] <http://www.diccionarios.com/consultas.php>

**Leopoldo, Clínica Fray. 2005.** NEUROFISIOLOGÍA GRANADA. *NEUROFISIOLOGÍA GRANADA*. [En línea] 2005. [Citado el: 2 de abril de 2009.] <http://www.neurofisiologiagranada.com/eeg/eeg-quees.htm>

**Llenez, Clara Carmen. 2005.** Mapeo Cerebral Humano: Un reto del siglo XXI. *Mapeo Cerebral Humano: Un reto del siglo XXI*. [En línea] 19 de Septiembre de 2005. [Citado el: 15 de Marzo de 2009.] <http://www.voltairenet.org/article128202.html>

**López, Pablo. 2009.** Mitos y realidades del Software libre. *Mitos y realidades del Software libre*. [En línea] 28 de abril de 2009. [Citado el: 29 de abril de 2009.] [http://www.cnti.gob.ve/index.php?option=com\\_content&view=article&id=2015:mitos-y-realidades-del-software-libre&catid=50:software-libre&Itemid=112](http://www.cnti.gob.ve/index.php?option=com_content&view=article&id=2015:mitos-y-realidades-del-software-libre&catid=50:software-libre&Itemid=112)

**Lorca Quezada, Carlos Felipe: Ingeniero de ejecución en Informática Universidad Católica de Temuco. 2007.** Cuando queda chico un motor base de datos... ¿Qué hacer? *Cuando queda chico un motor base de datos... ¿Qué hacer?* [En línea] 7 de marzo de 2007. [Citado el: 21 de marzo de 2009.] <http://www.mmug.cl/articulos.php?id=293>

**Microsoft. 2007.** Las diez mejores ventajas de Microsoft Office Access 2007. *Las diez mejores ventajas de Microsoft Office Access 2007*. [En línea] 2007. [Citado el: 23 de marzo de 2009.] <http://office.microsoft.com/es-es/access/HA101650213082.aspx>

**Microsoft, Ayuda y Soporte. 2004.** Cómo identificar los componentes de motor de base de datos Jet. *Cómo identificar los componentes de motor de base de datos Jet*. [En línea] 5 de agosto de 2004. [Citado el: 23 de marzo de 2009.] <http://support.microsoft.com/kb/141796/es>

**Microsoft, Ayuda y Soporte. 2008.** Como mantener una base de datos Jet 4.0 en buenas condiciones. *Como mantener una base de datos Jet 4.0 en buenas condiciones*. [En línea] 4 de marzo de 2008. [Citado el: 23 de marzo de 2009.] <http://support.microsoft.com/kb/303528/es>

**Migración. Migración.** [En línea]

<http://www.diccionarios.com/consultas.php?palabra=migracion&diccionario=definicion>

Migración. *Migración*. [En línea] [Citado el: 19 de marzo de 2009.]

<http://www.mastermagazine.info/termino/5893.php>

**Real Academia Española, RAE.** Migración. *Migración*. [En línea] [Citado el: 22 de marzo de 2009.]

<http://buscon.rae.es/drael/>

**Ruiz, Angela.** Vulnerabilidad en Microsoft Jet Database (MSJET40.DLL). *Vulnerabilidad en Microsoft Jet Database (MSJET40.DLL)*. [En línea] [Citado el: 23 de marzo de 2009.]

<http://www.vsantivirus.com/vul-bugtraq-12960.htm>

**Sun Microsystems, Inc. 2008-2009.** MySQL 5.0 Reference Manual. *MySQL 5.0 Reference Manual*. [En línea] Sun Microsystems, Inc, 2008-2009. [Citado el: 21 de marzo de 2009.]

<http://dev.mysql.com/doc/refman/5.0/es/index.html>

**Tedeschi, Nicolás. 2009.** ¿Qué es un Patrón de Diseño? . *¿Qué es un Patrón de Diseño?* . [En línea] 2009. [Citado el: 20 de mayo de 2009.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>

**Tomografía Axial Computarizada (TAC).** *Tomografía Axial Computarizada (TAC)*. [En línea] [Citado el: 1 de junio de 2009.] <http://www.ideagdl.com.mx/index.php/enlaces/61-tomografia-axial-computarizada>

**UCI, Departamento de Ingeniería de Software de la. 2009.** Conferencia 7: Fase de Elaboración. Flujo de trabajo de Análisis y Diseño. *Conferencia 7: Fase de Elaboración. Flujo de trabajo de Análisis y Diseño.* [En línea] 2009. [Citado el: 26 de marzo de 2009.]

<http://teleformacion.uci.cu/mod/resource/view.php?id=13151>

**Welicki, León. 2009.** Patrones y Antipatrones: una Introducción - Parte I . *Patrones y Antipatrones: una Introducción - Parte I .* [En línea] 2009. [Citado el: 20 de mayo de 2009.]

<http://msdn.microsoft.com/es-es/library/bb972242.aspx>

**2005.** ICBM International Consortium for Brain Mapping. *ICBM International Consortium for Brain Mapping.* [En línea] 2005. [Citado el: 15 de Marzo de 2009.] <http://www.loni.ucla.edu/ICBM/>

