



Universidad de las Ciencias Informáticas

Facultad 2

**Trabajo de Diploma para optar por el Título de
Ingeniero en Ciencias Informáticas**

MÓDULO REPORTES ESTADÍSTICOS PARA EL SISTEMA DE GESTIÓN PARA INGENIERÍA
CLÍNICA Y ELECTROMEDICINA (SIGICEM) v2.0.

Autores: Roberto Carlos Quintana Ross
Miguel Angel Sánchez Hurtado

Tutores: Ing. Dennys Hernández Peña
Ing. Idayana Bastarreche Calistre

La Habana, junio 2014
"Año 56 de la Revolución"

Datos de contacto

Nombre y apellidos: Idayana Bastarache Calistre

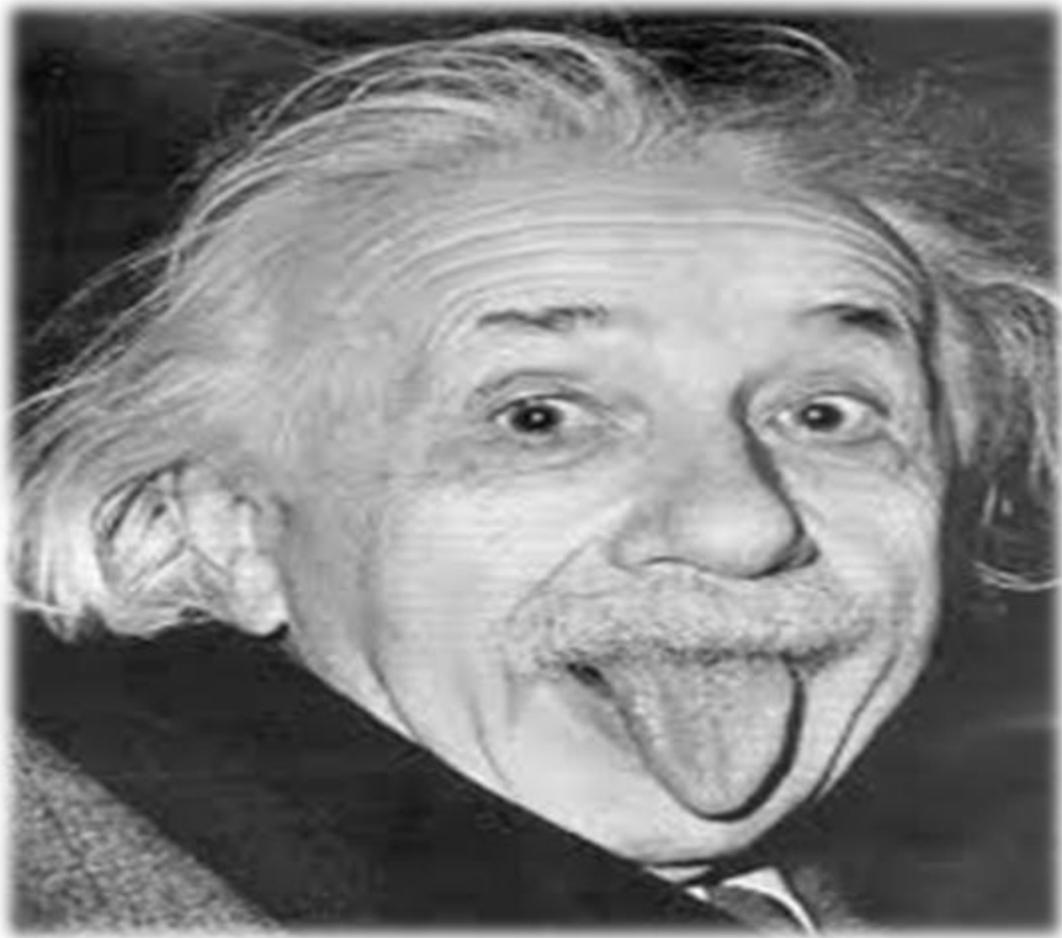
Graduado de Ingeniero en Ciencias Informáticas, en el año 2011, en la Universidad de las Ciencias Informáticas. Ha sido cotutora de dos trabajos de diploma en el departamento. Actualmente se desempeña como analista principal en el proyecto SIGICEM.

Correo electrónico: ibastarache@uci.cu

Nombre y apellidos: Denys Hernández Peña

Ingeniero informático graduado del curso 2004-2005. CUJAE. Profesor Asistente. Ha sido líder del grupo de desarrollo para la gestión de equipos médicos (GDEM) durante 8 años. Ha impartido las asignaturas de Seguridad Informática, Sistema Operativo, Comercio Electrónico, Metodología de la Investigación Científica, Programación e Ingeniería de Software. Tribunal de tesis por 5 años consecutivos y tutor de más de 25 trabajos de tesis de grado. Actualmente arquitecto principal del proyecto SIGICEM del Dpto. Sistemas Especializados y profesor adjunto a la Facultad 7.

Correo electrónico: dhernandezp@uci.cu



*"Los grandes espíritus siempre se han encontrado
con oposición violenta por parte de las mentes
mediocres "*

Albert Einstein

Agradecimientos

Agradezco a mi familia el apoyo desmesurado que me han brindado, en especial a mis padres Ida y Miguel y a mis otros padres Norma y Terry. Les agradezco a mis tutores y a aquellas personas que me ayudaron por sus críticas constructivas y buenos consejos. A mi eterna profesora Rosa de la Caridad que siempre estuvo presente en cada paso, en cada momento; al igual que al resto de los profesores que me han formado durante este periodo. A mi novia que siempre aportó su pedacito en este trabajo al igual que esfuerzo. En fin agradezco a todos mis compañeros de aula, a los de mi apartamento que de igual forma me ayudaron a llegar al fin de este largo y complicado camino.

Miguel Angel

Agradezco a toda mi familia por siempre estar al pendiente de estos 5 largos años de Universidad, en especial a mis padres por estar siempre atentos de mis cosas y por inculcarme el sentimiento de superación y que todas las metas son posibles si uno se esfuerza. Le agradezco a Diana por compartir juntos todo la vida universitaria y ayudarme en todo momento, a la vida por ponerme el reto de ser papá y enfrentarme a los estudios desde una perspectiva diferente, a todas mis amistades, en especial al piquete del apartamento 81106, que conformamos una familia y nos apoyamos en todas las cosas, a mi colega de tesis y amigo por confiar en mi para realizar juntos nuestro Trabajo de Diploma y a mis tutores por su colaboración y consejos para concluir con éxito la investigación.

Roberto Carlos

Agradecimientos especiales para las personas que hicieron posible la culminación de una forma u otra de este trabajo. A los monstruos informáticos: Arieskjen, Nelson, Adrian, Carlos Javier, Ranniel y Luismel, así como a las maestras de las letras Maryelines y Lianne.

Miguel Angel y Roberto Carlos

Dedicatoria

Le dedico este trabajo a alguien super especial que fue, es y será ejemplo de todo en esta vida, aunque ya no estes, este trabajo es tan tuyo como mío, para ti mi tía del alma: Felicia Hurtado Poll.

*Miguel
Angel*

Le dedico este trabajo a mis padres y a mis hijitas Camila y Carla, ojalá que algún día les sirva de objeto de estudio en su vida y a una persona que aunque no esté hoy presente fue un ejemplo a seguir: mi abuelito Israel Ross Sardiñas, pese a no tener estudios universitarios sus consejos fueron una guía para que hoy este trabajo sea un sueño hecho realidad.

*Roberto
Carlos*

Resumen

Una parte fundamental e indispensable de los sistemas informáticos en la actualidad es la generación de los reportes estadísticos, los cuales se han usado desde hace más de una década en centros médicos debido a sus particularidades. Todo esto deviene de la gran cantidad de información existente con las que trabajan y la importancia de su uso.

La Universidad de Ciencias Informáticas brinda este servicio a través de un sistema de gestión de información para las unidades de ingeniería clínica y electromedicina. Para ello posee un módulo de reportes estadísticos, con el fin de supervisar y controlar los equipos médicos. En este trabajo se presenta la segunda versión de dicho módulo, el cual basa su funcionamiento en componentes desarrollados con PHP 5.1, los frameworks Symfony 1.4 y ExtJS 2.2, el cual genera un nuevo grupo de reportes, en este caso gráficos, capaces de suplir con nuevas funcionalidades y mejorar las ya existentes.

Las nuevas funcionalidades aplicadas a los reportes rigieron el desarrollo de los mismos, permitiendo la entrega al cliente de una solución confiable y consistente, que puede ser utilizada para estudios de eventos ocurridos con el equipamiento médico. Facilitando, con la existencia de nuevas estrategias y funciones, la agilidad para la toma de decisiones para los especialistas vinculados con el sistema.

Palabras claves: electromedicina, framework, ingeniería clínica, reportes estadísticos.

Índice de contenido

INTRODUCCIÓN.....	1
Capítulo I: Fundamentación teórica.....	7
1.1 Términos relacionados con la investigación.....	7
1.2 Sistemas para la generación de reportes estadísticos	8
1.3 Metodología de desarrollo de software	13
1.4 Herramientas y tecnologías de desarrollo utilizadas.....	14
Capítulo II: Características del sistema.....	36
2.1 Modelo de dominio	36
2.2 Conceptos Fundamentales del Dominio.....	36
2.3 Propuesta del sistema.....	39
2.4 Especificación de los requisitos de software.....	40
2.5 Modelo de casos de uso del sistema.....	43
2.6 Descripción textual de los casos de uso	46
Capítulo III: Análisis y diseño del sistema.....	36
3.1 Modelo de análisis	36
3.2 Diagramas de interacción. Comunicación.....	38
3.3 Descripción de la arquitectura	40
3.4 Patrones del diseño	43
3.5 Modelo de diseño	45
Capítulo IV: Implementación del sistema.....	49
4.1 Modelo de datos	49
4.2 Modelo de implementación.....	56
4.3 Estilos y estándares de codificación.....	59

4.4 Tratamiento de errores.....	62
Conclusiones.....	65
Recomendaciones	66
Referencias Bibliográficas	67
Bibliografía	73
Glosario de términos	78
Anexos.....	79

Índice de tablas

Tabla 1: Requisitos funcionales.	41
Tabla 2: Requisito No Funcional Software.....	43
Tabla 3: Requisito No Funcional Hardware	43
Tabla 4: Actores del sistema.....	44
Tabla 5: Descripción textual del caso de uso: Generar reporte personalizado de equipos médicos	46
Tabla 6: Descripción textual del caso de uso: Generar reporte gráfico estado técnico y disponibilidad técnica por unidad de subordinación nacional	47
Tabla 7: Descripción textual del caso de uso: Generar reporte gráfico disponibilidad técnica por provincia	48
Tabla 8: Modelo Vista Controlador SIGICEM.....	42
Tabla 9: Descripción de la tabla: t_equipo	52
Tabla 10: Descripción de la tabla: t_especialista.....	53
Tabla 11: Descripción de la tabla: t_mantenimiento	54
Tabla 12: Descripción de la tabla: t_pieza	55
Tabla 13: Descripción de la tabla: t_subordinación	55
Tabla 14: Descripción de la tabla: t_unidad_salud	56
Tabla 15: Estándares y Estilos de Codificación.....	62
Tabla 16: Generar reporte personalizado de equipos médicos.....	81
Tabla 17: Generar reporte gráfico estado técnico y disponibilidad técnica por subordinación nacional.....	84
Tabla 18: Generar reporte gráfico disponibilidad técnica por provincia	87

Índice de figuras

Figura 1: Diagrama del modelo de dominio.	39
Figura 2: Paquetes reportes gráficos	45
Figura 3: Paquetes reportes generales	45
Figura 4: Diagrama de Clases de Análisis: Generar reporte personalizado de equipos médicos.	37
Figura 5: Diagrama de Clases de Análisis: Generar reporte gráfico estado técnico y disponibilidad técnica por unidad de subordinación nacional	38
Figura 6: Diagrama de Clases de Análisis: Generar reporte gráfico disponibilidad técnica por provincia ..	38
Figura 7: Diagrama de Comunicación: Generar reporte personalizado de equipos médicos.	39
Figura 8: Diagrama de Comunicación: Generar reporte gráfico estado técnico y disponibilidad técnica por unidad de subordinación nacional.....	39
Figura 9: Diagrama de Comunicación Generar reporte gráfico disponibilidad técnica por provincia	40
Figura 10:Estructura Modelo Vista Controlador	41
Figura 11:Elemplo del patrón Creador	44
Figura 12:Elemplo del patrón Alta Cohesión.....	44
Figura 13:Diagrama de clases del Diseño: Generar reporte personalizado de equipos médicos.....	46
Figura 14:Diagrama de clases del Diseño: Generar reporte gráfico disponibilidad técnica por provincia. ..	47
Figura 15:Diagrama de clases del Diseño: Generar reporte gráfico disponibilidad técnica por subordinación nacional.	47
Figura 16:Modelo de datos	50
Figura 17:Diagrama de Despliegue Módulo Reportes Estadísticos v2.0.	59
Figura 18:Diagrama de Componentes Módulo Reportes Estadísticos V2.0.	58
Figura 19:Ejemplo de Cuadros de Selección utilizados	63
Figura 20:Ejemplo de Menú de Selección utilizados	63

Figura 21:Interfaz Principal Módulo Reportes Estadísticos v2.0	87
Figura 22:Interfaz Reporte Gráfico Estado Técnico por Subordinación	88
Figura 23:Interfaz Reportes Gráficos	88

INTRODUCCIÓN

En la nueva era de paradigmas informáticos, los medios como las computadoras y las comunicaciones, permiten procesar y transmitir información desde lugares distantes en solo segundos a un ordenador. Las Tecnologías de la Información y las Comunicaciones (TIC) en la sociedad cumplen un papel fundamental, pues sus elementos aumentan la productividad y el número de empleos, cambiando para beneficio la calidad de vida de las personas ya que el tiempo y la distancia dejan de ser inconvenientes.

Uno de los renglones económicos que mueve la economía de un país y sus políticas, es el uso correcto de la informática como recurso indispensable para su desarrollo. Al respecto el 5 de abril de 1987, en la clausura del V Congreso de la Unión de Jóvenes Comunistas, Fidel Castro Ruz expresó: “Creo que el socialismo va a ser muy difícil de construir plenamente sin la computación, porque la necesita todavía más que la sociedad capitalista, y la sociedad capitalista hoy no podría vivir sin la computación”. (1)

En esta época de aperturas informáticas, en las ciencias de la salud y la medicina se han realizado grandes investigaciones, la manera de trabajar en estas disciplinas ha ido en evolución y se ha podido agilizar el trabajo de especialistas y médicos, brindando atención a un número considerable de enfermedades y procesos del cuerpo humano con el uso de tecnologías informáticas capaces de mostrar áreas de la complejidad humana desconocidas.

En Cuba, se trabaja actualmente en el desarrollo de un grupo de aplicaciones cuya implementación es posible gracias a la participación activa del Ministerio de la Informática y Comunicaciones y del Ministerio de Salud Pública. Los proyectos que se acometen están relacionados con la actividad de la atención primaria de salud, la actividad hospitalaria, las redes de especialidades como la de los servicios de nefrología e imagenología, neurociencias y genética médica, el desarrollo de software educativos y sistemas que permiten tener centralizada la información útil para la planificación y toma de decisiones generadas por cada uno de los anteriores de manera confiable y oportuna. (2)

Debido a los avances de la informática en el país muchos centros se han dado a la tarea de informatizar los procesos que llevan a cabo. Una de estas instituciones es el Centro Nacional de

Electromedicina (CNE) que incluye la vinculación de la Ingeniería Clínica con la Electromedicina, con vista a la necesidad de llevar un registro tecnológico, ficha técnica y estado de cada equipo. Esto conllevó a la creación de un moderno sistema que permite realizar la gestión de la información clínica y electromédica actual, en las instituciones médicas del país, al mismo se le denominó Sistema de Gestión para Ingeniería Clínica y Electromedicina (SIGICEM).

La Universidad de las Ciencias Informáticas (UCI) es un centro docente y productivo, surgido como parte de los proyectos de la Batalla de Ideas, que busca nuevas oportunidades en el campo de la investigación y la sustitución de importaciones en Cuba. En la actualidad, SIGICEM se encuentra en desarrollo por un grupo de trabajo perteneciente al Departamento Sistemas Especializados en Salud (SES) del Centro de Informática Médica (CESIM).

Esta aplicación posee entre sus potencialidades, el módulo Reportes Estadísticos, el cual funciona como un mecanismo de apoyo a la toma de decisiones por parte de los directivos de la actividad técnica de la empresa. Una vez culminada la primera versión del SIGICEM y desplegada en los centros provinciales de ingeniería clínica del país, los especialistas vinculados en el desarrollo se dieron a la tarea de comenzar a implementar la segunda versión, basados en las diversas dificultades, necesidades de mejoras identificadas y las posibilidades técnicas existentes para lograr un producto de mayor robustez.

En este sentido se han realizado un conjunto de peticiones de la dirección de vigilancia tecnológica del Ministerio de Salud Pública. Las mismas requieren de la extensión del módulo para garantizar que se obtengan nuevos tipos de reportes estadísticos, pues este no cuenta con una manera gráfica de mostrar los datos reales de la disponibilidad técnica y estado técnico de los equipos médicos que están en función de los diferentes niveles de organización y los principales servicios que se brindan en cada área de salud.

La disponibilidad técnica evalúa el por ciento que representan los equipos médicos en estado funcionando del total de equipos, la cual es de mucha ayuda para la toma de decisiones de los directivos. El estado técnico, es el comportamiento que presenta un equipo médico la cual se describe su atendiendo a su rendimiento. En el sistema aparecen los siguientes estados: funcionando, por instalar, defectuoso, roto, pendiente a baja técnica y baja técnica.

Algunas deficiencias identificadas son las limitaciones inherentes a las librerías utilizadas para la creación de documentos en formato pdf y hojas de cálculo Excel, las cuales no presentan los mecanismos implícitos que son esenciales para crear reportes gráficos. Estos mecanismos son las distintas funciones gráficas como la de barras, pasteles, líneas y puntos; funciones matemáticas las cuales comparan los datos capaces de informar en tiempo real lo que sucede en las actividades de los procesos y el control de la información necesaria para la correcta toma de decisiones.

Estas carencias en cuanto al funcionamiento han conllevado al desarrollo de nuevas estrategias para la generación de reportes y la reimplementación de muchas de sus tareas básicas para poder cumplir con las exigencias de los especialistas que las utilizan.

Otro inconveniente que ha surgido es la invariabilidad de los reportes, limitando potencialmente la manualidad del experto. Esta situación se manifiesta en los informes estáticos, que no poseen paneles de búsqueda avanzada, ni filtros en las cabeceras, provocando lentitud en el procesamiento y manejo de la información.

Por lo anteriormente expuesto se presenta el siguiente **problema a resolver**: El proceso de obtención de reportes del Sistema de Gestión para Ingeniería Clínica y Electromedicina carece de un mecanismo capaz de corregir deficiencias existentes y suplir las nuevas necesidades de la dirección de vigilancia tecnológica.

Definiéndose como **objeto de estudio**: Proceso de gestión de la información en los sistemas generadores de reportes estadísticos, enmarcado en el **campo de acción**: El módulo Reportes Estadísticos del Sistema de Gestión para la Ingeniería Clínica y Electromedicina.

Todo ello encaminado a cumplir el siguiente **objetivo general**: Desarrollar la versión 2.0 del módulo Reportes Estadísticos del Sistema de Gestión para la Ingeniería Clínica y Electromedicina.

Para dar cumplimiento al objetivo general se definieron las siguientes tareas de la investigación:

- Establecer los fundamentos teórico-metodológicos para el desarrollo del módulo Reportes Estadísticos del Sistema de Gestión para la Ingeniería Clínica y Electromedicina.

- Caracterizar el proceso de gestión de reportes en el Sistema de Gestión para la Ingeniería Clínica y Electromedicina.
- Identificar la metodología, herramientas y tecnologías que deben sostener el Módulo Reportes Estadísticos del Sistema de Gestión para la Ingeniería Clínica y Electromedicina v2.0.
- Desarrollar el módulo Reportes Estadísticos v2.0 para el Centro de Ingeniería Clínica y Electromedicina.

Entre los beneficios esperados con el desarrollo de la investigación se destacan:

- Gestión la información referente al equipamiento médico instalado en las unidades de salud del país, cumpliendo con las nuevas características establecidas en el centro.
- Determinación para la toma de decisiones ante eventos ocurridos con el equipamiento médico.
- Concepción de las bases para la monitorización y el aseguramiento del proceso de gestión tecnológica del equipamiento médico, a través del perfeccionamiento y la automatización de la actividad postmercado.
- Definición de las principales premisas a tener en cuenta para evitar la importación innecesaria de piezas de repuesto. Utilizadas en la reparación de equipos médicos, a través de la información obtenida por medio del módulo Reportes Estadísticos del Sistema de Gestión para la Ingeniería Clínica y Electromedicina v2.0.

Una vez conocidas las tareas a enfrentar, se hizo necesario el estudio, análisis y definición de los métodos científicos que se utilizarían para lograr la solución deseada, con los cuales se logra conocer las principales características; beneficiando que la investigación sea más rápida y eficiente, en la gestión de la información. Para el desarrollo de la misma se emplearon los siguientes métodos:

Teóricos:

- **Análítico-sintético:** Son dos procesos que buscan la esencia de los fenómenos, los rasgos que lo caracterizan, su objetivo es analizar las teorías y documentos permitiendo la extracción de los elementos importantes que se relacionan con el objeto de estudio. Se utilizó para reestructurar el proceso de generación de reportes, así como para las mejoras en el desarrollo del software y de la documentación, permitiendo conocer los rasgos y tendencias de las nuevas tecnologías aplicadas en la solución.
- **Histórico-lógico:** Permite estudiar de forma analítica la trayectoria histórica real de los fenómenos, su evolución y desarrollo. Su objetivo es constatar la evolución de un determinado fenómeno en un período de tiempo, en toda su trayectoria o en un fragmento temporal de la lógica de su desarrollo. Necesario para el conocimiento evolutivo del software, sus etapas, funciones principales y desarrollo de cada proceso.

- Inductivo-deductivo: Son formas de razonamiento que permiten llegar a un grupo de conocimientos generalizadores, tanto desde el análisis de lo particular a lo general, como desde el análisis de elementos generalizadores a uno de menor nivel de generalización. Importante para arribar tanto a conclusiones particulares como generales del proceso de generación de reportes.
- Modelación: El modelo científico es un instrumento de carácter material o teórico, creado para reproducir el objeto que se está estudiando. Constituye una reproducción simplificada de la realidad que cumple una función heurística que permite descubrir nuevas relaciones y cualidades del objeto de estudio. Empleado en la búsqueda de alternativas y estrategias en el ámbito de la generación de reportes.

Empíricos:

- Grupo de discusión: Método que sirve para la toma de decisiones entre un conjunto de personas sobre un tema determinado, necesario para la adquisición de información. Aplicado para establecer una comparación de los resultados obtenidos por diferentes vías, a partir de criterios establecidos que permitan obtener una mayor precisión en la información recogida.
- Observación científica: Consiste en apreciar, ver, analizar un objeto, un sujeto o una situación determinada, con la orientación de un guía o cuestionario, para guiar la observación. Aplicado en la búsqueda y recolección de información referente al objeto en estudio. (3)

El contenido desarrollado queda estructurado por:

Capítulo 1 Fundamentación teórica: Posee una percepción de los conceptos fundamentales y un estudio del estado del arte de las herramientas que se utilizan a nivel internacional, nacional y en el ámbito de la Universidad para la generación de reportes gráficos y estadísticos. Se plantean las técnicas, metodología y tecnologías utilizadas en la aplicación, para contribuir a la calidad y éxito de la investigación.

Capítulo 2 Características del sistema: En esta sección se aborda sobre las nuevas singularidades del proceso de gestión de reportes a través de grupos de discusión realizados directamente a ejecutivos del Centro de Ingeniería Clínica y Electromedicina (SIGICEM) para la implementación de la solución

propuesta. De esta manera se realiza una descripción detallada del modelo de negocio definido y se especifican los requisitos funcionales y no funcionales a partir de los cuales se representan los casos de uso del sistema y la descripción de los mismos.

Capítulo 3 Análisis y diseño del sistema: Se especifican los artefactos del flujo de trabajo de análisis y diseño, estableciendo las bases para el desarrollo del módulo Reportes Estadísticos para el Sistema de Gestión para la Ingeniería Clínica y Electromedicina (SIGICEM) v2.0.

Capítulo 4 Implementación del sistema: Se describe la evolución de la solución a través del modelo de implementación. Donde se evidencian los estilos y estándares de la codificación de la solución propuesta.

Capítulo I: Fundamentación teórica.

El capítulo tiene como objetivo realizar una breve descripción de los conceptos fundamentales relacionados con la investigación. Se explican las características de sistemas homogéneos que se utilizan para la realización de reportes actualmente en el mundo. Se ofrece un análisis de la metodología, tecnologías y herramientas a emplear en el desarrollo de la solución propuesta.

1.1 Términos relacionados con la investigación

La informática en la actualidad genera en el mundo entero un gran cúmulo de información, la cual puede ser información dudosa o información precisa, las mismas se ven envueltas en grandes debates en las redes y son usadas muchas veces para verificar datos, hechos y conocimientos predefinidos sobre un tema o de manera general. Muchas de estas informaciones viajan sobre una red privada o entre la red de redes integrándose una con otras y formando un dominio que converge hacia nuevas vías de aprovechamiento.

La **información** está constituida por un grupo de datos ya supervisados y ordenados, que sirven para construir un mensaje basado en un cierto fenómeno o ente. La información permite resolver problemas y tomar decisiones, ya que su aprovechamiento racional es la base del conocimiento. (4)

La información precisa es aquella que cabe entre la clasificación de ser:

- **Información privilegiada:** es el tipo de información que solamente un grupo de personas conoce o puede tener acceso a ella. Por lo tanto, es información que no es pública o que, en el mejor de los casos, su conocimiento es muy restringido.
- **Información pública:** es el tipo de información que es abierta para todo el público y accesible. Por lo tanto, cualquier persona la puede conocer.
- **Información confidencial:** como su nombre lo indica, se trata de información secreta y no pública. Por lo tanto, solo la pueden conocer un círculo muy cerrado de personas. Generalmente es información sensible e importante que por ello no puede ser divulgada.
- **Información externa:** es el tipo de información que se hace pública según ciertos parámetros de construcción. Es decir, es un tipo de información creada para un fin en específico.

- **Información interna:** es el tipo de información que se maneja al interior de un grupo de personas. En realidad no es un tipo de información especial, sino que generalmente solo les interesa a esas personas.
- **Información personal:** es el tipo de información que presentan los datos sobre la vida de una personal. (5)

La información derivada según su clasificación es utilizada muchas veces para la toma de decisiones importantes y para el uso de reportes que corroboren su uso y desempeño en un campo o esfera dada. Estos reportes significan el cambio total del paradigma que tenía fijado la información y es debido que permite una serie de beneficios, capacitando al usuario o portador de la información de un nuevo punto de vista de la misma.

Empezando así una nueva forma de gestionar informaciones a nivel mundial y con numerosos frutos, empeñados en prevalecer en el orden mundial. En el ámbito de la informática, los reportes son informes que organizan y exhiben la información contenida en una Base de Datos (BD). Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios. El reporte, de esta forma, confiere una mayor utilidad a los datos. (6)

Dentro de los reportes están los llamados reportes estadísticos los cuales “sentaron las bases para la generación de estadísticas locales sobre disimiles esferas, los cuales permiten obtener información rápida para la toma de decisiones y facilita la generación de evidencias. A partir de dichos reportes, es posible saber en línea, o sea, al instante (a través de un clic), la cantidad de datos, características demográficas, servicios a los cuales consultar, posibles problemas que existirían o existentes.” (7)

1.2 Sistemas para la generación de reportes estadísticos

Internacionales:

- PHP ReportMaker:

PHP ReportMaker está diseñado para la alta flexibilidad, las páginas web generadas son en formato PHP, numerosas opciones le permiten generar los reportes que mejor se adapten a sus necesidades. Los códigos generados son limpios y asequibles de personalizar. Los scripts PHP se pueden ejecutar en los servidores Windows (MySQL / PostgreSQL / Access / MSSQL / Oracle) o servidores Linux / Unix (MySQL / PostgreSQL / Oracle). (8)

Características de PHP ReportMaker:

- Informes de resumen y detalle.
- Vista compacta sólo del resumen.
- Informe de tablas de contingencia.
- Informes de desglose.
- Intervalos de agrupación.
- Filtros extendidos con valor predeterminado y validación.
- Resumen opcional a distintos niveles.
- Editor de menús.
- Crea, modifica y coloca vistas.
- Eventos del servidor y secuencias de comandos del cliente.
- Exporta a formatos HTML/Word/Excel/correo electrónico/PDF.
- Carga de tabla dinámica.
- Plantilla personalizable.
- Sincroniza los ajustes de proyectos con los cambios en la base de datos. (9)

- JasperReport:

Es el motor de informes Java más utilizado del mundo. Permite combinar fuentes de datos y producir documentos “pixel perfect” para su visualización, impresión y exportación a una variedad de formatos gracias a su potente herramienta de informes. Dentro de su alta gama de servicios aparecen estas características que lo hacen excelente:

- ✓ Funciones completas de layout e interactividad

Presentación de dashboards, tablas, tablas cruzadas, gráficos y widgets. Diseño perfecto con salida continua o por páginas para su impresión o visualización en web. Los subinformes y elementos de tabla interactivos proporcionan layouts complejos y altamente interactivos.

- ✓ Arquitectura escalable

Los reguladores de consultas controlan el uso de los recursos del sistema. Los virtualizadores de informes optimizan el consumo de la memoria. Informes de tamaño ilimitado. (10)

Su principal inconveniente viene dado porque solo trabaja con código Java, su uso e interacción con otros lenguajes es muy engorroso y solo es posible a través de protocolos y librerías como el PHPJavaBridge, para su empleo con en el lenguaje php.

- DynamicReports:

Se basa en JasperReports. Permite crear diseños de informes dinámicos y no necesita de un diseñador de informes visual. Puede crear rápidamente informes y preparar los documentos que se pueden visualizar, imprimir o exportar en muchos formatos populares, tales como PDF, Excel, Word y otros. Un diseño es creado por un código simple de Java puro.

A pesar de que genera los reportes en tiempo de ejecución, se puede implementar cualquier lógica en su código que decide cómo el informe se verá, así a diferencia de los informes estáticos (plantillas de JasperReports.jrxml) donde el diseño definido no puede cambiarse en tiempo de ejecución.

Su uso es viable, lo que no sólo ahorra una gran cantidad de su tiempo y dinero, sino que también aumenta la productividad de informes y disminuye el tiempo de formación. Sólo se necesita una pequeña cantidad de código para crear un informe simple e incluso algunos más complejos, será claro, fácil de mantener y comprensible. (11)

- Reportico:

Es una herramienta completa de informes web de código abierto escrito puramente en PHP. Reportico ofrece una pantalla de front-end basado en web para el diseño y la visualización de informes

almacenados en formato XML. Reportico admite la selección de criterios flexibles y podrán presentarse en formato HTML, PDF, CSV, XML y formatos JSON.

Algunas de sus características son:

- Criterios de selección de usuarios con cuadros de lista, casillas de verificación, botones de radio.
- Garantiza el etiquetado y el formato de los datos de informe.
- Contiene aspectos configurables a través de hojas de estilo y plantillas.
- Integra los informes dentro de una página web con sólo unas pocas líneas de PHP
- Produce barras, barras apiladas y gráficos de líneas.
- Imágenes de la base de datos dentro de la salida del informe. (12)

Nacionales:

- **PATDSI:**

El Paquete de Ayuda a la Toma de Decisiones y Soluciones Integrales cuenta con un grupo de herramientas con el fin de propiciar ayuda a la toma de decisiones de las empresas, ideado como una plataforma web para la inteligencia de negocio. Desarrollado por el departamento de Soluciones Integrales perteneciente al Centro de Tecnologías de Gestión de Datos de la Universidad de las Ciencias Informáticas.

Este paquete incluye un generador de reportes dinámicos; el mismo elabora los reportes en tiempo de diseño por estar orientado al programador y no al usuario final; por lo cual no permite la modificación dinámica del diseño del reporte. Además de que no todos los usuarios son capaces de diseñar reportes de manera rápida y con la calidad requerida, debido al nivel de complejidad que la aplicación puede presentar para la creación de un reporte. (13)

- **Olympia**

Olympia es una aplicación desarrollada sobre el framework Symfony y escrita en PHP. Es multiplataforma. Soporta imágenes, gráficas y sub-reportes, así como varios orígenes de datos. Proporciona a los usuarios, entre otras opciones, agilizar la toma de decisiones, generar reportes en varios formatos y con gran variedad de opciones en su diseño, marcando una diferencia entre los reportes tradicionales y los reportes dinámicos, objetos de este producto. Está compuesto por varias aplicaciones entre las que se encuentran el Visor de reportes, Diseñador de modelos y el Diseñador de reporte. Aunque permite abstraerse en parte de los conocimientos relacionados con los gestores de

bases de datos, el usuario aún debe poseer conocimientos básicos. Además, se entorno de trabajo está estructurado de forma que es difícil guiarse en el creación y generación de reportes. (14)

- Grato Report

Una herramienta desarrollada en el año 2008 por el proyecto alasGRATO de la Facultad 6, en la Universidad de las Ciencias Informáticas, para la generación de reportes referentes a la Química. Fue desarrollado sobre el lenguaje JAVA usando la biblioteca para la generación de reportes Dynamic JasperReports como base para su funcionamiento (14)

- SIGICEM:

El Sistema de Gestión para Ingeniería Clínica y Electromedicina (SIGICEM) constituye una solución eficaz la cual contribuye a alcanzar niveles de eficiencia en los procesos de gestión tecnológica sobre el equipamiento médico. Mediante los conceptos exigidos internacionalmente como: organización, planificación, control y análisis para la administración de la información.

El mismo posee un módulo para la generación de reportes permitiendo ampliar la variedad de la información, agilizar el proceso de análisis y toma de decisión ante eventos con el equipamiento médico y mayor asequibilidad para el proceso de monitoreo de los datos.

Al término de la investigación sobre las herramientas de generación de reportes en el ámbito internacional y nacional, se ha llegado a la conclusión que el uso de las mismas no es el más idóneo para la aplicación. Puesto a que las mismas crean dependencias innecesarias e incompatibilidad en los servidores desplegados y no se cuenta en las áreas de salud con los elementos esenciales para el mínimo funcionamiento de las tareas a cumplir.

Por lo tanto, mediante el estudio realizado, se hizo evidente la necesidad de contar con otro medio que se ajustara al propósito general, decidiendo la inclusión de una librería que no presentara conflictos en su práctica.