**2007.** MS Access. *MS Access.* [En línea] 12 de septiembre de 2007. [Citado el: 23 de marzo de 2009.]

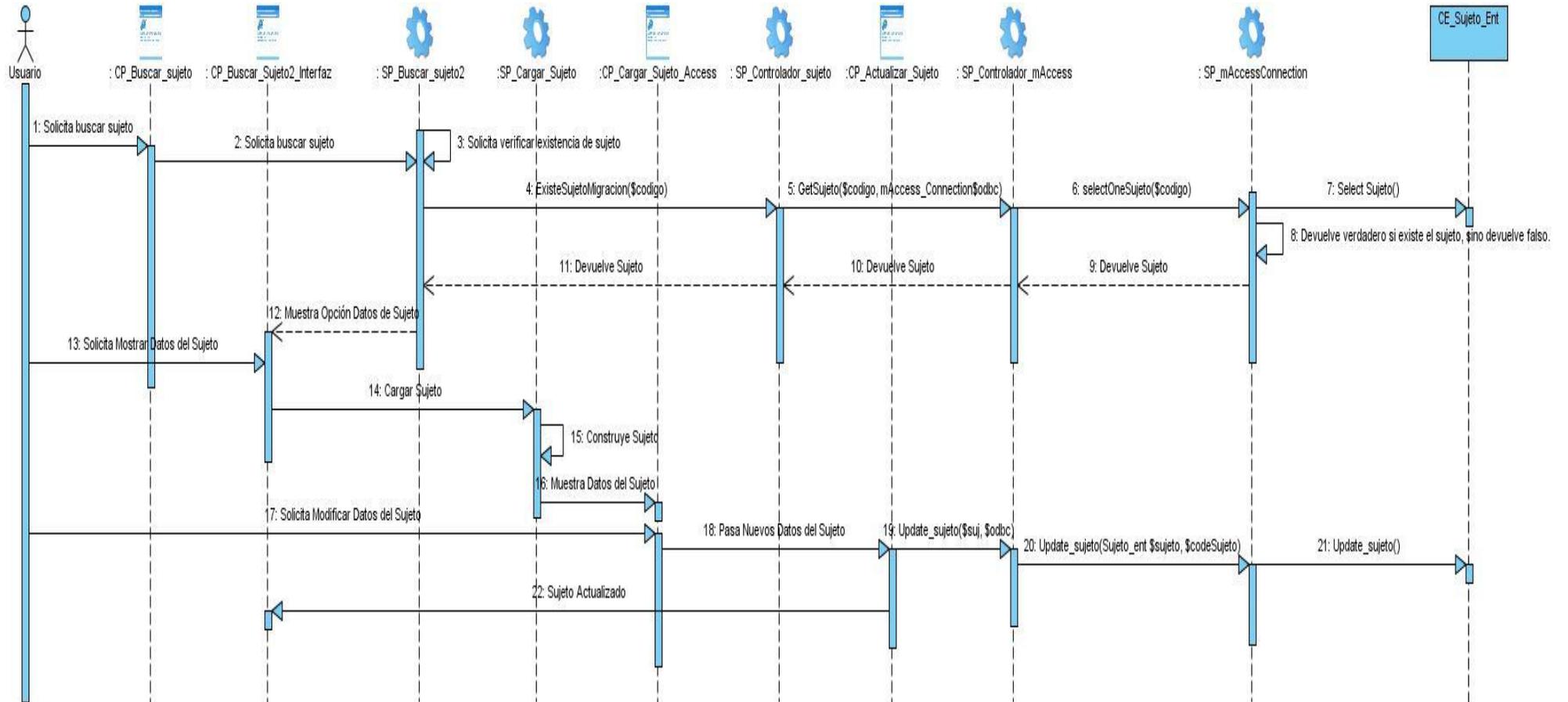
<http://www.arsys.info/programacion/bases-de-datos/ms-access/>

**2009.** Patrones y Antipatrones: una Introducción - Parte II. *Patrones y Antipatrones: una Introducción - Parte II.* [En línea] 2009. [Citado el: 20 de mayo de 2009.] [http://msdn.microsoft.com/es-](http://msdn.microsoft.com/es-es/library/bb972251.aspx#mainSection)

[es/library/bb972251.aspx#mainSection](http://msdn.microsoft.com/es-es/library/bb972251.aspx#mainSection)

# ANEXOS

**Fig.33 Diagrama de Secuencia: CU Modificar Sujeto Access.**



**Fig.34 Diagrama de Secuencia: CU Modificar Sujeto Propel.**

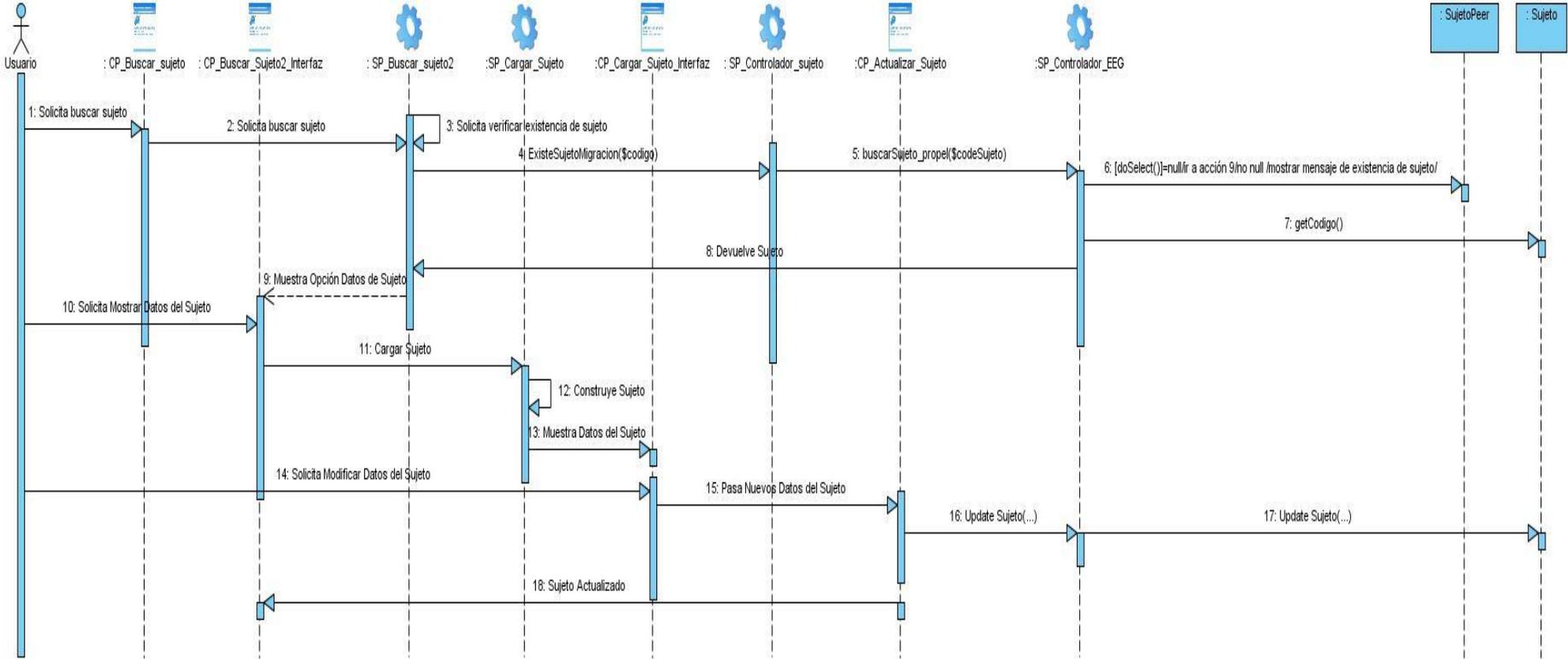
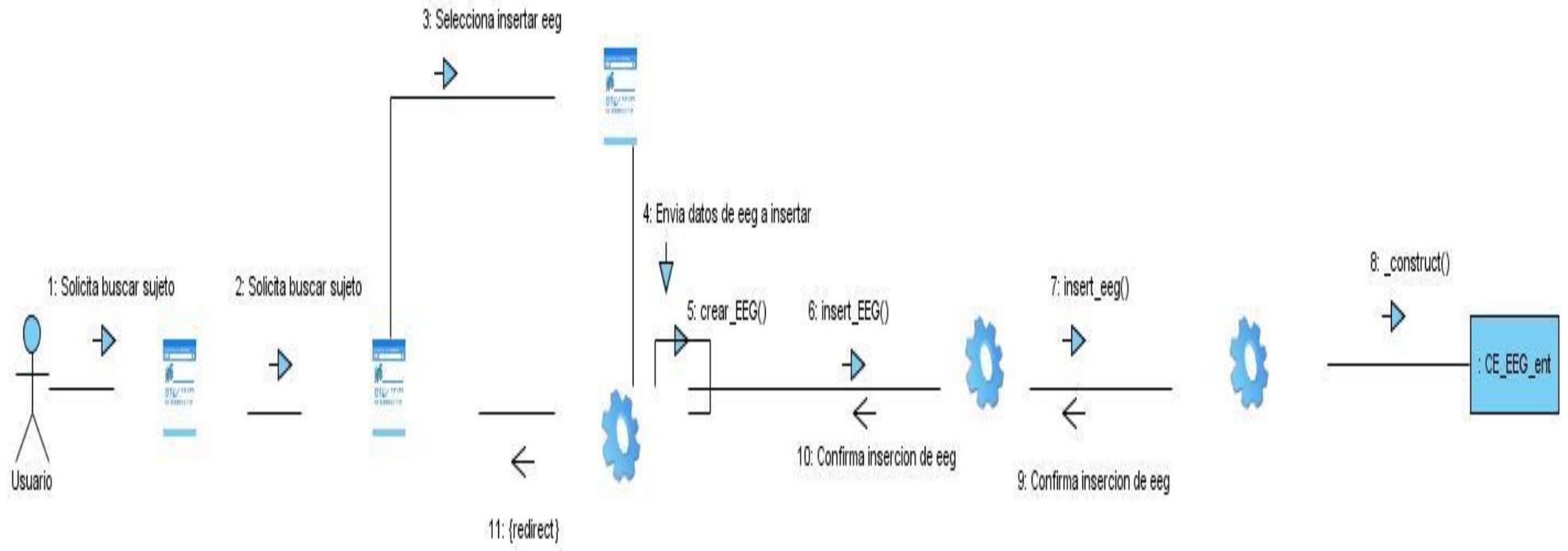