1.3 Metodología de desarrollo de software

Proceso Unificado de Software (RUP)

El Proceso Unificado es un proceso de software genérico que puede ser utilizado para una gran cantidad de tipos de sistemas de software, diferentes áreas de aplicación, diferentes tipos de

organizaciones, diferentes niveles de competencia y diferentes tamaños de proyectos. Se basa en componentes, lo que significa que el sistema en construcción está hecho de componentes de software interconectados por medio de interfaces bien definidas.

El Proceso Unificado usa el Lenguaje de Modelado Unificado (UML) en la preparación de todos los planos del sistema. Sus aspectos distintivos están capturados en tres conceptos claves: dirigido por casos de uso, centrado en la arquitectura e iterativo incremental.

En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean el diseño usando la arquitectura como guía, implementan el diseño en componentes y verifican que los componentes satisfacen los casos de uso. La arquitectura provee la estructura sobre la cual guiar el trabajo en iteraciones, mientras que los casos de uso definen las metas y dirigen el trabajo en cada iteración. Remover cualquiera de estos conceptos reducirá severamente el valor del Proceso Unificado. (15)

Lenguaje Unificado de Modelado 2.1

Unified Modeling Language o UML, por sus siglas en inglés, fue desarrollado por Grady Booch y Jim Rumbaugh en 1995, con el objetivo de crear un lenguaje para el modelado. Mediante la modelación se logran hacer más entendibles distintos puntos de la realidad que sean de interés, todo esto a partir de la complementación de varias técnicas de modelado en una sola. Integrado por un conjunto de diagramas, los cuales se encuentran agrupados en dos categorías fundamentales: diagramas de estructura y diagramas de comportamiento.

Este lenguaje de modelado permite crear un nivel de comprensión y entendimiento entre los analistas, desarrolladores o cualquier personal involucrado, haciendo más soluble la comunicación existente entre ellos; permitiendo visualizar, especificar, construir y documentar la información con la que se cuenta. (16)

1.4 Herramientas y tecnologías de desarrollo utilizadas

A continuación se presentan las herramientas y tecnologías seleccionadas para la realización del sistema, teniendo en cuenta las utilizadas en el departamento.

Herramienta CASE

Conjunto de aplicaciones y metodologías de desarrollo que ayudan a los desarrolladores y analistas durante todo el proceso de desarrollo de software y aceleran el desarrollo de los sistemas y aumentan la calidad del software a través de la unión de las herramientas de software y las metodologías de desarrollo de software formales.

- Visual Paradigm for UML 6.4

Visual Paradigm es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

La versión Visual Paradigm UML 6.4, fue diseñada para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque orientado a objetos. Permitiendo mejoras de disponibilidad, diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad y el uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.

Entorno integrado de desarrollo

Un Entorno Integrado de Desarrollo o IDE (por sus siglas en inglés) es un entorno de programación que ha sido empaquetado como un programa de aplicación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los mismos pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Los IDE ofrecen un marco de trabajo amigable para la mayoría de los lenguajes de programación, donde pueden funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto. Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación mediante los plugins que le añaden soporte de lenguajes adicionales. (17)

- Netbeans 7

Para realizar el desarrollo del módulo se utilizará como IDE la herramienta Netbeans en su versión 7, la cual permite realizar el trabajo bajo las últimas tenencias de PHP, integrando gran variedad de herramientas modernas para mejorar la extensibilidad con el apoyo de los principales marcos de PHP.

Cuenta con una enorme cantidad de extensiones y plugins que pueden incorporarse al sistema para facilitar las tareas de programación, también posee soporte para el framework de Symfony a utilizar en la aplicación, también cuenta con experiencia de las versiones anteriores en la legibilidad y completamiento de sentencias de código. Se destaca su intuitiva interfaz de usuario, su sistema de depuración de programas y las demás herramientas de ayuda, que facilitarán el desarrollo del módulo, la cual mediante su dinámica interacción permite al usuario mismo instruirse sobre las nuevas ventajas y funcionalidades de la herramienta.

Frameworks

El framework es una estructura de soporte definido para resolver los problemas básicos que presenta el software o las especificidades de los módulos. Trabaja utilizando una llamada automatización del código fuente, la cual exige al desarrollador la creación de un código simple, es el esqueleto sobre el cual varios objetos son integrados para facilitar una solución dada.

-Symfony 1.4

Es un framework completo que permite la optimización del patrón Modelo Vista Controlador, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony puede ser completamente personalizado para cumplir con los requisitos de las empresas que disponen de sus propias políticas y reglas para la gestión de proyectos y la programación de aplicaciones. Por defecto incorpora varios entornos de desarrollo diferentes e incluye varias herramientas que permiten automatizar las tareas más comunes de la ingeniería del software. (18)

-ExtJs 2.2

Es una biblioteca de JavaScript del lado del cliente para el desarrollo de aplicaciones web interactivas usando tecnologías como AJAX, DHTML y DOM, dispone de un conjunto de componentes para incluir en las aplicaciones web, las cuales están capacitados para comunicarse con el servidor usando AJAX.

También contiene numerosas funcionalidades que permiten añadir interactividad a las páginas que utiliza.

ExtJs también trae un rico paquete de datos que permite a los desarrolladores utilizar un controlador de modelo-vista (MVC) en la construcción de su aplicación. El MVC aprovecha características como Big Data Grids permiten un nuevo nivel de interactividad en las aplicaciones web. (19)

Permite la existencia de un balance entre Cliente – Servidor, donde la carga de procesamiento se distribuye, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo. También aporta una comunicación asíncrona, en este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se dé cuenta. (20)

Plataforma de desarrollo LAMP

LAMP está considerada como una de las mejores herramientas disponibles para que cualquier organización o individuo pueda emplear un servidor web versátil y potente. Aunque creados por separado, cada una de las tecnologías que lo forman disponen de una serie de características comunes (Linux, Apache, MySQL y uno de los siguientes lenguajes: Perl, Python o PHP).

Especialmente interesante es el hecho que estos cuatro productos pueden funcionar en una amplia gama de hardware, con requerimientos relativamente pequeños sin perder estabilidad. Esto ha convertido a LAMP en la alternativa más adecuada para pequeñas y medianas empresas. (21)

La plataforma está integrada por:

- Linux con su distribución Ubuntu 12.04 como Sistema Operativo.
- Apache como servidor web en su versión 2.2.
- MySQL como gestor de base de datos en su versión 5.1.
- PHP como lenguaje de programación en su versión 5.1.

Librería JQueryPlot

JQuery es una biblioteca gratuita de JavaScript, cuyo objetivo principal es simplificar las tareas de creación de páginas web, la cual funciona en todos los navegadores modernos. JQuery ayuda a la

concentración de gran manera en el diseño de un sitio, al abstraer por completo todas las características específicas de cada uno de los navegadores. Otra de las grandes ventajas de jQuery es que se enfoca en simplificar los scripts y en acceder/modificar el contenido de una página web, esta librería agrega una cantidad impresionante de efectos nuevos a JavaScript. (22)

Beneficios del uso de jQuery:

- jQuery utiliza sintaxis muy parecida a CSS.
- Funciona con series de elementos.
- Permite manipular series de elementos y modificarlas con una simple línea de código.
- Ayuda a concentrarse en el resultado final.
- jQuery es sencillo de expandir, pues cuenta con gran cantidad de plugins que se pueden utilizar o hasta crear uno propio.
- Compatible con todos los navegadores modernos.

JqPlot es un graficador y el plugin utilizado por el framework jQuery JavaScript. JqPlot produce líneas, barras y gráficos circulares con características tales como:

- Numerosas opciones de estilo gráfico.
- Texto de los ejes girado.
- Información resaltada de datos especificados.
- Parámetros por defecto para facilitar su uso.

En el desarrollo del capítulo, el análisis de las deficiencias presentadas en la generación de reportes y los nuevos requerimientos identificados, evidencia la necesidad de llevar a cabo el desarrollo de la segunda versión del módulo Reportes Estadísticos para el Sistema de Gestión para Ingeniería Clínica y Electromedicina.

La realización de un estudio previo sobre las herramientas y sistemas utilizados para la generación de reportes estadísticos, a nivel internacional y nacional, posibilitó obtener una guía para el posterior desarrollo de la solución. El estudio y asimilación de las herramientas y tecnologías definidas por el departamento, permitió obtener una mejor preparación para el desarrollo de la investigación en curso.

Capítulo II: Características del sistema

En el presente capítulo se exponen las particularidades principales del proceso de gestión de reportes a tener en cuenta en el desarrollo de la solución propuesta. Los elementos fundamentales a ser descritos son: el modelo de dominio definido, las especificaciones de los requisitos funcionales y no funcionales reconocidos, a través de los cuales se representan los casos de uso del sistema y sus descripciones.

2.1 Modelo de dominio

El modelo de dominio es un artefacto utilizado por el analista como un medio para comprender las actividades que son objeto de automatización en el sistema, construido con las reglas UML durante la fase de concepción. Los objetos, conceptos o clases conceptuales incluidos en este modelo describen entidades que están vinculadas al problema en cuestión, su objetivo principal es capturar todo lo necesario para comprender cómo va a funcionar la aplicación que se ha diseñado.

2.2 Conceptos Fundamentales del Dominio

Para brindar una mejor comprensión del diagrama del Modelo de Dominio se realiza una breve descripción de los conceptos encontrados en el problema.

Especialista: Es el encargado del manejo de datos y genera los reportes referentes a la información dentro del área a la que pertenece, cuenta con ciertos privilegios en la aplicación, posee una especialidad principal y puede dominar otras.

Reporte: Informes que organizan y exhiben la información contenida en una base de datos. Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios. (6)

Reporte estadístico: Maneja los informes que organizan y exhiben la información contenida en el sistema. Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios. El reporte de esta forma, confiere una mayor utilidad a los datos.

Reporte estático: Maneja los datos estadísticos de relevancia para el sistema, presentando informes con cifras específicas y precisas para la gestión de la información.

Reporte personalizado: Informe que reúne la información acerca de los componentes esenciales del sistema SIGICEM: equipos médicos, especialistas, piezas, mantenimiento, orden de servicio. Manejando datos precisos que permiten la gestión de cada una de estas áreas de forma personalizada, para que el usuario que interactúe con ellos pueda seleccionar su criterio de búsqueda.

Reporte gráfico: Representación en forma de gráfico que describe la disponibilidad técnica y los estados técnicos de equipos según un criterio determinado seleccionado por el usuario.

Equipo médico: Aparato, implemento o máquina capaz de realizar un diagnóstico, prevenir, controlar o aliviar una dolencia o enfermedad determinada. El equipo puede pasar por diferentes estados atendiendo a sus prestaciones de funcionamiento.

Mantenimiento: Maneja la información acerca de los equipos médicos, sus especialistas y el grupo de trabajo que labora con ellos, también lleva la constancia de la frecuencia con la que se debe realizar el mismo.

Fallas: Defecto no deseado que trae como consecuencia que el equipo médico funcione a capacidad parcial o detenga su funcionamiento. En el sistema están establecidos los tipos de fallas más comunes y la descripción asociada a ella.

Piezas: Contiene la información sobre las piezas utilizadas por los equipos, llevando un control de su marca, modelo y precio, importante para el mejoramiento del estado de los equipos médicos.

Orden de servicio: Según el estado técnico que presente el equipo médico se realizará un tipo de trabajo que puede ser: instalación, mantenimiento, reparación, asistencia. Cada orden cuenta con una fecha de inicio y una fecha de fin. Al concluir el servicio la misma presenta un estado final en dependencia del éxito que haya tenido su ejecución.

Baja técnica: Estado técnico que presentan los equipos médicos que sus condiciones se han deteriorado de tal forma que ya su utilización es imposible y se cancela su vida técnica.

Defectuoso: Estado técnico que presentan los equipos médicos, donde alguna de sus potencialidades principales pierden al menos un 10% de su rendimiento óptimo.

Funcionando: Estado técnico que presentan los equipos médicos cuando se encuentran al 100% de su capacidad ideal de trabajo.

Pendiente a baja: Estado técnico que presentan los equipos médicos cuando es muy difícil que su rotura pueda ser solucionada en un tiempo que garantice que su posterior uso comience con la calidad técnica necesaria.

Roto: Estado técnico que presentan los equipos médicos cuando sufren una rotura parcial o total que impide su funcionamiento y deja de ser utilizado para ser reparado.

Por instalar: Estado técnico que presentan los equipos médicos cuando están en espera de su instalación en las unidades de salud, posteriormente pasan a la condición técnica: Equipo Funcionando.

Especialidad principal: Es la ocupación principal que rige el trabajo del especialista en el sistema.

Especialidades que dominan: Compendio de materias que el especialista debe dominar para la realización de su trabajo.

Diagrama del modelo de dominio

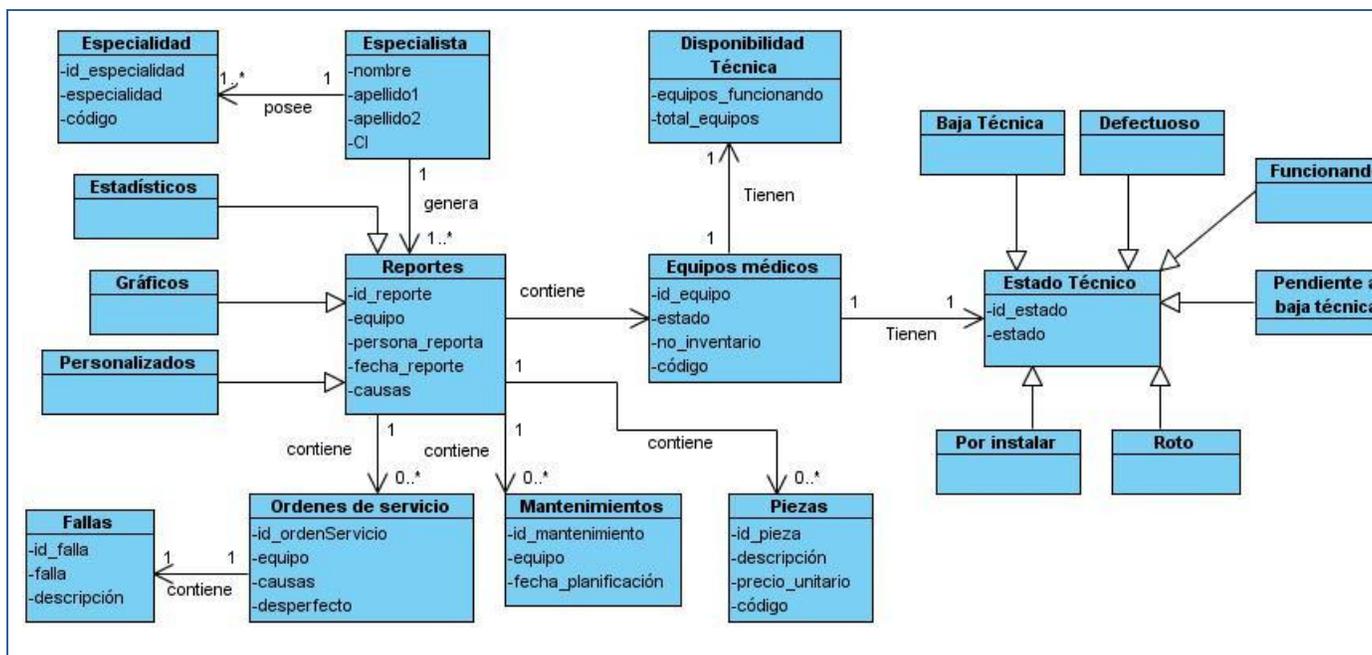


Figura 1: Diagrama del modelo de dominio.

2.3 Propuesta del sistema

La aplicación que se propone tiene como principal objetivo la generación de reportes gráficos capaces de mostrar y almacenar en tiempo real los datos generales de la ingeniería clínica y electromédica para su procesamiento. De igual manera establecerá un control en el manejo de la información referente a los equipos médicos, especialistas, piezas y mantenimiento, permitiendo la interpretación de las mismas por parte de la dirección de vigilancia tecnológica del MINSAP. Esta contará con una interfaz intuitiva, que brinde confianza e integridad en los datos mostrados y almacenados.

Establecerá un conjunto de beneficios para los usuarios, entre los cuales está la interacción, visualización y almacenamiento de la información asociados al sistema en tiempo real; brindando a los especialistas un mayor aprovechamiento de los datos, mediante gráficas que describan el comportamiento del estado o disponibilidad técnica del equipamiento. Para una mejor comprensión de lo antes expuesto se relacionan en los anexos una serie de imágenes que representan las principales interfaces gráficas de usuario del sistema.

2.4 Especificación de los requisitos de software

Teniendo en cuenta los conceptos relacionados en el modelo de dominio se determinaron las funcionalidades que el sistema debe desarrollar para dar solución al problema planteado. Las mismas conforman los requisitos de software, que son las condiciones a cumplir por el cliente, por un contrato o un documento oficial.

Los requisitos se encuentran divididos en dos grupos: los funcionales que representan la funcionalidad del sistema y los no funcionales que son aquellos atributos con los que debe contar, pero que no son una función concreta.

Requisitos Funcionales:

Describen la funcionalidad o los servicios que se espera que el sistema provea sus entradas, salidas y excepciones.

	Requisitos funcionales
RF1	Generar reporte personalizado de equipos médicos.
RF2	Generar reporte personalizado de piezas.
RF3	Generar reporte personalizado de especialistas.
RF4	Generar reporte personalizado de mantenimiento
RF5	Generar reporte personalizado de órdenes de servicio
RF6	Generar reporte gráfico estado técnico y disponibilidad técnica por unidad de subordinación nacional.
RF7	Generar reporte gráfico estado técnico y disponibilidad técnica por unidad de subordinación provincial.
RF8	Generar reporte gráfico estado técnico y disponibilidad técnica por unidad de

	subordinación municipal.
RF9	Generar reporte gráfico estado técnico y disponibilidad técnica por unidad de salud.
RF10	Generar reporte gráfico estado técnico y disponibilidad técnica por área de salud.
RF11	Generar reporte gráfico estado técnico y disponibilidad técnica por departamento.
RF12	Generar reporte gráfico estado técnico y disponibilidad técnica por local.
RF13	Generar reporte gráfico estado técnico y disponibilidad técnica por especialidad.
RF14	Generar reporte gráfico cantidad total de equipos en el sistema.
RF15	Generar reporte gráfico disponibilidad técnica por provincia.
RF16	Generar reporte gráfico disponibilidad técnica por municipio.
RF17	Generar informe mensual.
RF18	Generar listado de especialistas por horas de servicio.
RF19	Generar listado de especialistas por mantenimiento.

Tabla 1: *Requisitos funcionales.*

Requisitos No Funcionales:

Usabilidad: La aplicación debe permitir un conjunto de facilidades de uso para los usuarios finales, colaborando en su comprensión y aceptación del producto.

Fiabilidad: Es imprescindible ofrecer consistencia y confianza en la información que se le brindará al usuario a partir de su veracidad, desde el comienzo de su entrada al sistema hasta el fin de uso, con el objetivo de evitar errores en el mismo. Permitiendo a los usuarios noveles comodidad y eficiencia en el trabajo. Para evitar la ocurrencia de información errónea y para preservar la seguridad de la aplicación, el trabajo en la misma se realizará a través de campos a seleccionar.

Disponibilidad: La aplicación estará en servicio a tiempo completo, permitiendo a los usuarios gestionar o realizar los reportes cuando estimen conveniente.

Eficiencia: La aplicación debe contar ante la petición de información con una respuesta rápida para los usuarios a la hora de realizar tanto las actualizaciones como en el momento de consultar la información, con el fin de lograr un buen resultado debe tener un buen procesamiento de los datos pues los usuarios estarán en constante interacción con la aplicación.

Soporte: Con el inicio de la utilización de la aplicación se almacenará toda la información referente a las deficiencias y no conformidades existentes para incorporar las mejoras al sistema.

Interfaz de Usuario: El sistema será diseñado con una interfaz fácil de usar, similar a la del sistema Operativo Windows para garantizar la familiarización con los usuarios que interactúen con él. Para la selección de colores y criterios se considerará las pautas definidas en el manual de entidad corporativa del CICEM, asimilando sustancialmente las definidas por el departamento SES.

Documentación de usuarios en línea y Ayuda del Sistema: El sistema poseerá una ayuda, en la que se explica de forma detallada el uso de las funcionalidades, lo que permite el buen desempeño de los usuarios.

Requisitos de Licencia: El sistema será desarrollado en base a las políticas del software libre, que fueron ajustadas al Sistema Nacional de Salud.

Portabilidad: El sistema será multiplataforma (Linux-Windows), habilitando su ejecución sobre diferentes sistemas operativos sin importar sus versiones.

Seguridad: El sistema contará con una política de seguridad sustentada a la restricción de usuarios en dependencia del nivel jerárquico que cumpla su rol dentro del mismo.

Políticos-culturales: Se deberá hacer un uso correcto del idioma español en la interfaz de la aplicación, con logotipos e imágenes que se encuentren en correspondencia con el carácter de la misma.

Software:

Cliente:	Servidor:
<ul style="list-style-type: none"> • Mozilla Firefox 3.6 o superior 	<ul style="list-style-type: none"> • Symfony 1.4 • ExtJS 2.2 • Plataforma LAMP

Tabla 2: Requisito No Funcional Software

Hardware:

Requerimientos mínimos para el servidor:	Requerimientos mínimos el cliente:
<ul style="list-style-type: none"> • Microprocesador superior a 2.0 GHz • 512 MB RAM o superior • 40 GB de Disco Duro como mínimo 	<ul style="list-style-type: none"> • Microprocesador superior a 1.0 GHz • 128 MB RAM o superior • MODEM o red con TCP-IP para conexión al servidor

Tabla 3: Requisito No Funcional Hardware

2.5 Modelo de casos de uso del sistema

Los componentes primarios de un modelo de casos de uso son: los casos de uso, los actores y sus relaciones. Los actores son entidades externas que se modelan y que interactúan con el sistema. Los casos de uso van representar los requisitos o capacidades internas del sistema, y las relaciones son las encargadas de expresar la interacción entre caso de uso y actor o entre actores o entre casos de uso. (23)

El actor identificado en el sistema es:

Actor:	Descripción:
Especialista	Usuario encargado del manejo de datos y de generar los reportes referentes a la información dentro del área a la que pertenece, cuenta con ciertos privilegios en la aplicación, posee una especialidad principal y puede

	dominar otras.
--	----------------

Tabla 4: Actores del sistema

Diagrama de casos de uso del sistema

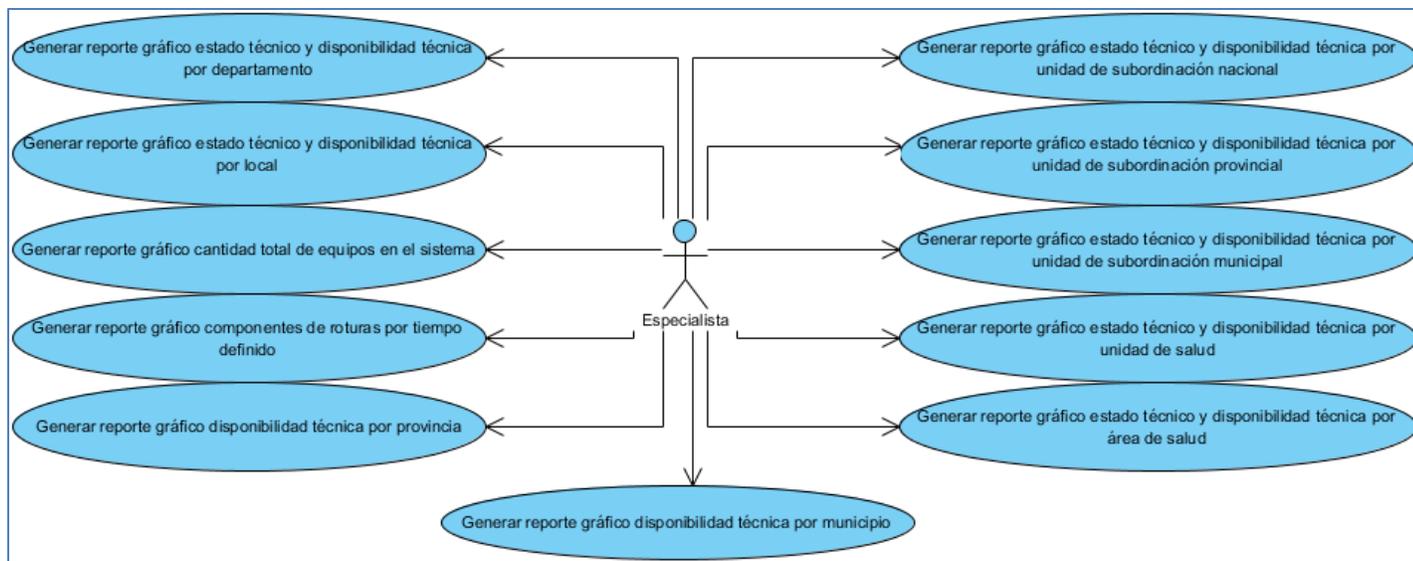


Figura 2: Paquete reportes gráficos

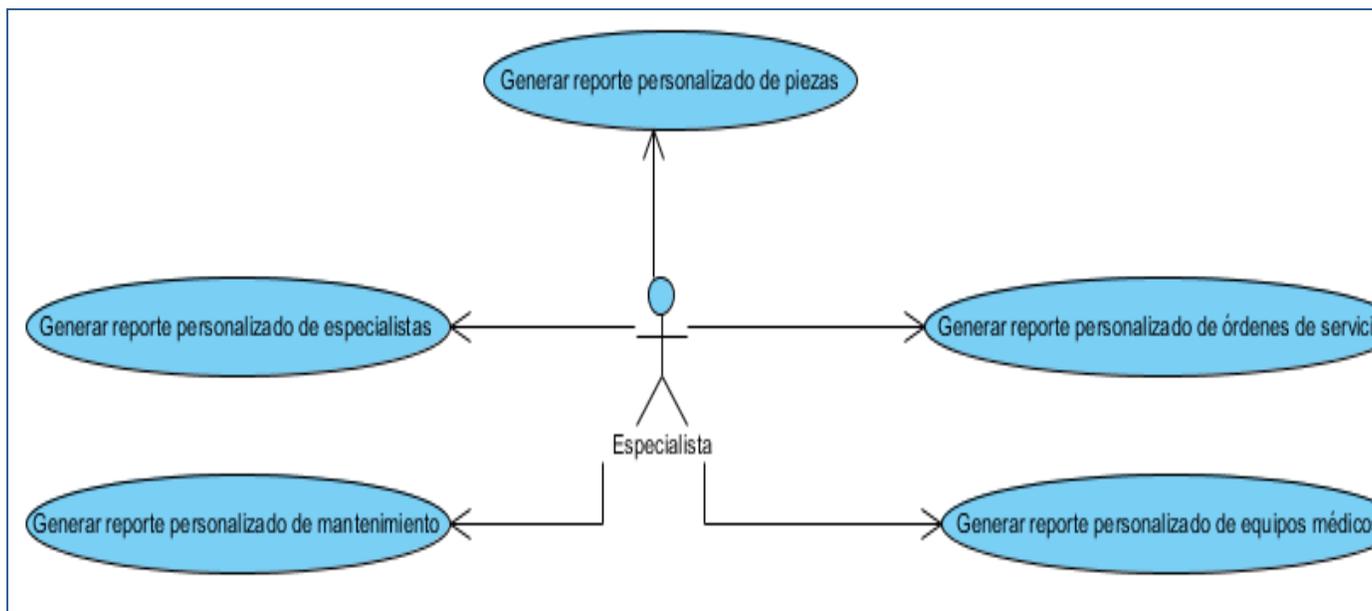


Figura 3: Paquete reportes generales

2.6 Descripción textual de los casos de uso

CU 1. Generar reporte personalizado de equipos médicos

Objetivo	Conocer información relacionada con los equipos médicos según los criterios de búsqueda seleccionados.
Actores	Especialista.
Resumen	Inicia cuando el actor accede a la opción Reportes Estadísticos en el menú inicio y presiona el ícono que indica la acción Reporte personalizado de equipos médicos, el sistema brinda la posibilidad de introducir los criterios para generar el reporte, el actor introduce dichos criterios, el sistema a partir de los criterios seleccionados y consultando las entidades correspondientes, genera el reporte de información. El actor puede imprimir el reporte, consultarlo o guardarlo según sus necesidades, el caso de uso termina.
Complejidad	Media.
Prioridad	Crítico.
Precondiciones	Debe existir información en el sistema correspondiente a los equipos médicos.
Postcondiciones	Realizado el reporte personalizado de equipos médicos se generó un informe con la documentación seleccionada por el actor.

Tabla 5: Descripción textual del caso de uso: Generar reporte personalizado de equipos médicos

CU 2. Generar reporte gráfico estado técnico y disponibilidad técnica por unidad de subordinación nacional

Objetivo	Conocer los datos relacionados con el estado técnico y disponibilidad técnica en las instituciones de subordinación nacional.
Actores	Especialista.

Resumen	Inicia cuando el actor accede a la opción Reportes Estadísticos en el menú inicio y presiona el ícono que indica la acción Reporte gráfico estado técnico y disponibilidad técnica por subordinación, el sistema brinda la posibilidad de escoger los criterios para generar el reporte, el actor toma su decisión, el sistema a partir de los criterios seleccionados y consultando las entidades correspondientes, genera el reporte gráfico. El actor puede consultarlo o guardarlo según sus necesidades, el caso de uso termina
Complejidad	Alta.
Prioridad	Crítico.
Precondiciones	Debe existir información en el sistema correspondiente a las instituciones de subordinación nacional.
Postcondiciones	

Tabla 6:

Descripción textual del caso de uso: Generar reporte gráfico estado técnico y disponibilidad técnica por unidad de subordinación nacional

CU 3. Generar reporte gráfico disponibilidad técnica por provincia

Objetivo	Conocer los datos relacionados con la disponibilidad técnica en las provincias
Actores	Especialista.
Resumen	Inicia cuando el actor accede a la opción Reportes Estadísticos en el menú inicio y presiona el ícono que indica la acción Reporte gráfico disponibilidad técnica por provincia, el cual brinda la posibilidad de escoger los criterios para generar el reporte, el actor toma su decisión, el sistema a partir de los criterios seleccionados y consultando las entidades correspondientes, genera el reporte gráfico. El actor puede consultarlo o guardarlo según sus necesidades, el caso de uso termina

	guardarlo según sus necesidades, el caso de uso termina
Complejidad	Alta.
Prioridad	Crítico.
Precondiciones	Debe existir información en el sistema correspondiente a los equipos
Postcondiciones	

Tabla 7: Descripción textual del caso de uso: Generar reporte gráfico disponibilidad técnica por provincia

Al finalizar el capítulo quedaron descritas algunas de las funcionalidades que presentará el módulo Reportes Estadísticos v2.0. Se describen brevemente los conceptos primordiales que evidencian el entendimiento del problema. Producto del análisis se define un modelo de dominio donde se expone la relación entre los criterios encontrados. Se definen el conjunto de requisitos funcionales y no funcionales implícitos en el sistema, así como un diagrama de casos de uso partiendo de los requerimientos funcionales identificados.

Capítulo III: Análisis y diseño del sistema

En el capítulo se muestra una descripción detallada del análisis y el modelo del diseño. Mostrándose el principal patrón de diseño que se pone en práctica y los diagramas de clases y de secuencia del modelo de diseño, el modelo de datos con la descripción de las tablas que lo componen y de despliegue del sistema.

3.1 Modelo de análisis

Ofrece una especificación precisa de los requisitos y estructura de los mismos, facilitando su comprensión, preparación, modificación y su mantenimiento en general. El modelo se describe utilizando el lenguaje de los desarrolladores, lo cual permite introducir un mayor formalismo y ser utilizado para razonar sobre los funcionamientos internos del sistema. (24)

Una vez identificados los requisitos que debe cumplir el software a desarrollar, es importante realizarles un análisis con la finalidad de comprenderlos mejor antes de introducirse en su diseño, garantizando así una arquitectura robusta, eficiente y capaz de resistir a cambios. La disciplina del análisis tiene como propósito conseguir una comprensión más precisa de los requisitos, refinarlos y estructurarlos utilizando el lenguaje de los desarrolladores para analizar con profundidad los requisitos funcionales y proporcionar una visión general del sistema.

También se comienza a observar el software de forma interna, en función de clases y paquetes que permiten el cumplimiento de todos los requisitos que deben ser cumplidos. El modelo de análisis puede considerarse como una primera aproximación al modelo de diseño, pues representa una entrada fundamental cuando se da forma al sistema en el diseño y en la implementación.

Durante su desarrollo el modelo presenta un conjunto de tres estereotipos, los cuales intervienen en la realización de las clases de análisis para su identificación:

- Entidad: Modela los datos que poseen larga vida y que son a menudo persistente, generalmente poseen información específica de una entidad determinada.
- Control: Coordina la realización de uno o varios casos de uso, coordinando las acciones de los objetos que implementan la funcionalidad del caso de uso.

- Interfaz: Representa el acceso del usuario al sistema. Atiende todas las selecciones y peticiones que pueda realizar un determinado usuario.

Ventajas del modelo de análisis:

- Suaviza la transición al diseño.
- Apoya el cambio a otra plataforma de programación.
- Sirve para tener una visión general de la propuesta del sistema.
- Sirve para planificar y dividir el diseño e implementación en pequeños módulos, lo cual facilita el trabajo con diferentes equipos de desarrolladores.
- Apoya la aplicación de reingeniería a aplicaciones existentes, ya que el lenguaje es menos técnico y se puede entender mucho más fácil la propuesta de solución. (25)

Diagrama de clases de análisis

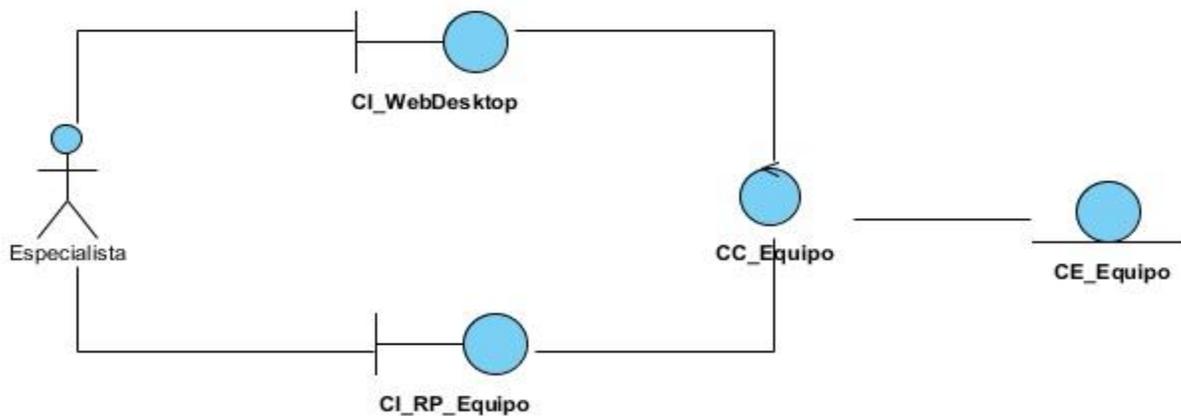


Figura 4: Diagrama de Clases de Análisis: Generar reporte personalizado de equipos médicos.