Fig.35 Diagrama de Colaboración: CU Insertar EEG Access.



**Fig.36 Diagrama de Colaboración: CU Insertar EEG Propel.**

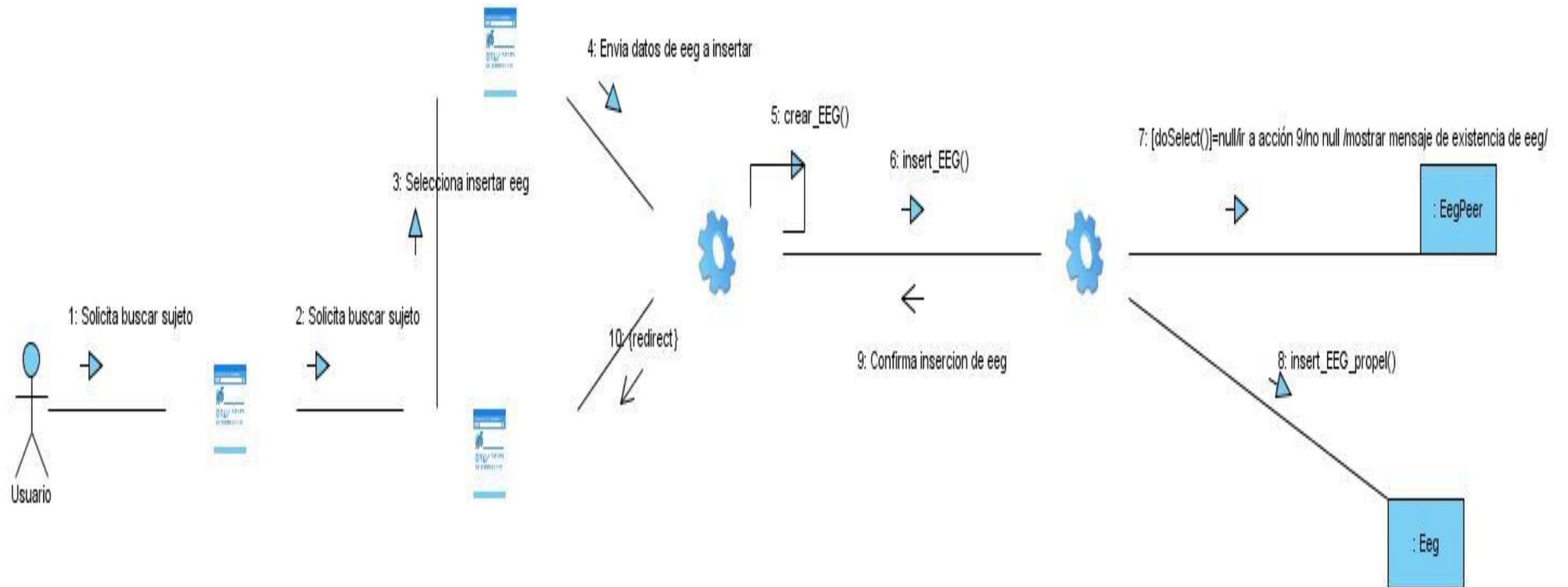
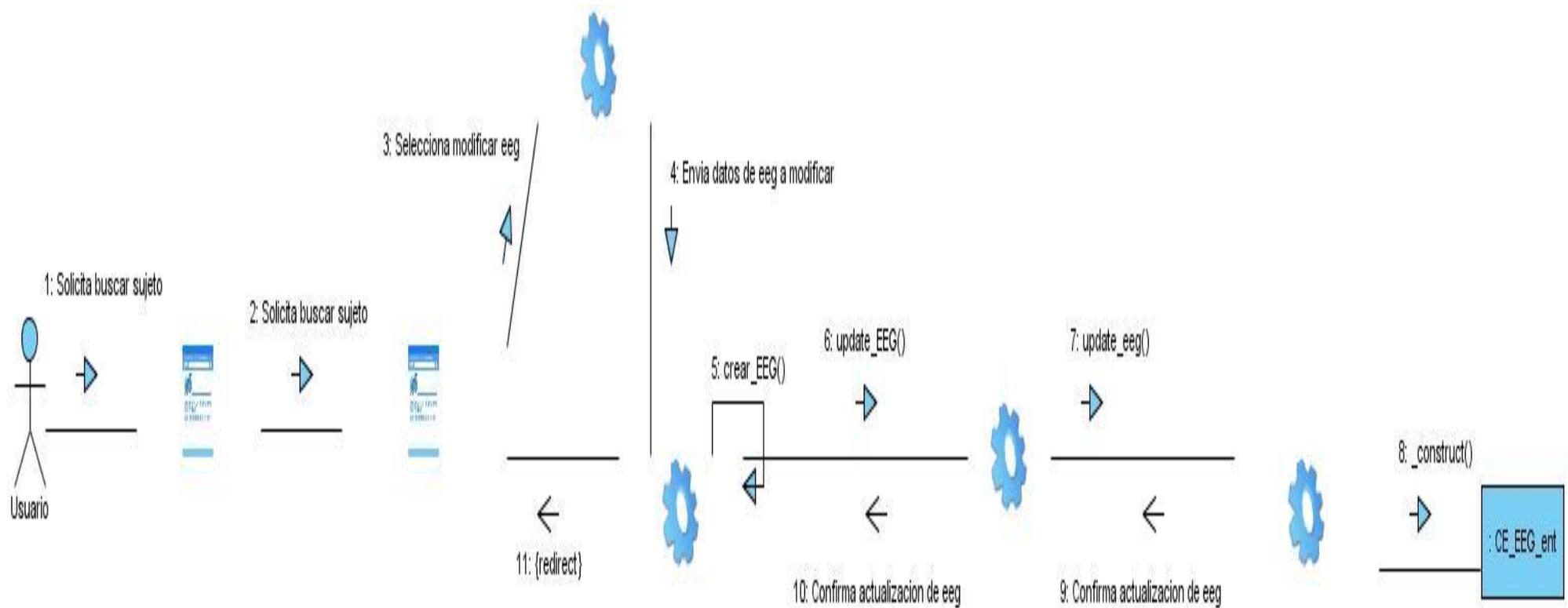


Fig.37 Diagrama de Colaboración: CU Modificar EEG Access.



**Fig.38 Diagrama de Colaboración: CU Modificar EEG Propel.**

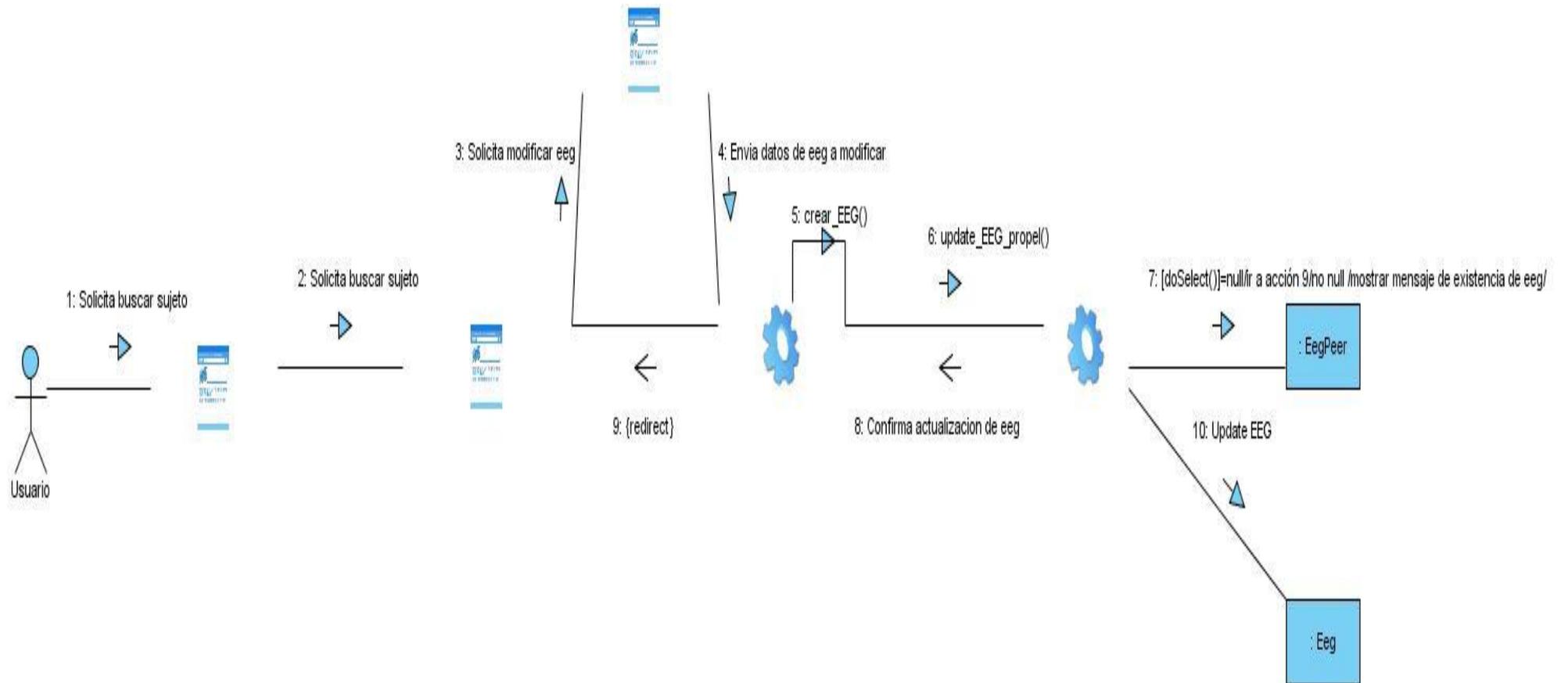


Fig.39 Diagrama de Colaboración: CU Eliminar EEG Access.