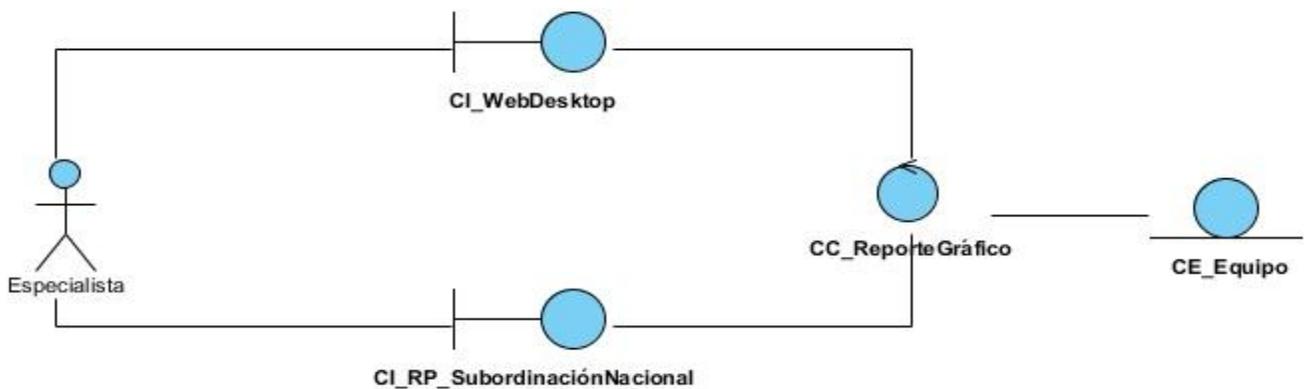


Figura 5: Diagrama de Clases de Análisis: Generar reporte gráfico estado técnico y disponibilidad técnica por unidad de subordinación nacional

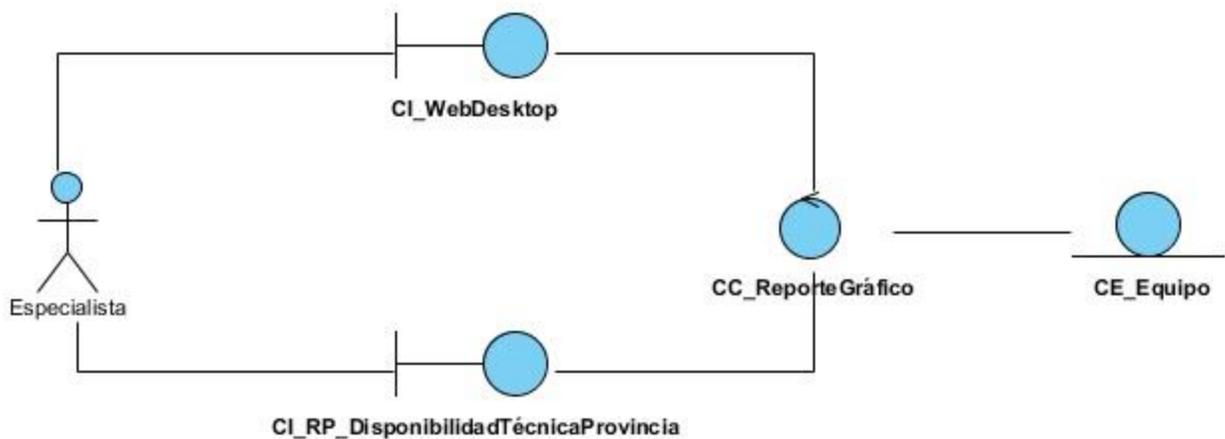


Figura 6: Diagrama de Clases de Análisis: Generar reporte gráfico disponibilidad técnica por provincia

3.2 Diagramas de interacción. Comunicación

Los diagrama de comunicación, muestran la interacción que existe entre los roles. Las relaciones expresan los objetos actuales y cómo ellos se relacionan unos con otros. Así como los diagramas de secuencia, los diagramas de colaboración pueden ser utilizados para ilustrar la ejecución de una operación, una ejecución de un caso de uso o un escenario de interacción dentro del sistema. Las

relaciones se dibujan con líneas y se puede agregar una etiqueta para un mensaje y un número que define la secuencia de estas. (26)

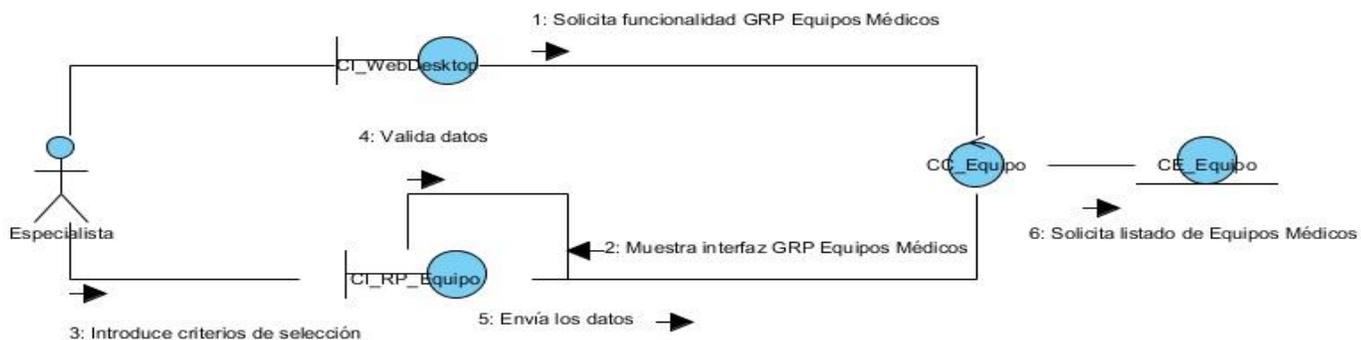


Figura 7: Diagrama de Comunicación: Generar reporte personalizado de equipos médicos.

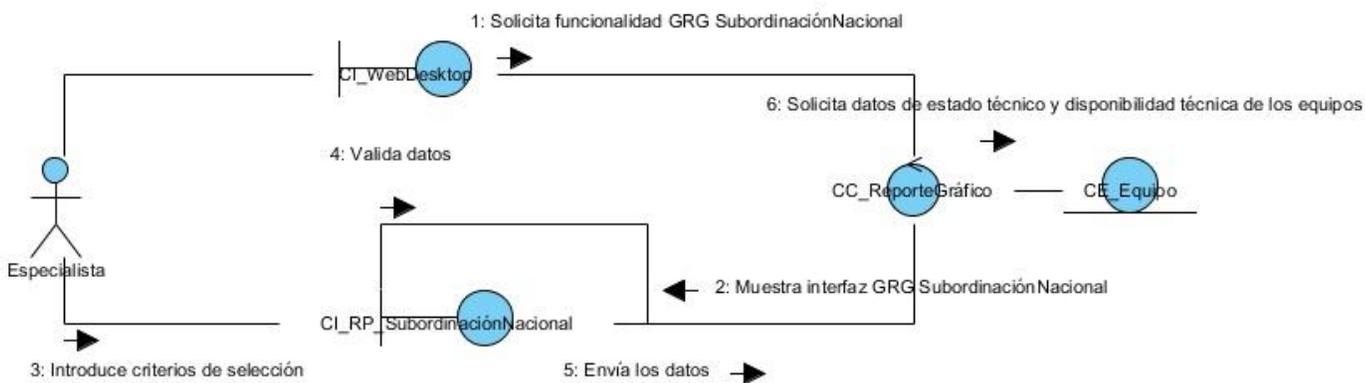


Figura 8: Diagrama de Comunicación: Generar reporte gráfico estado técnico y disponibilidad técnica por unidad de subordinación nacional

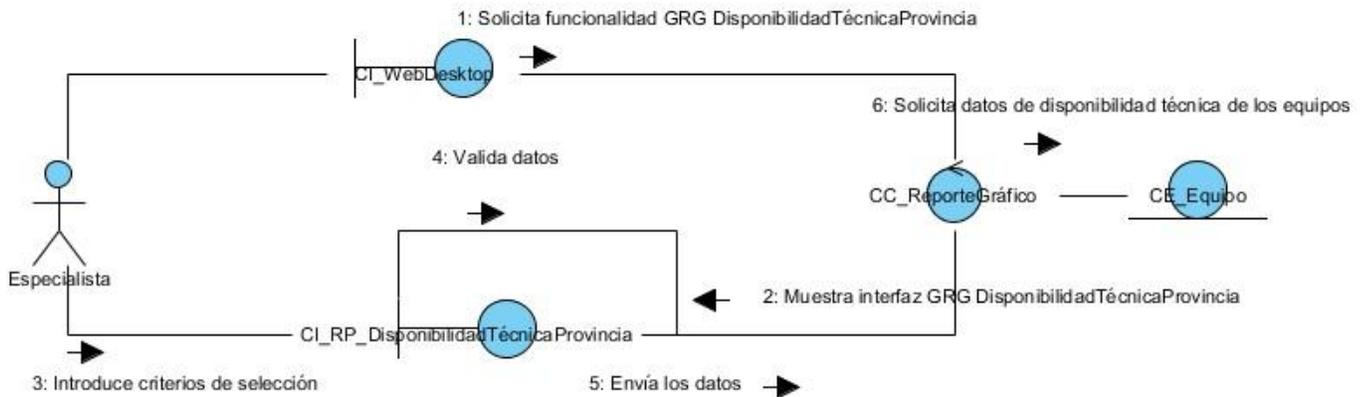


Figura 9: Diagrama de Comunicación Generar reporte gráfico disponibilidad técnica por provincia

3.3 Descripción de la arquitectura

La arquitectura de software es el diseño de más alto nivel de un sistema, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco de referencia necesario para guiar la construcción de un sistema. Permitiendo a los programadores, analistas y todo el conjunto de desarrolladores, compartir una misma línea de trabajo cubriendo todos los objetivos y restricciones de la aplicación. (27)

En vista a la metodología y tecnologías expuestas, se define como parte de la estrategia base de la arquitectura, la implementación del patrón de diseño Modelo Vista Controlador (MVC). Este permite realizar la programación multicapa, separando los datos de la aplicación, la interfaz de usuario y la lógica de control, en tres componentes distintos.

Modelo Vista Controlador (MVC): Es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

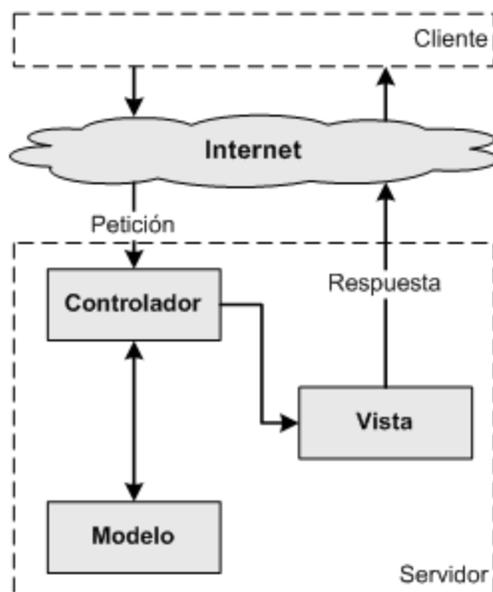


Figura 10: Estructura Modelo Vista Controlador

La separación entre lógica de negocio y presentación, permite que se le puedan aplicar al sistema opciones como el multilinguaje, distintos diseños de presentación, sin alterar la lógica de negocio. La separación de capas como presentación, lógica de negocio y acceso a datos es fundamental para el desarrollo de arquitecturas consistentes, reutilizables y mantenibles, lo que al final resulta un ahorro de tiempo en el desarrollo.

- Modelo: Contiene la lógica de negocio, la base de datos se encuentra almacenada en esta capa, en la que se engloban los datos y las funcionalidades. Se comporta de manera independiente a cualquier representación de salida y comportamiento de entrada, lo que asegura la integridad de los datos y permite derivar nuevos.
- Vista: Es la parte que utilizan los usuarios para interactuar con la aplicación, simplemente la página HTML que se muestra al usuario. Esta capa se centra en presentar al modelo en un formato adecuado para el intercambio con el usuario final. Los gestores de plantillas se incluyen dentro de esta.
- Controlador: Se enfoca en realizar las llamadas necesarias al modelo para capturar los datos y devolverlos a la vista para que en esta sean mostrados al usuario. Este responde a eventos,

usualmente acciones del usuario e invoca cambios en el modelo. En resumen, se encarga de asimilar las peticiones del usuario y realizar los cambios apropiados en el modelo o en la vista.

Relación entre el framework y arquitectura:

Las capas del MVC que utiliza Symfony, están compuestas principalmente por seis archivos (scripts), los cuales se distribuyen de la siguiente forma:

Modelo:	Vista:	Controlador:
<ul style="list-style-type: none"> Abstracción de la base de datos. Acceso a datos. 	<ul style="list-style-type: none"> Plantillas. Layouts. 	<ul style="list-style-type: none"> Controladora final. Acciones.

Tabla 8: Modelo Vista Controlador SIGICEM

Propel, como Objeto de Mapeo Relacional (ORM, por sus siglas en inglés), trabaja generando de forma automática las clases existentes en la capa del modelo en dependencia de la estructura de datos de la aplicación, creando los métodos para acceder y modificar los datos. La abstracción de la base de datos se gestiona de forma transparente para el programador, con la utilización Objetos de Datos de PHP, razón por la cual es posible migrar de sistema gestor de bases de datos.

La lógica de la vista se transforma en un archivo de configuración simple, para que esto no conlleve a grandes cambios, debido a que generalmente, el layout es global en toda la aplicación o en un grupo de páginas. La plantilla sólo se encarga de dar salida a las variables definidas en el controlador. El controlador afronta como principal objetivo el manejo de las peticiones del usuario, de la seguridad, cargar la configuración de la aplicación y otras funciones.

Este se genera de forma automática, dividiendo las tareas en un controlador frontal, que es único para cada aplicación, y las acciones, que incluyen el código específico del controlador de cada página. El mismo ofrece un punto de entrada único para toda la aplicación, por lo que de ser necesario impedir el acceso a la aplicación, solamente se debe editar el script correspondiente. El controlador frontal y el layout, generalmente son comunes para todas las acciones de la aplicación.

Beneficios de su uso:

Permite la inserción de nuevas vistas y mejora las formas de procesar las órdenes del usuario. Las labores de mantenimiento también se simplifican y se reduce el tiempo necesario para ellas. Las correcciones solo se deben hacer en un solo lugar y no en varios como sucedería si se tuviese una mezcla de presentación e implementación de la lógica del negocio. Las vistas también son susceptibles de modificación sin necesidad de provocar que todo el sistema se paralice. Adicionalmente el patrón MVC propende a la especialización de cada rol del equipo, por tanto en cada liberación de una nueva versión se verán los resultados. (28)

3.4 Patrones del diseño

Los patrones de diseño son un conjunto de prácticas de óptimo diseño que se utilizan para abordar problemas recurrentes en la programación orientada a objetos, son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. Existen varios patrones de diseño, los cuales se clasifican como se muestra a continuación:

- 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: Describen la comunicación entre los objetos y clases. (29)

Los **Patrones GRASP** describen los principios fundamentales del diseño de objetos para la asignación de responsabilidades. A las clases del sistema se le asignaron tareas que podían realizar según la información que poseían, así como la creación de objetos (instancias) de otras clases, haciéndose notar los patrones experto y creador. (30)

- Experto: Es un patrón que se utiliza para asignar responsabilidades, es un principio básico que es útil en la programación orientada a objetos. Se conserva el encapsulamiento pues los objetos se valen de su propia información para realizar lo que se pide. Con la inclusión de Propel como ORM, este genera las clases para la gestión de las entidades asignando las responsabilidades correctamente.

- Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. En el siguiente ejemplo se muestra como la clase **funcionesjqplot** es la responsable de toda la información referente a las gráficas que se utilizan para generar los reportes



Figura 11: Ejemplo del patrón Creador

- Bajo Acoplamiento: Es una medida de la fuerza con que una clase está conectada a otras clases, una clase no depende de muchas. Su principal beneficio es que no se afectan por cambios otros componentes usados, por lo cual son fáciles de reutilizar. En Symfony se manifiesta este patrón en las clases del modelo, las cuales no se asocian con las vistas sino con los controladores; manteniendo de esta forma una baja dependencia, dando resultado a un bajo acoplamiento.
- Alta Cohesión: Es una medida de hasta qué punto están relacionadas y enfocadas las responsabilidades de una clase, caracteriza a las clases con responsabilidades estrechamente relacionada que no realice un trabajo enorme. Sus ventajas vienen dadas porque mejora la claridad y facilidad con que se entiende el diseño y soporta mayor capacidad de reutilización.

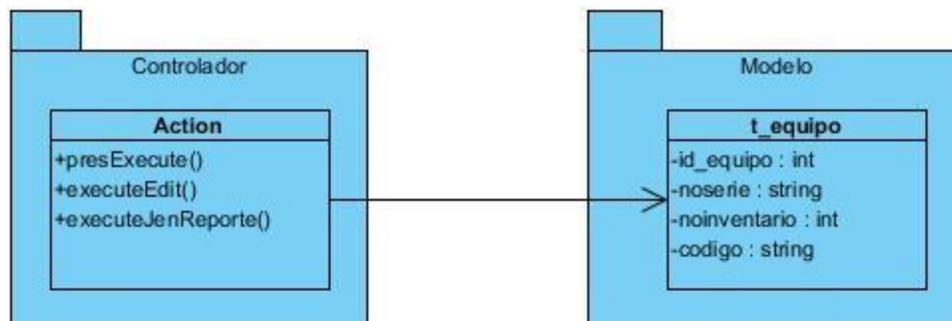


Figura 12: Ejemplo del patrón Alta Cohesión

- **Controlador:** Asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. (31)

La presencia del controlador frontal, es el prototipo básico que evidencia de forma clara su utilización en Symfony. Este maneja las peticiones del usuario, la seguridad, carga la configuración de la aplicación y otras tareas, es único para cada aplicación.