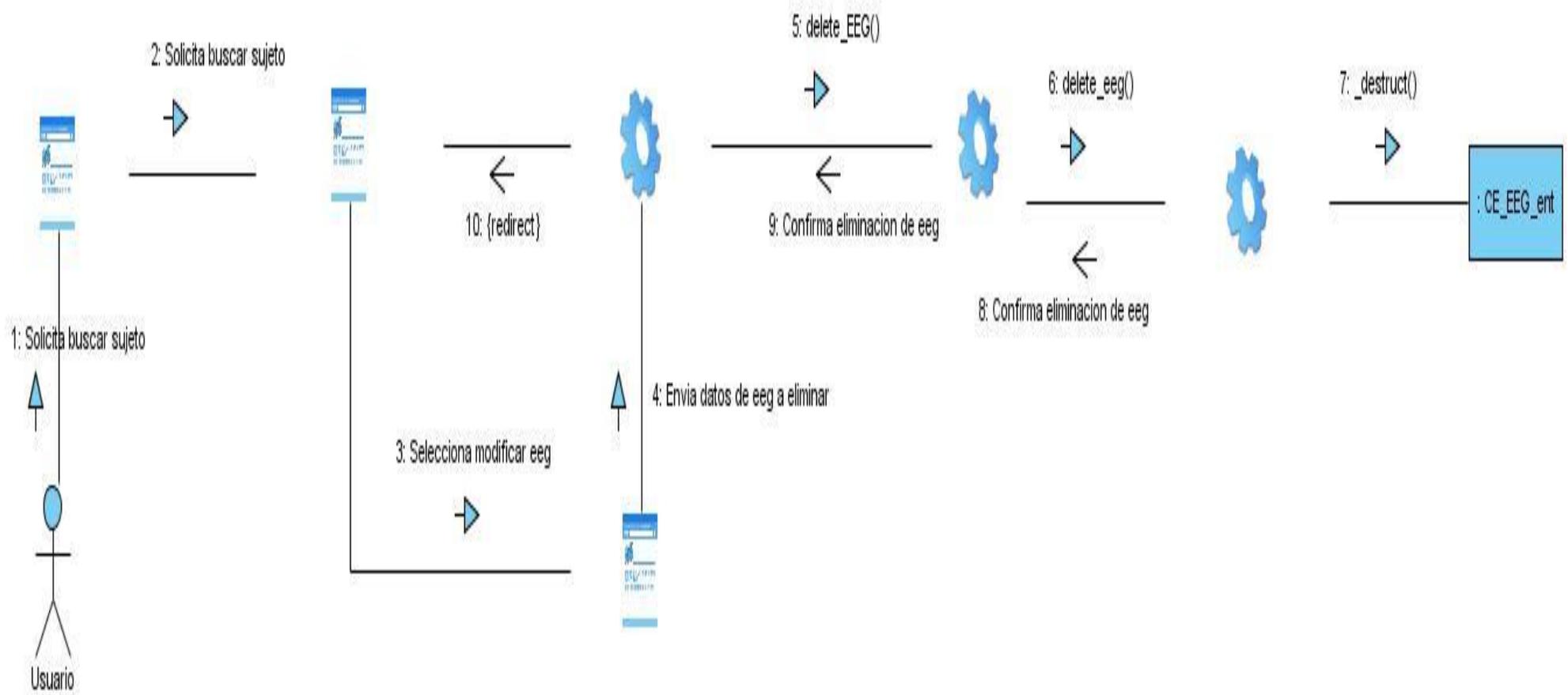
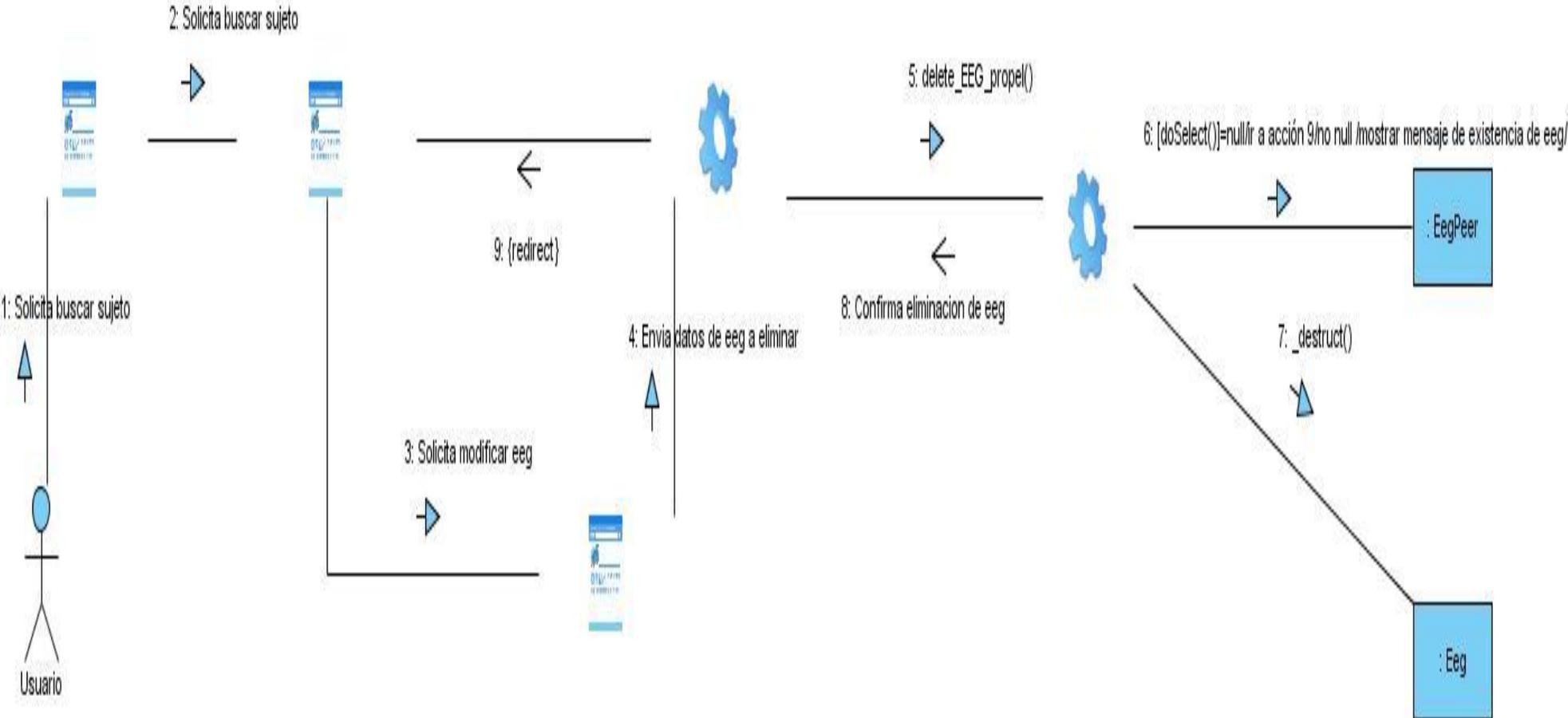
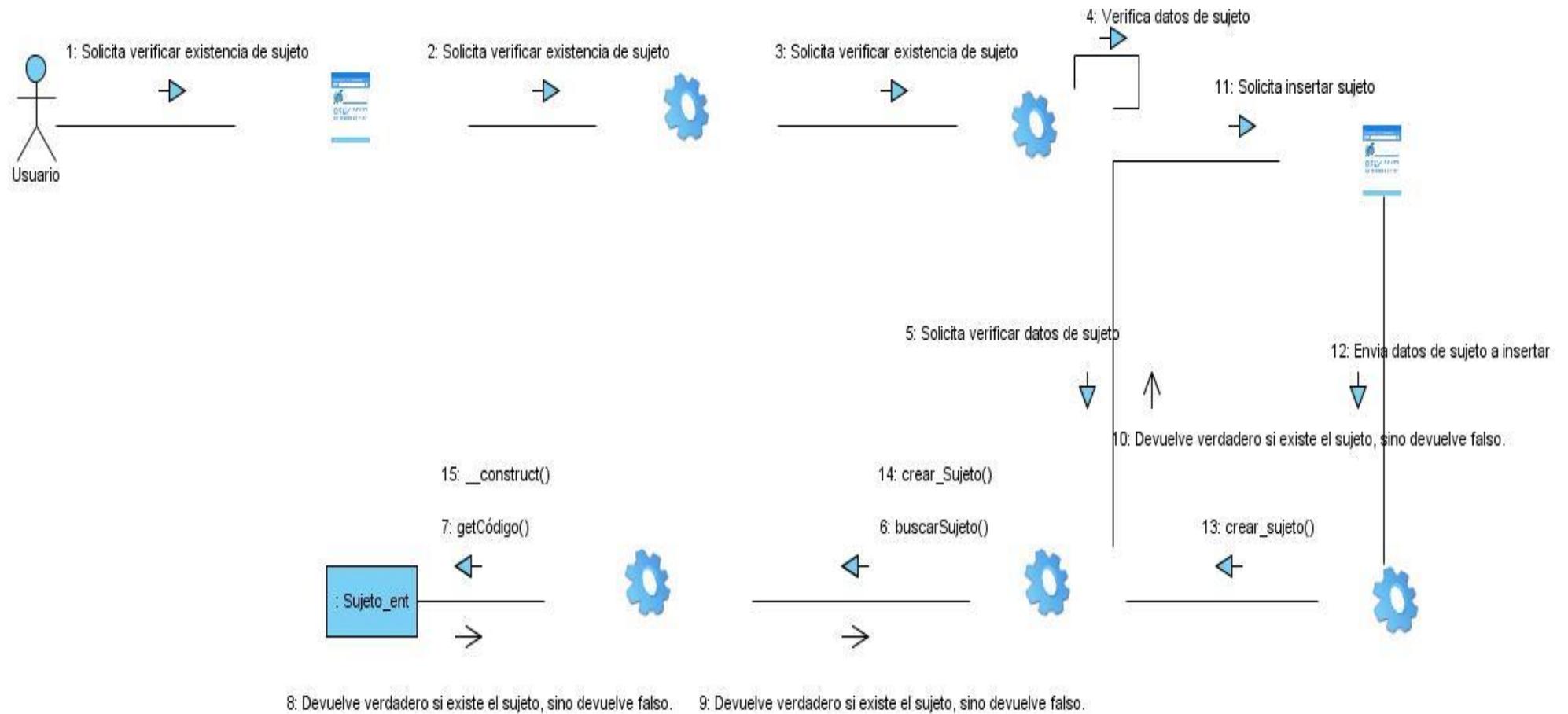


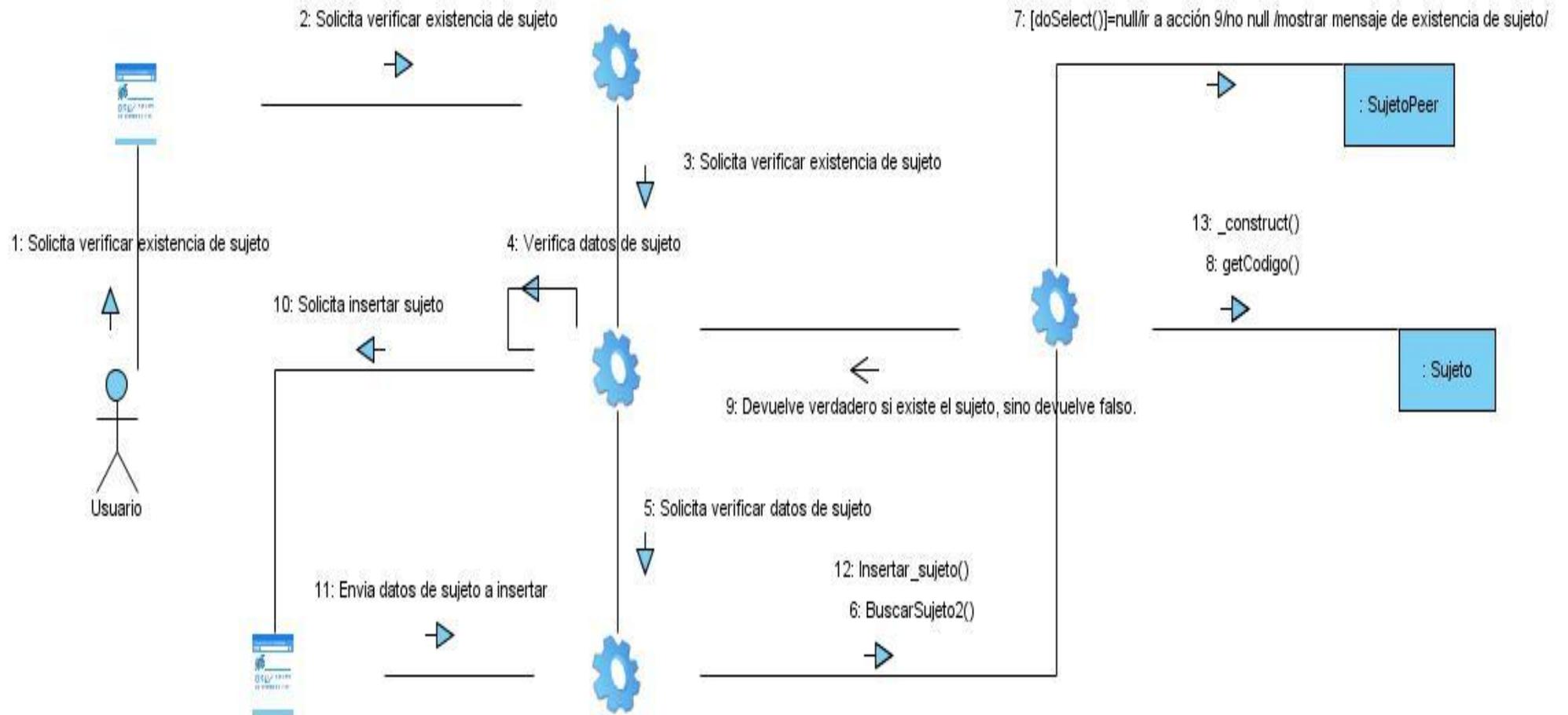
Fig.40 Diagrama de Colaboración: CU Eliminar EEG Propel.