Patrones GOF: Gang of four o Pandilla de los cuatro

Los patrones GOF se describen como una forma indispensable de enfrentarse a la programación, trabajan en procesos de desarrollo de software orientados a la web. Inicialmente se construye un conjunto de criterios para evaluar y seleccionar procesos de desarrollo formales de gran envergadura, ellos proponen soluciones a problemas concretos, no son teorías genéricas. Se utilizan en situaciones frecuentes, debido a que se basan en la experiencia acumulada al resolver problemas reiterativos, favoreciendo la reutilización de código.

- **Decorador:** El framework Symfony posee una plantilla global (layout.php) para todo el proyecto, en donde se almacena todo el código que es similar en todas las interfaces de la aplicación, impidiendo tener que repetirlo en cada una de las mismas. De esta forma el contenido de cada interfaz se integra a la plantilla global.

3.5 Modelo de diseño

El modelo de diseño es una representación de la solución que se va a construir. Se realiza basándose en los requisitos del cliente, y al mismo tiempo la calidad se puede evaluar y comparar con el conjunto de criterios predefinidos para obtener un buen diseño. En el contexto de la ingeniería del software el diseño se centra en cuatro áreas importantes de interés: datos, arquitectura, interfaces y componentes. (32)

Debido al uso de Symfony, es objetivo delimitar que elementos del diseño corresponden al framework. En vista de lograr dicha separación se engloban de manera general en un subsistema denominado “Componentes de Symfony”, todos aquellos componentes propios de Symfony que actúan o intervienen.

De igual forma ocurre con los componentes propios que han sido implementados para el SIGICEM. Los desarrolladores han definido varias funcionalidades que disminuyen el grado de dificultad de integración de Symfony con ExtJS, también algunos que permiten la importación y exportación de documentos en formatos XLS y PDF, entre otras funcionalidades. Por ello se crea el subsistema “Componentes de SIGICEM”, en el que se reúnen estas funcionalidades. Ambos subsistemas se encontrarán representados dentro del subsistema “Componentes”, del cual se consumirán las prestaciones.

Diagramas de clases del diseño

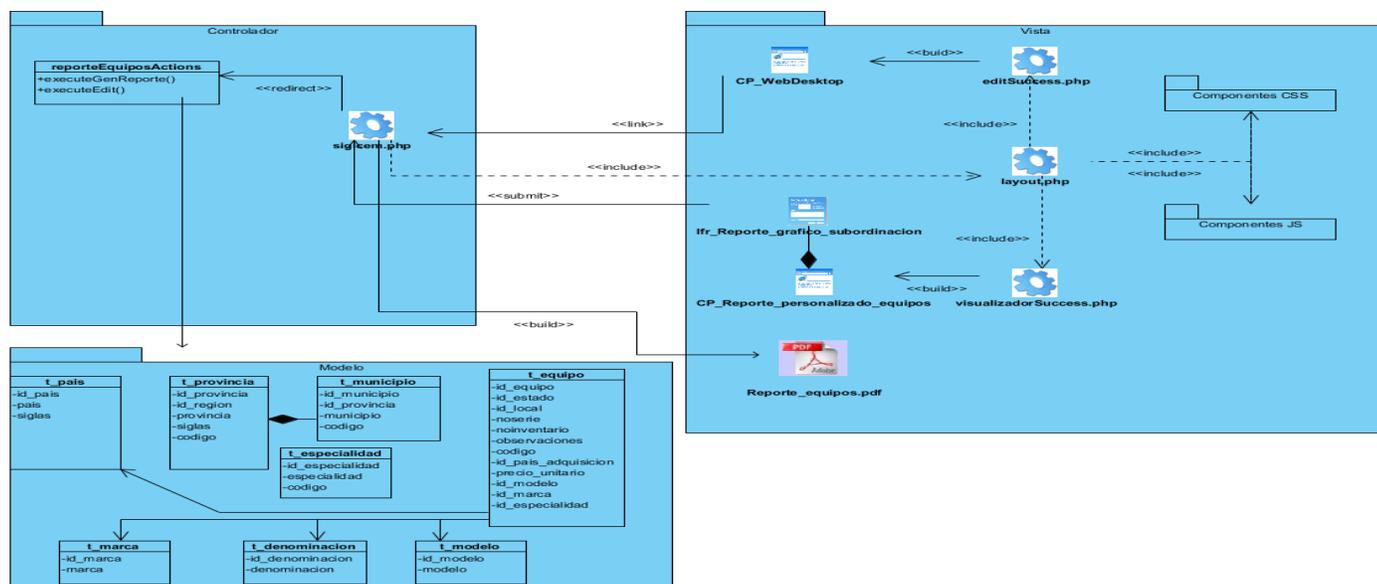


Figura 13: Diagrama de clases del Diseño: Generar reporte personalizado de equipos médicos.

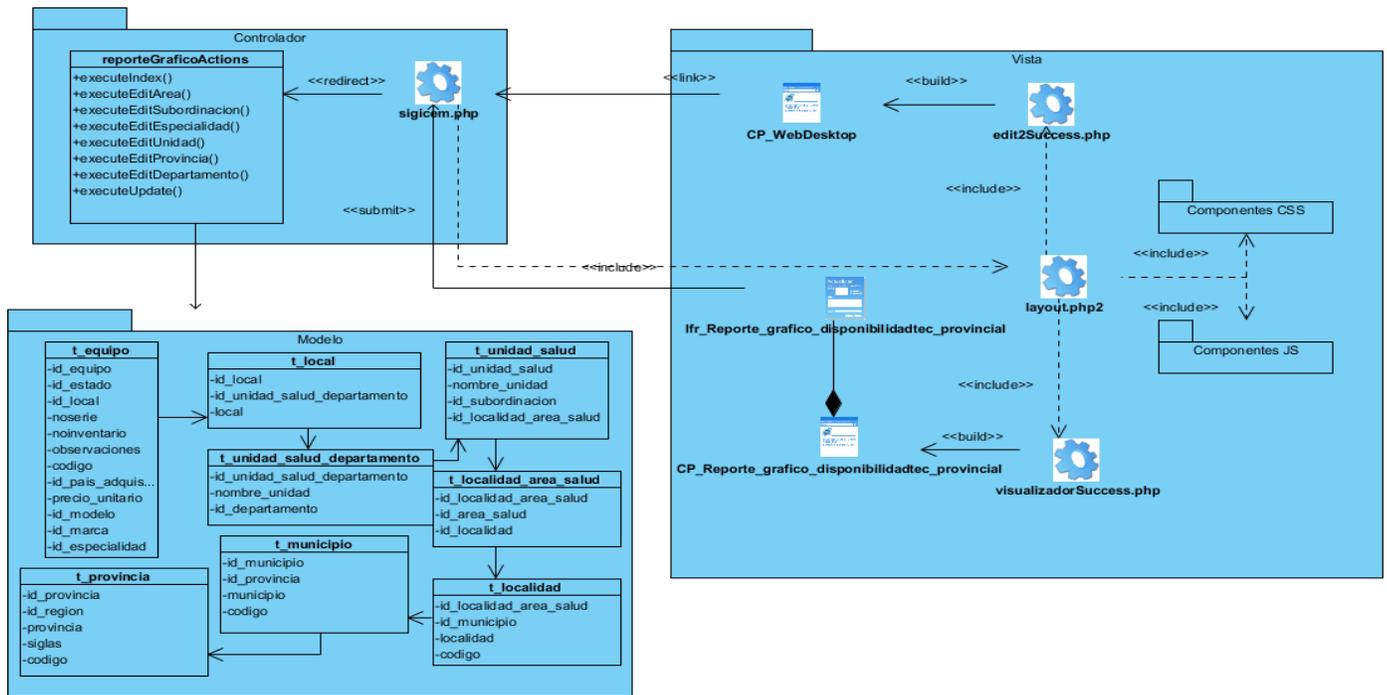


Figura 14: Diagrama de clases del Diseño: Generar reporte gráfico disponibilidad técnica por provincia.

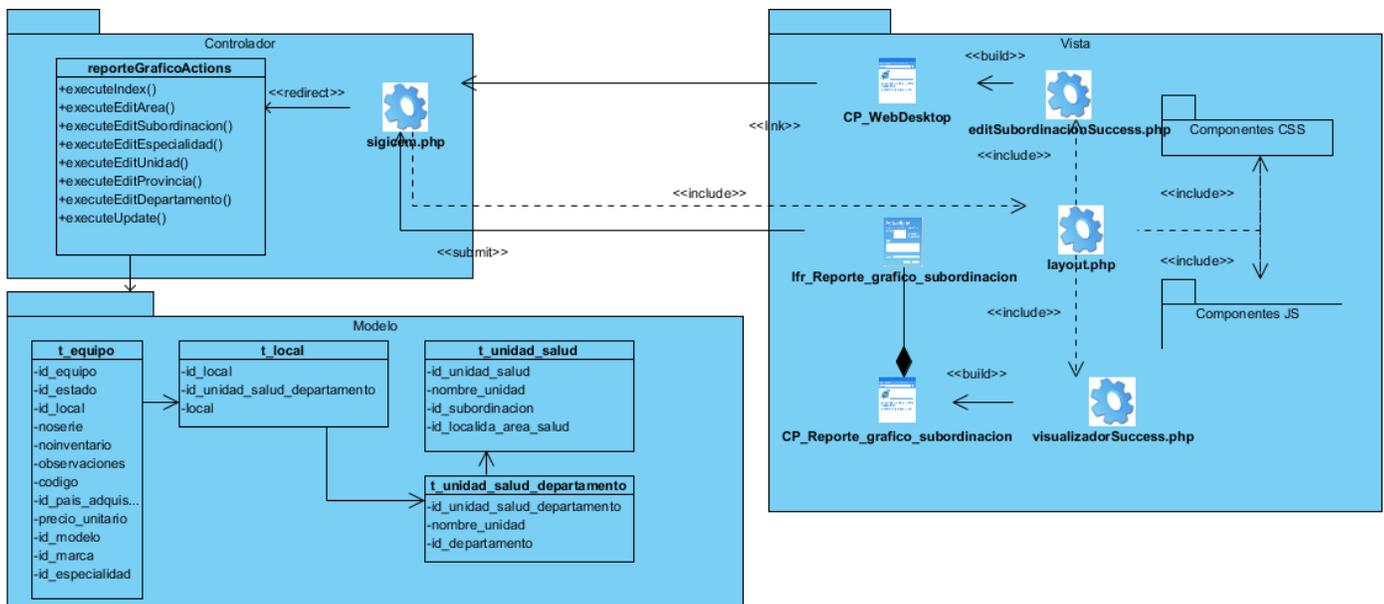


Figura 15: Diagrama de clases del Diseño: Generar reporte gráfico disponibilidad técnica por subordinación nacional.

En el capítulo se puntualizó la arquitectura a utilizar atendiendo a los beneficios que generan su uso para la implementación y desarrollo en el sistema. Se realizaron los diagramas de análisis y diseño de cada una de las funcionalidades del sistema para brindar al desarrollador una idea más clara de lo que se debe implementar.

Capítulo IV: Implementación del sistema

Este capítulo presentan las principales características de la implementación del sistema, se muestran los principales componentes y sus relaciones, haciendo uso de los diagramas en correspondencia a los casos de uso definidos anteriormente. Además se define el diagrama de despliegue, artefacto correspondiente al flujo de trabajo implementación. Se exponen las estrategias de seguridad, los estándares de codificación así como el tratamiento a los errores.

4.1 Modelo de datos

El modelo de datos es concebido por RUP dentro del proceso que incluye generar la base de datos siguiendo esta metodología. Muestra las tablas de la base de datos en la implementación del subsistema, permite describir los datos contenidos en las mismas y las relaciones que existen entre cada tabla. El modelo de datos, el cual es el resultado de la ingeniería inversa, brinda la base conceptual para el desarrollo del sistema y la decisión de cómo se almacenarán los datos y cómo se accederá a ellos.

Proporciona una representación visual y física de los datos persistentes del sistema, que en el futuro serán la base de datos. Surge a partir del diagrama de clases persistentes y su forma se expresa mediante un diagrama de UML. Determina la estructura de la información con el propósito de beneficiar la comunicación y la precisión en aplicaciones que usan e intercambian datos.

Para obtener mayor información acerca de las tablas del modelo de datos utilizadas en la realización del módulo Reportes Estadísticos se recomienda consultar el expediente del proyecto SIGICEM del departamento Sistema Especializados en Salud (SES). (33)

A continuación se presenta la modelo de datos del módulo:

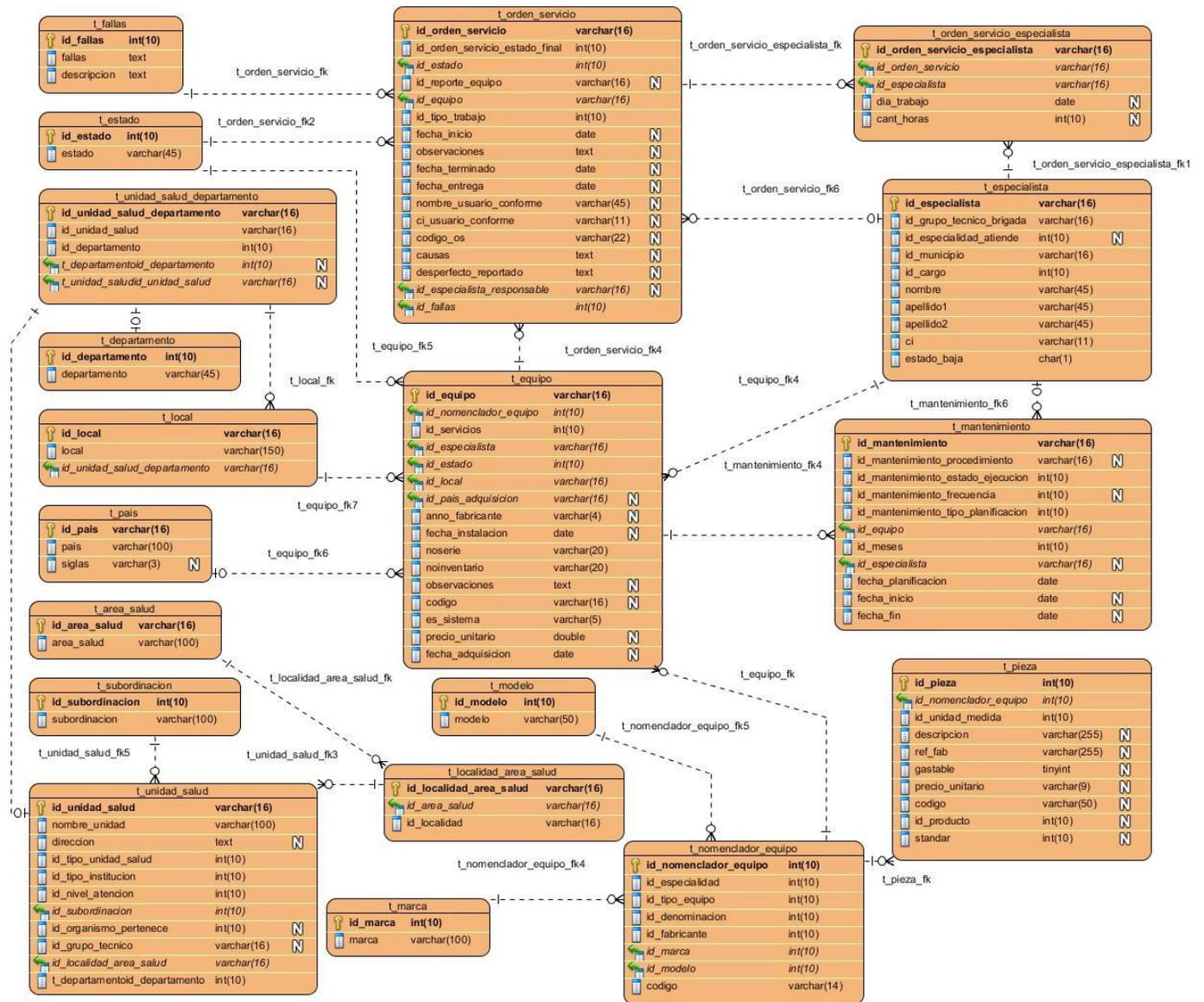


Figura 16: Modelo de datos

Descripción de las tablas del modelo de datos

A continuación se muestra para cada una de las tablas con que cuenta el módulo Reportes Estadísticos v2.0, una breve descripción de la entidad, así como de los principales atributos que emplea: llave primaria (PK por sus siglas en inglés primary key) y llave foránea (FK por sus siglas en inglés foreign key) y tipo de dato.

Nombre de la clase: t_equipo		
Descripción: Tabla que almacena la información referente a los equipos médicos del sistema.		
Atributo	Tipo	Descripción
id_equipo(PK)	varchar(16)	Almacena el identificador de los equipos médicos.
id_nomenclador_equipo(FK)	int(10)	Representa la llave foránea que viene de la tabla t_nomenclador_equipo.
id_servicios(FK)	int(10)	Representa la llave foránea que viene de la tabla t_servicios.
id_especialista(FK)	varchar(16)	Representa la llave foránea que viene de la tabla t_especialista.
id_estado(FK)	int(10)	Representa la llave foránea que viene de la tabla t_estado.
id_local(FK)	varchar(16)	Representa la llave foránea que viene de la tabla t_local.
id_pais_adquisicion(FK)	varchar(16)	Representa la llave foránea que viene de la tabla t_pais.
anno_fabricante	varchar(4)	Representa la fecha en que fabricó el equipo.
fecha_instalación	date	Representa la fecha en que se realizó la instalación del equipo.
noserie	varchar(20)	Representa el número de serie del equipo.
noinventario	varchar(20)	Representa el número de inventario del equipo.
observaciones	text	Datos de interés referente al equipo.
código	varchar(16)	Código propio del equipo asignado por el fabricante.

es_sistema	varchar(5)	Representa si el equipo está compuesto por varios dispositivos y en su conjunto conforma un sistema.
precio_unitario	double	Valor monetario correspondiente a cada equipo.
fecha_adquisicion	date	Representa la fecha en que se adquirió el equipo.

Tabla 9: Descripción de la tabla: t_equipo

Nombre de la clase: t_especialista		
Descripción: Tabla que almacena la información referente a los especialistas del sistema.		
Atributo	Tipo	Descripción
id_especialista(PK)	varchar(16)	Almacena el identificador de los especialistas.
id_grupo_tecnico_brigada(FK)	varchar(16)	Representa la llave foránea que viene de la tabla t_grupo_tecnico_brigada.
id_especialidad_atiende(FK)	int(10)	Representa la llave foránea que viene de la tabla t_especialidad_atiende.
id_municipio(FK)	varchar(16)	Representa la llave foránea que viene de la tabla t_municipio.
id_cargo(FK)	int(10)	Representa la llave foránea que viene de la tabla t_cargo.
nombre	varchar(45)	Nombre del especialista.
apellido1	varchar(45)	Primer apellido del especialista.
apellido2	varchar(45)	Segundo apellido del especialista.
ci	varchar(11)	Carnet de identidad correspondiente

		a cada especialista.
estado_baja	char(1)	Valor binario: 1 si el especialista ha sido dado de baja y 0 si no ha causado baja

Tabla 10: Descripción de la tabla: t_especialista

Nombre de la clase: t_mantenimiento		
Descripción: Tabla que almacena la información referente a los mantenimientos del sistema.		
Atributo	Tipo	Descripción
id_mantenimiento(PK)	varchar(16)	Almacena el identificador correspondiente a cada mantenimiento.
id_mantenimiento_procedimiento(FK)	varchar(16)	Representa la llave foránea que viene de la tabla t_mantenimiento_procedimiento.
id_mantenimiento_estado_ejecución(FK)	int(10)	Representa la llave foránea que viene de la tabla t_mantenimiento_estado_ejecución.
id_mantenimiento_frecuencia(FK)	int(10)	Representa la llave foránea que viene de la tabla t_mantenimiento_mantenimiento_frecuencia.
id_mantenimiento_tipo_planificación(FK)	int(10)	Representa la llave foránea que viene de la tabla t_mantenimiento_tipo_planificación.
id_equipo(FK)	varchar(16)	Representa la llave foránea que viene de la tabla t_equipo.
id_meses(FK)	int(10)	Representa la llave foránea que viene de la tabla t_meses.

id_especialista(FK)	varchar(16)	Representa la llave foránea que viene de la tabla t_especialista.
fecha_planificación(FK)	date	Representa la fecha en que se planificó el mantenimiento.
fecha_inicio(FK)	date	Representa la fecha en que comenzó el mantenimiento.
fecha_fin(FK)	date	Representa la fecha en que terminó el mantenimiento.

Tabla 11: Descripción de la tabla: t_mantenimiento

Nombre de la clase: t_pieza		
Descripción: Tabla que almacena la información referente a las piezas del sistema.		
Atributo	Tipo	Descripción
id_pieza(PK)	int(10)	Almacena el identificador correspondiente a cada pieza.
id_nomenclador_equipo(FK)	int(10)	Representa la llave foránea que viene de la tabla t_nomenclador_equipo.
id_unidad_medida	int(10)	Representa la llave foránea que viene de la tabla t_unidad_medida.
descripción	varchar(255)	Representa las principales características de la pieza.
ref_fab	varchar(255)	Identificador que define el fabricante de cada pieza.
gastable	tinyint	Valor binario: 1 si la pieza deprecia en su funcionamiento, 0 sino pierde facultades de rendimiento.
precio_unitario	int(9)	Valor monetario correspondiente a

		cada pieza.
código	varchar(50)	Código propio de la pieza asignado por el fabricante.
id_producto	int(10)	Almacena el identificador correspondiente a cada pieza como producto.
standar	int(10)	Almacena el estándar correspondiente a la pieza según sus normas de producción

Tabla 12: Descripción de la tabla: t_pieza

Nombre de la clase: t_subordinación

Descripción: Tabla que almacena la información referente a cada tipo de subordinación de las áreas de salud existentes en el sistema.

Atributo	Tipo	Descripción
id subordinación(PK)	int(10)	Almacena el identificador correspondiente a cada subordinación.
subordinación	varchar(100)	Nombre del tipo de subordinación de las áreas de salud.

Tabla 13: Descripción de la tabla: t_subordinación

Nombre de la clase: t_unidad_salud

Descripción: Tabla que almacena la información referente a las unidades de salud del sistema.

Atributo	Tipo	Descripción
id_unidad_salud(PK)	varchar(16)	Almacena el identificador correspondiente a cada unidad de

		salud.
nombre_unidad	varchar(100)	Nombre de cada unidad de salud
dirección	text	Dirección de la área de salud
id_tipo_unidad_salud	int(10)	Representa la llave foránea que viene de la tabla t_tipo_unidad_salud.
id_tipo_institución	int(10)	Representa la llave foránea que viene de la tabla t_tipo_institución.
id_nivel_atención	int(10)	Representa la llave foránea que viene de la tabla t_nivel_atención.
id_subordinación	int(10)	Representa la llave foránea que viene de la tabla t_subordinación.
id_organismo_pertenece	int(10)	Representa la llave foránea que viene de la tabla t_organismo_pertenece.
id_grupo_tecnico	varchar(16)	Representa la llave foránea que viene de la tabla t_grupo_tecnico.
id_localidad_area_salud	varchar(16)	Representa la llave foránea que viene de la tabla t_localidad_area_salud.

Tabla 14: Descripción de la tabla: t_unidad_salud

4.2 Modelo de implementación

El modelo de implementación describe los elementos del modelo del diseño, es comprendido por un grupo de componentes y subsistemas que forman la composición física de la implementación del sistema. Estos incluyen: ficheros ejecutables, ficheros de código fuente, necesarios para la implantación y despliegue del sistema.

Este flujo de trabajo tiene como propósito definir la organización del código e implementar las clases y subsistemas encontrados durante el diseño, posibilitando la reutilización, que se pueda implementar por separado y disminuye el impacto que pueda traer consigo un cambio. (34)

Muestra las organizaciones y dependencias lógicas entre componentes. Presenta la separación del sistema de software en componentes físicos (cabecera, paquetes, archivos, módulos). Los diagramas de componentes muestran los componentes de software que constituyen una parte reusable, sus interfaces, y sus interrelaciones.

Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas, pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente. Describe la estructura de los componentes del sistema agrupados por paquetes lógicos y muestra las dependencias entre estos componentes. El mismo es utilizado para modelar la vista estática y dinámica de un sistema, además de mostrar organización entre sus componentes. (35)

Diagrama de componentes

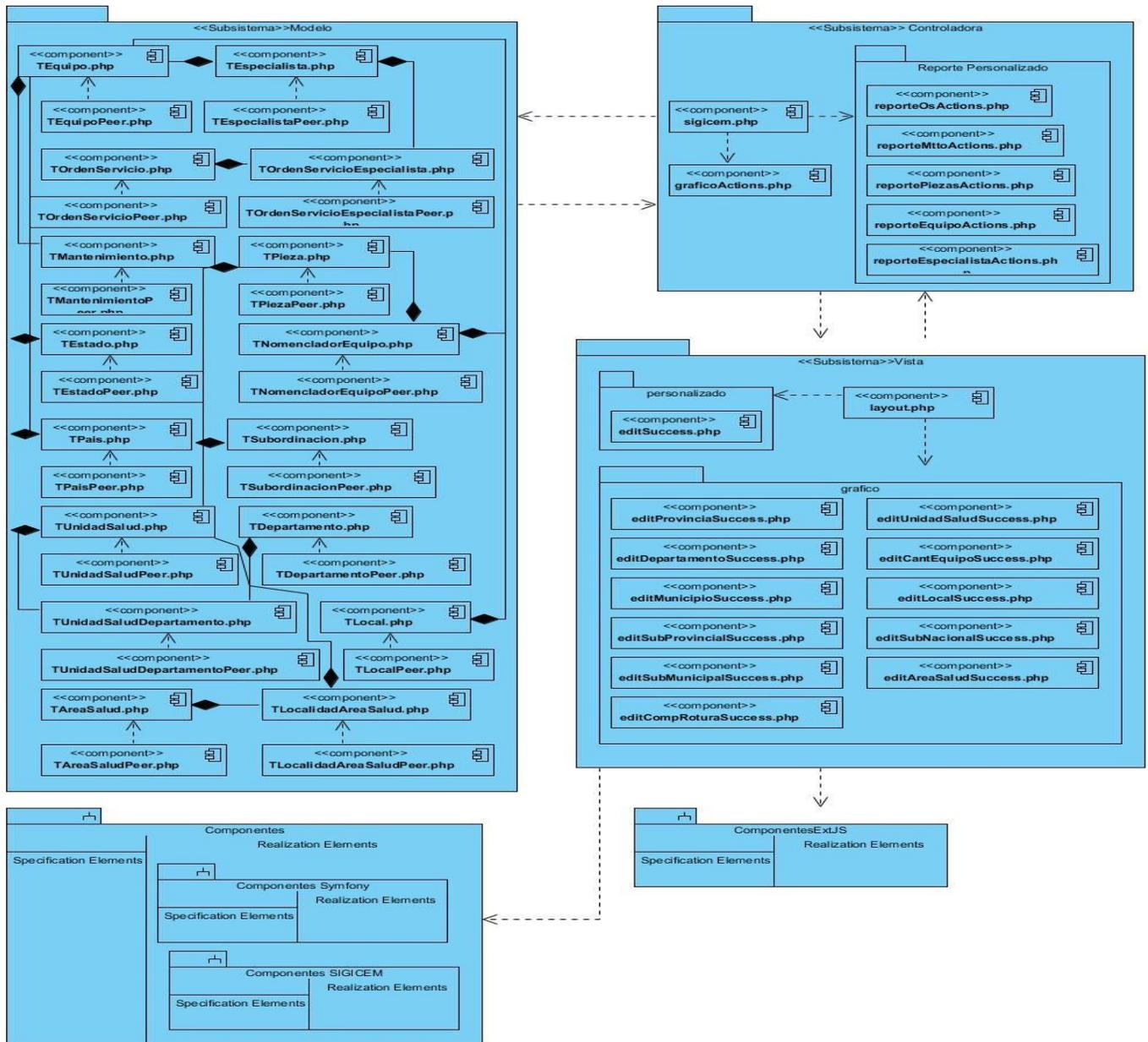


Figura 17: Diagrama de Componentes Módulo Reportes Estadísticos V2.0.

Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema mostrando cómo se distribuye la funcionalidad entre los nodos, con ellos se modela la topología del hardware sobre el que se ejecuta el sistema. El mismo se utiliza como entrada fundamental en las actividades de implementación y diseño debido a que la distribución del sistema tiene una influencia principal en su diseño. (36)

Se utiliza para tomar a los elementos de configuración del procesamiento y las conexiones entre esos elementos. También se utiliza para visualizar la distribución de los componentes de software en los nodos físicos. (37) A continuación se presenta el modelo de despliegue correspondiente al sistema SIGICEM.

Diagrama de despliegue

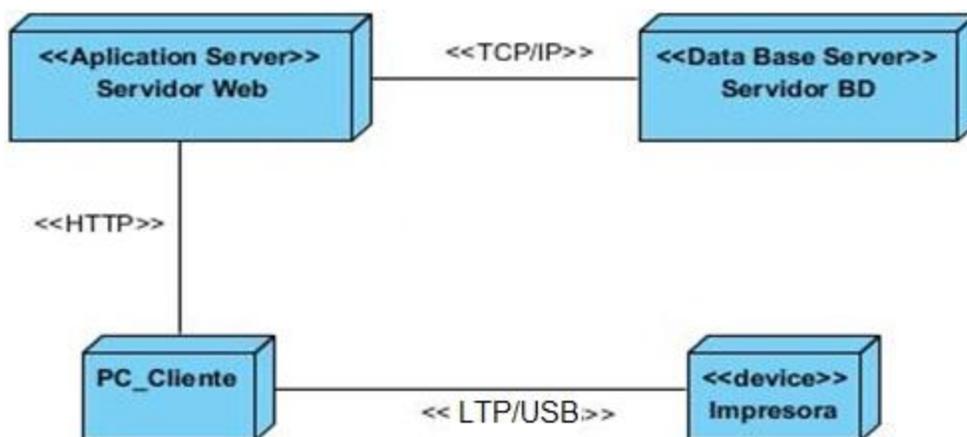


Diagrama de Despliegue Módulo Reportes Estadísticos v2.0.

4.3 Estilos y estándares de codificación

Un estándar de codificación son reglas que se siguen para la escritura del código fuente. De tal manera que a los programadores se les facilite entender dicho código. Es empleado como un modelo para evitar los posibles errores que pueden existir, para entender o escribir algún código, regula la realización de ciertos procesos o la fabricación de componentes para garantizar la interoperabilidad del sistema. (38)

El uso de estilos de codificación ayuda a la calidad del software por lo que es de gran importancia su puesta en práctica en proyectos informáticos. Luego de un estudio realizado acerca de los estilos y

estándares de codificación definidos por el centro (CESIM) y el departamento (SES), se determinó utilizar la misma codificación que la versión anterior del módulo, debido a que cumple con las pautas previstas para el desarrollo. (39)

A continuación, se presentan los estándares y estilos de codificación utilizados en la implementación del

Módulo Reportes Estadísticos v2.0:

Identación.		
Objetivo: Lograr una estructura uniforme para los bloques de código así como para los diferentes niveles de anidamiento.		
Inicio y fin de bloque	Se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque {}. Al igual que para el caso de las sentencias if, else, for, while, do while, switch, foreach.	
Aspectos generales	El indentado debe tomar dos espacios por bloque de código. No se recomienda el uso del tabulador; debido a que varían según la PC o la configuración de dicha tecla. Los inicios ({} y cierre ()) de ámbito deben de estar en línea debajo de la declaración a la que pertenecen y debe tratarse de eliminarse si sólo existe una instrucción. Nunca colocar ({} en cualquier línea de un código, esto requiere una línea propia.	
Comentarios, separadores, líneas, espacios en blanco y márgenes.		
Objetivo: Identificar un modo común para comentar o describir el código de forma tal que sea comprensible con sólo leerlo una vez.		
Ubicación de comentarios	Al iniciar cada clase o función y al final de cada bloque de código.	Se aconseja comentar al inicio de la clase o función especificando su objetivo, así como los parámetros que usa (tipos de dato y objetivo del parámetro).
Líneas en blanco	Se emplean al inicio y fin de los métodos, clases y estructuras.	Se advierte dejar una línea en blanco antes y después de la declaración de una clase o de una estructura y de una

		función.
Espacios en blanco	Se emplean entre operadores lógicos y aritméticos.	Se recomienda el uso de espacios en blanco entre operadores para establecer una mayor legibilidad en el código. Ejemplo: equipo = nomequipo.
Aspectos generales	Sobre los espacios en blanco.	Evitar los espacios después de un [y antes de un] de un arreglo, al igual que antes de un (y un), así como con el uso del punto y coma.
	Sobre el comentario.	Tratar de evitar comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. Para comentar una sola instrucción se elimina la línea en blanco o se escribe a continuación.
Variables y constantes.		
Aspectos generales	El nombre utilizado, debe permitir conocer el propósito de la misma.	
Clases y objetos.		
Objetivo: Nombrar las clases y objetos de forma estándar para toda la aplicación.		
Apariencia de clases y objetos	Primera letra en mayúscula.	Los nombres de las clases comenzarán con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing*. Ejemplo: MisFuncionesJq ().

Apariencia de atributos	Primera letra en minúscula.	El nombre de los atributos de las clases debe escribirse con letras minúsculas, en caso de que sea un nombre compuesto se utilizará underscore (_) para separarlo. Ejemplo: mis_atributos.
Apariencia de las funciones	Primera letra en mayúscula.	Para el nombre de las funciones se debe utilizar verbos que marcan la acción que realice la función. Se empleará notación PascalCasing. Ejemplo: function LineasDiscontinuas ().
Aspectos generales	El nombre utilizado para las clases, objetos, atributos y funciones debe permitir que con solo leerlo se conozca el propósito de los mismos.	

Tabla 15: Estándares y Estilos de Codificación

4.4 Tratamiento de errores

El tratamiento de errores es un proceso indispensable para el buen funcionamiento del sistema. Los errores más comunes que se pueden presentar en el funcionamiento del módulo implementado, están relacionados con la BD.