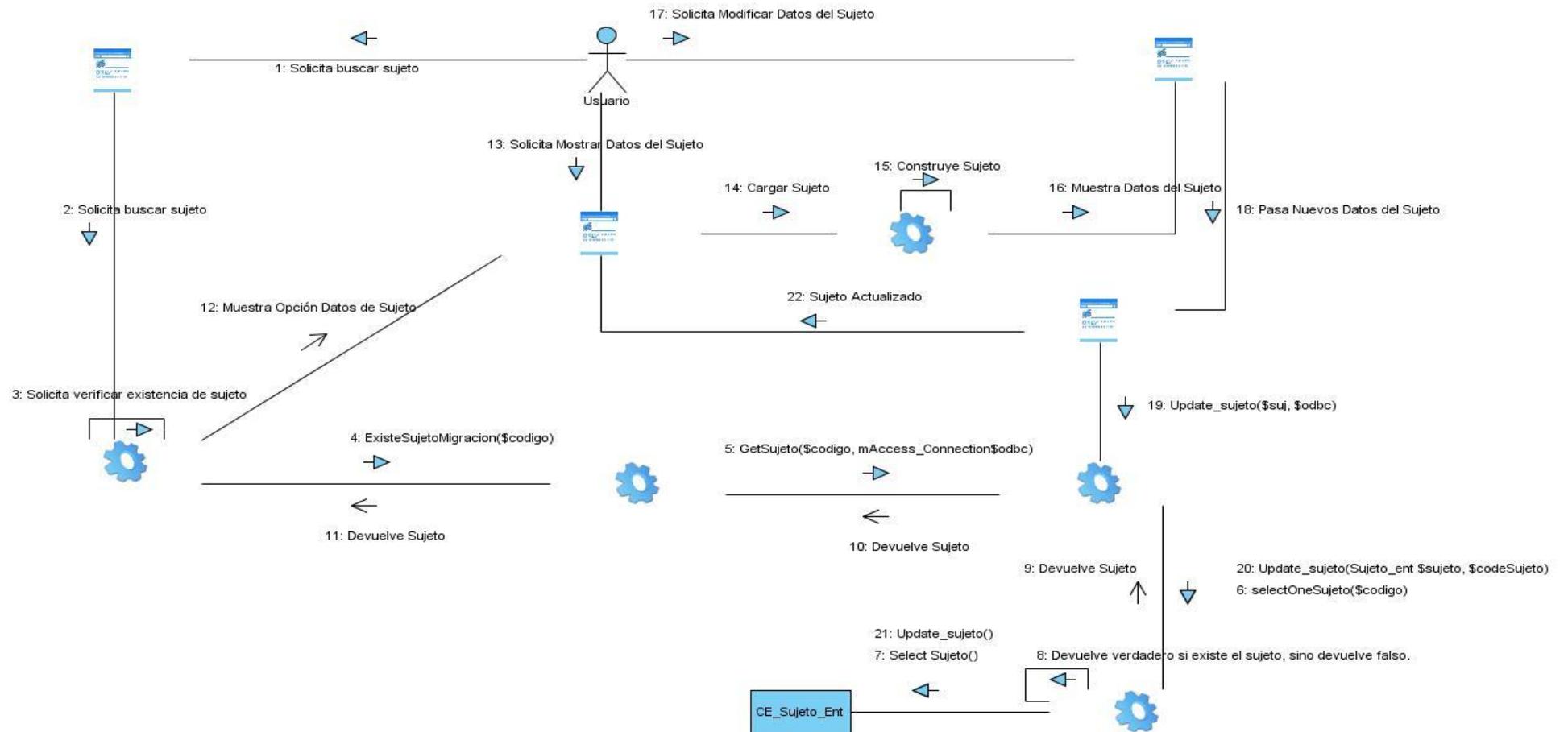
**Fig.41 Diagrama de Colaboración: CU Insertar Sujeto Access.**



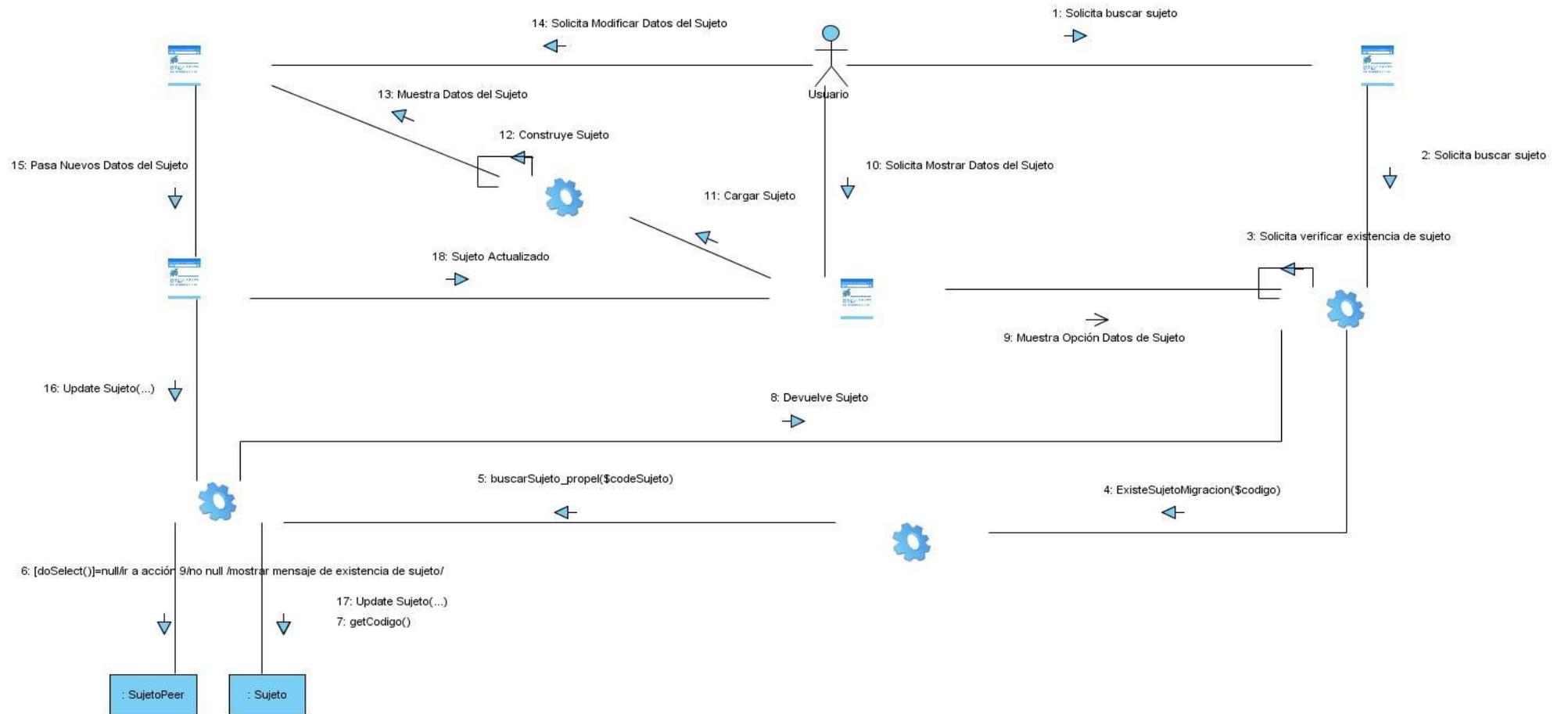
**Fig.42 Diagrama de Colaboración: CU Insertar Sujeto Propel.**



**Fig.43 Diagrama de Colaboración: CU Modificar Sujeto Access.**



**Fig.44 Diagrama de Colaboración: CU Modificar Sujeto Propel.**



## 1 GLOSARIO

2 **WYSIWYG:** El término WYSIWYG es el acrónimo de "What You See Is What You Get", que traducido  
3 al castellano quiere decir: "lo que ves es lo que obtienes", en los que de manera visual se pueden  
4 colocar distintos elementos sobre una vista previa de la página, encargándose el programa de generar  
5 el documento HTML.

6 **Unicode:** Es un estándar industrial cuyo objetivo es proporcionar el medio por el cual un texto en  
7 cualquier forma e idioma pueda ser codificado para el uso informático. El establecimiento de Unicode  
8 ha involucrado un ambicioso proyecto para reemplazar los esquemas de codificación de caracteres  
9 existentes, muchos de los cuales están muy limitados en tamaño y son incompatibles con entornos  
10 multilingües. Unicode se ha vuelto el más extenso y completo esquema de codificación de caracteres,  
11 siendo el más dominante en la internacionalización y adaptación local del software informático.

12 **SGML:** son las siglas de Standard Generalized Markup Language o "Lenguaje de Marcación  
13 Generalizado". Consiste en un sistema para la organización y etiquetado de documentos. La  
14 Organización Internacional de Estándares (ISO) ha normalizado este lenguaje en 1986.

15 **Test:** prueba, chequeo, control, ensayo, experimentación, experimento, tanteo; examen, test

16 **Open source:** Open Source es una marca de certificación propiedad de la Open Source Initiative. El  
17 software OpenSource se define por la licencia que lo acompaña, que garantiza a cualquier persona el  
18 derecho de usar, modificar y redistribuir el código libremente.

19 **ODBC:** (del inglés Open Database Connectivity Standard) Estándar de conectividad abierta de base de  
20 datos.

21 **Electroencefalograma (EEG):** es el registro de la actividad eléctrica de las neuronas del encéfalo.  
22 Dicho registro posee formas muy complejas que varían mucho con la localización de los electrodos y  
23 entre individuos. Esto es debido al gran número de interconexiones que presentan las neuronas y por  
24 la estructura no uniforme del encéfalo. (23)

25 **Imagen por Resonancia Magnética (IRM en español, o MRI, en inglés):** Una imagen por  
26 resonancia magnética también conocida como tomografía por resonancia magnética (TRM) o imagen  
27 por resonancia magnética nuclear (NMRI, por sus siglas en inglés) es una técnica no invasiva que  
28 utiliza el fenómeno de la resonancia magnética para obtener información sobre la estructura y  
29 composición del cuerpo a analizar. Esta información es procesada por ordenadores y transformada  
30 en imágenes del interior de lo que se ha analizado. Es utilizada principalmente en medicina para  
31 observar alteraciones en los tejidos y detectar cáncer y otras patologías.

1 **Resonancia Magnética Funcional (fMRI, del inglés: functional magnetic resonance imaging):** Es un  
2 procedimiento clínico y de investigación que permite mostrar en imágenes las regiones cerebrales  
3 que ejecutan una tarea determinada. El procedimiento se realiza en el mismo resonador utilizado  
4 para los exámenes de diagnóstico, pero con modificaciones especiales del software y del hardware.  
5 Para realizar una IRMf no se requiere inyecciones de sustancia alguna

6 **Tomografía Axial Computarizada (TAC):** Es un método de diagnóstico médico que permite obtener  
7 imágenes del interior del cuerpo humano mediante el uso de los Rayos X, a manera de rebanadas  
8 milimétricas transversales, con el fin de estudiarlo a detalle desde la cabeza hasta los pies. En un  
9 estudio convencional de rayos X el haz de radiación se emite de una manera difusa, pero en la  
10 tomografía axial computada (TAC) el haz está dirigido y tiene un grosor determinado que puede  
11 variar desde los 0.5 mm hasta 20 mm, dependiendo del tamaño de la estructura a estudiar.

12 **Tomografía por Emisión de Positrones (PET):** La tomografía por emisión de positrones (PET) es un  
13 tipo de procedimiento de medicina nuclear que mide la actividad metabólica de las células de los  
14 tejidos del cuerpo. La PET es en realidad una combinación de medicina nuclear y análisis bioquímico.  
15 Se utiliza principalmente en pacientes que tienen enfermedades del corazón o del cerebro y cáncer, la  
16 PET ayuda a visualizar los cambios bioquímicos que tienen lugar en el cuerpo, como el metabolismo  
17 (proceso por el cual las células transforman los alimentos en energía después de que han sido  
18 digeridos y absorbidos en la sangre) del músculo cardíaco.