Un formulario debe incluir validaciones de los datos que proporciona el usuario antes de enviar la información al servidor. Para esto se deben realizar en el cliente tantos chequeos como sea posible, así se mejora el tiempo de respuesta real y se reduciría la carga del servidor. Con la validación de los datos, se evitan muchos de estos problemas; esta se puede realizar en el lado del servidor y en el lado del cliente. En el servidor es obligatoria para no corromper la base de datos con datos incorrectos y en el lado del cliente es opcional, pero mejora la experiencia de usuario. (40)

Los sistemas informáticos durante su uso pueden producir errores en las diferentes rutinas, a estos errores en el sistema se les conoce por el nombre de excepciones o fallas. Mediante el tratamiento de las excepciones se asegura un estado en que las rutinas pueden continuar su ejecución, lo que permite lograr un sistema confiable y seguro. (41)

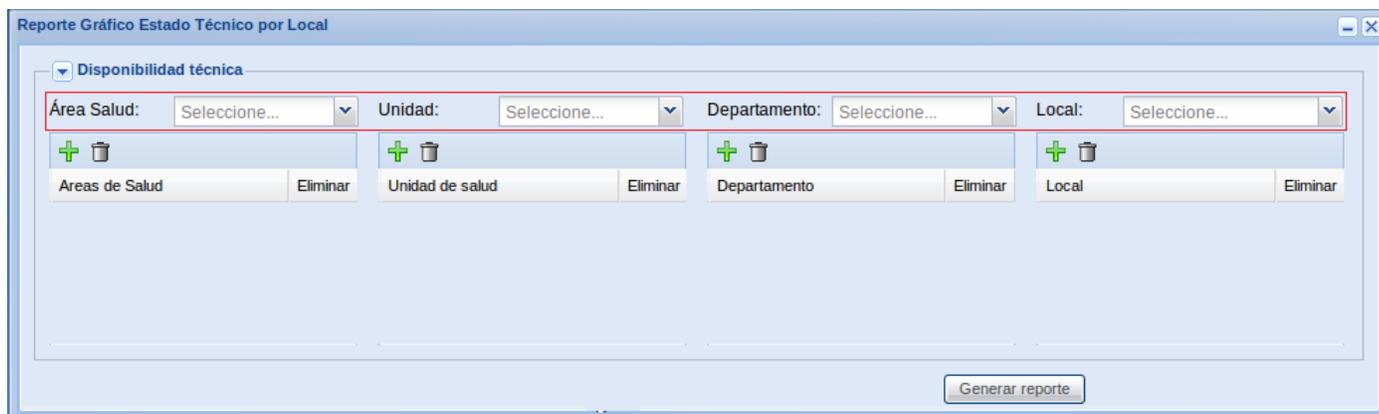


Figura 18: Ejemplo de Cuadros de Selección utilizados .

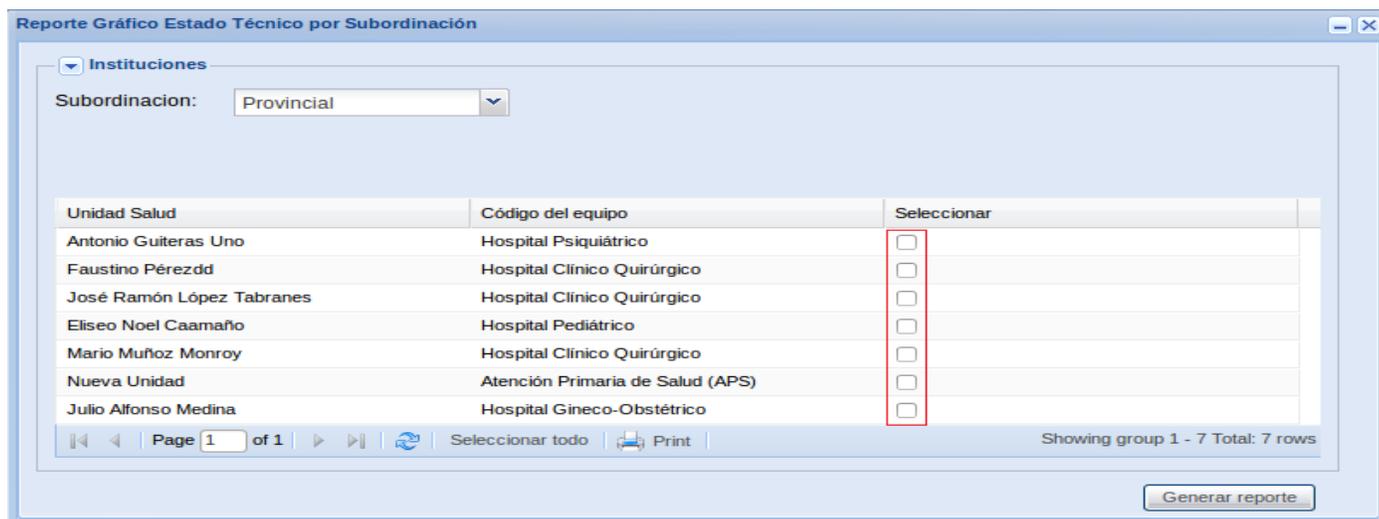


Figura 19: Ejemplo de Menú de Selección utilizados .

En este capítulo se desarrolló la descripción de la estructura de las tablas sobre las que se constituye el modelo de datos del sistema, así como sus atributos y relaciones. Se presentó el diagrama de despliegue y los elementos vinculados con la implementación del sistema en términos de componentes, cumpliendo además con el tratamiento de errores definido anteriormente y las estrategias de codificación.

Conclusiones

- ✓ El estudio realizado acerca de los sistemas utilizados para la generación de reportes estadísticos en diferentes ámbitos, las deficiencias encontradas en la anterior aplicación y los nuevos requerimientos identificados, evidenciaron la necesidad de llevar a cabo el desarrollo del módulo Reportes Estadísticos para el Sistema de Gestión para Ingeniería Clínica y Electromedicina v2.0.
- ✓ El análisis de los procesos de gestión de reportes en el Sistema de Gestión para Ingeniería Clínica y Electromedicina, así como las nuevas peticiones por parte del Ministerio de Salud Pública, marcaron los pasos para el diseño e implementación de la solución desarrollada.
- ✓ La descripción de la metodología, las herramientas y las tecnologías definidas por el departamento Sistemas Especializados en Salud, ratificó que estas poseían las características necesarias para desarrollar módulo Reportes Estadísticos.
- ✓ La elaboración de los artefactos correspondientes a las disciplinas: Modelado de Negocio, Requisitos, Análisis y Diseño e Implementación según la metodología RUP, permitió establecer la documentación necesaria de la investigación, sirviendo como guía al desarrollador para implementar la solución obtenida.
- ✓ El desarrollo de la segunda versión del módulo Reportes Estadísticos dota al Sistema de Gestión para Ingeniería Clínica y Electromedicina de un nuevo conjunto de facilidades y funcionalidades para los especialistas del Centro de Ingeniería Clínica y Electromedicina.

Recomendaciones

- ✓ Integrar el módulo Reportes Estadísticos v2.0 al Sistema de Gestión para la Ingeniería Clínica y Electromedicina, para permitir su uso por los especialistas en las áreas de salud.
- ✓ Incorporar al sistema funcionalidades para el uso del protocolo PHP/JavaBridge, el cual permitirá la utilización de las bondades del motor JasperReport para la generación de reportes.

Referencias Bibliográficas

1. **Salomón, Omar Pérez.** Los Joven Club en Fidel Castro. *Cubadebate*. [En línea] [Citado el: 1 de febrero de 2014.] <http://www.cubadebate.cu/especiales/2011/09/15/los-joven-club-en-fidel-castro/>.
2. **Ledo, Ariel Delgado Ramos y María Vidal.** Informática en la salud pública cubana . [En línea] [Citado el: 1 de febrero de 2014.] http://bvs.sld.cu/revistas/spu/vol32_3_06/spu15306.htm.
3. **Hernández, Sayda.** *El proceso de investigación científica*. La Habana : Editorial Universitaria, 2008.
4. Concepto de información-Definición,Significado. [En línea] [Citado el: 16 de enero de 2014.] <http://definicion.de/informacion/#ixzz2mgKE89UF>.
5. Tipos de Información. [En línea] [Citado el: 20 de febrero de 2014.] <http://10ejemplos.com/tipos-de-informacion>.
6. Definición de reporte. [En línea] [Citado el: 16 de enero de 2014.] <http://definicion.de/reporte/#ixzz2mzFDUKPG>.
7. **Daniela Chueke.** Las ventajas de informatizar un hospital desde su apertura. [En línea] 5 de marzo de 2012. [Citado el: 28 de noviembre de 2013.] <http://www.ehealthreporter.com/es/noticia/verNoticia/902>.
8. PHP Report Maker. [En línea] 20 de enero de 2014. [Citado el: 18 de febrero de 2014.] <http://www.hkvstore.com/phpreportmaker/>.
9. PHP Report Maker. [En línea] [Citado el: 22 de febrero de 2014.] <http://www.softpedia.es/programa-PHP-Report-Maker-65336.html>.
10. JasperReports Library. [En línea] [Citado el: 19 de noviembre de 2013.] <https://www.jaspersoft.com/es/node/38063>.
11. **Ricardo Mariaca.** DynamicReports. [En línea] [Citado el: 18 de febrero de 2014.] <http://www.dynamicreports.org/>.

12. Reportico-Open Source PHP Report Designer. [En línea] [Citado el: 17 de febrero de 2014.] <http://www.reportico.org/site/index.php>.
13. **Aragón, Ismaray Morera.** *Módulo de Reportes Estadísticos del Sistema de Gestión para la Ingeniería Clínica y Electromedicina.* La Habana : s.n., 2013.
14. **Olivares, Yanoski Agneri Martínez Hernández y José Rolando Lafaurie.** *Sistema para la generación de reportes en la plataforma alasGRATO.* La Habana : s.n., 2008.
15. El Proceso Unificado de Desarrollo de Software (RUP). [En línea] [Citado el: 10 de febrero de 2014.] <http://yaqui.mx/uabc.mx/~molguin/as/RUP.htm>.
16. **Slideshare.** Arquitectura de 3 Capas. [En línea] 24 de junio de 2008. [Citado el: 14 de febrero de 2014.] <http://www.slideshare.net/decimo/arquitectura-3-capas>.
17. Ecured. [En línea] [Citado el: 5 de febrero de 2014.] http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3n.
18. LibrosWeb. [En línea] 3 de febrero de 2014. http://librosweb.es/symfony_1_4/capitulo_1/symfony_en_pocas_palabras.html.
19. ExtJs Comunidad en Español. [En línea] [Citado el: 13 de diciembre de 2013.] <http://extjses.com/>.
20. **Jesús, Pedro.** CodeManía: un espacio para escribir códigos. [En línea] [Citado el: 10 de febrero de 2014.] <http://codemania.cubava.cu/2013/09/que-es-extjs/>.
21. Master LAMP: La plataforma web libre . [En línea] [Citado el: 12 de diciembre de 2014.] http://www.ciberaula.com/curso/lamp/que_es/.
22. ¿Qué es jQuery? [En línea] [Citado el: 5 de febrero de 2014.] <http://www.mexired.com/blog/que-es-jquery/>.
23. **Mediavilla, Elena.** Programación Orientada a Objetos . [En línea] [Citado el: 24 de marzo de 2014.] http://www.ctr.unican.es/ asignaturas/MC_OO/Doc/Casos_de_uso.pdf.

24. Mundo Kramer's blog . [En línea] [Citado el: 26 de 4 de 2014.] <http://mundokramer.wordpress.com/2011/05/20/modelo-de-analisis-software/>.
25. Enciclopedia virtual. [En línea] [Citado el: 26 de 4 de 2014.] <http://www.eumed.net/libros-gratis/2008b/390/Modelos%20de%20 analisis.htm>.
26. **Zamitz, Carlos Alberto Román.** TEMAS ESPECIALES DE COMPUTACIÓN. *TEMAS ESPECIALES DE COMPUTACIÓN*. [En línea] [Citado el: 31 de marzo de 2014.] http://profesores.fi-b.unam.mx/carlos/aydoo/sequence_colaboration.html.
27. Facultad de Ingeniería. [En línea] [Citado el: 2014 de 4 de 25.] <http://www.fing.edu.uy/cpap/cursos/arquitectura-de-software>.
28. Neleste. [En línea] [Citado el: 25 de 4 de 2014.] <http://www.neleste.com/modelo-vista-controlador/>.
29. Kioskea. [En línea] [Citado el: 26 de 4 de 2014.] <http://es.kioskea.net/contents/224-patrones-de-diseno>.
30. **Cortez, Casallas, Gloria y Rubby.** *Introduccion a los patrones de software*. Andes : universidad de los Andes, 2010.
31. **Tedeschi, Nicolás.** Microsoft. [En línea] [Citado el: 26 de 4 de 2014.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
32. **Mendoza Navarro, Javier.** Diseño del Sistema de Tarjeta de Crédito. *Diseño del Sistema de Tarjeta de Crédito*. [En línea] 2012. [Citado el: 7 de 5 de 2013.] http://sisbib.unmsm.edu.pe/bibvirtualdata/tesis/basic/mendoza_nj/cap5.pdf.
33. REDMIC. [En línea] [Citado el: 8 de 4 de 2014.] <http://www.redmic.es/web/index.php/modelo-de-datos>.
34. [En línea] [Citado el: 8 de 4 de 2014.] <http://ithleovi.blogspot.com/2013/06/unidad-5-modelo-deimplementacion-el.html>.

35. ALTOVA. [En línea] [Citado el: 8 de 4 de 2014.] <http://www.altova.com/es/umodel/uml-deployment-diagrams.html>.
36. [En línea] [Citado el: 8 de 4 de 2014.] http://sazfcorp.com/rup.es/largeprojects/core.base_rup/workproducts/rup_deployment_model_57DF1DF5.html.
37. [En línea] [Citado el: 8 de 4 de 2014.] <http://umldiagramadespliegue.blogspot.com/>.
38. PHP Framework Interop Group. [En línea] [Citado el: 8 de 4 de 2014.] <http://www.php-fig.org/psr/psr-2/es/>.
39. Microsoft Developer Network. [En línea] [Citado el: 8 de 4 de 2014.] <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
40. National Instruments. [En línea] [Citado el: 8 de 4 de 2014.] <http://www.ni.com/academic/students/learnlabview/esa/debugging.htm>.
41. [En línea] [Citado el: 8 de 4 de 2014.] <http://www.bilib.es/fileadmin/descargas/guia/html/node18.html>.
42. **Arias, Yanet Sánchez Hernández y José Manuel Matos.** *Reportes Estadísticos del Registro de Partos y Nacimientos*. La Habana : Universidad de las Ciencias Informáticas, 2013.
43. Diccionario de Sinónimos y Antónimos. [En línea] [Citado el: 13 de febrero de 2014.] <http://www.wordreference.com/sinonimos/ciencia>.
44. Ejecutar código Java en PHP y viceversa. [En línea] 28 de julio de 2008. [Citado el: 18 de diciembre de 2013.] <http://gonetil.wordpress.com/2008/07/28/instalar-phpjava-bridge-en-linux/>.
45. **Soria, Julio César.** Clasificación de los reportes durante el desarrollo de sistemas. . *Clasificación de los reportes durante el desarrollo de sistemas.* . [En línea] 2005. [Citado el: 20 de febrero de 2014.] <http://lobyte.blogspot.com/2006/05/clasificacin-de-los-reportes-durante.html>.

46. Malavida. *Malavida*. [En línea] 26 de septiembre de 2009. [Citado el: 29 de enero de 2014.] <http://www.malavida.com/blog/13144/como-desinstalar-programas-en-ubuntu-completamente>.
47. SlideShare. *SlideShare*. [En línea] 25 de mayo de 2009. [Citado el: 6 de noviembre de noviembre.] <http://www.slideshare.net/ajdgeniz/aprenda-reportes-con-netbeans-y-ireport>.
48. Tutorial básico del programador web: PHP desde cero. *Tutorial básico del programador web: PHP desde cero*. [En línea] 2006. [Citado el: 4 de abril de 2014.] http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=70&Itemid=193.
49. **Orteza., Miquel Camps**. Aprender a programar PHP desde cero. *Aprender a programar PHP desde cero*. [En línea] [Citado el: 15 de abril de 2014.] <http://betabeers.com/curso/aprender-programar-php-desde-cero-402/>.
50. Php/ Java brigde. *Php/ Java brigde*. [En línea] [Citado el: 1 de noviembre de 2013.] <http://php-java-bridge.sourceforge.net/pjb/>.
51. stackoverflow. *stackoverflow*. [En línea] 2013. [Citado el: 20 de marzo de 2014.] <http://stackoverflow.com/questions/14886781/jqplot-display-button>.
52. MySQL. *MySQL*. [En línea] 2011. [Citado el: 20 de marzo de 2014.] <http://dev.mysql.com/doc/refman/5.0/es/examples.html>.
53. Comandos básicos para la terminal de Ubuntu . *Comandos básicos para la terminal de Ubuntu* . [En línea] [Citado el: 25 de abril de 2014.] <http://www.ubuntu-guia.com/2009/07/comandos-basicos-de-linux.html>.
54. **Guerrero, Carlos A**. Scielo. *Scielo*. [En línea] 28 de enero de 2013. [Citado el: 2 de abril de 2014.] http://www.scielo.cl/scielo.php?pid=S0718-07642013000300012&script=sci_arttext.

Bibliografía

[En línea] [Citado el: 8 de 4 de 2014.] <http://ithleovi.blogspot.com/2013/06/unidad-5-modelo-deimplementacion-el.html>.

[En línea] [Citado el: 8 de 4 de 2014.] http://sazfcorp.com/rup.es/largeprojects/core.base_rup/workproducts/rup_deployment_model_57DF1DF5.html.

[En línea] [Citado el: 8 de 4 de 2014.] <http://umldiagramadespliegue.blogspot.com/>.

[En línea] [Citado el: 8 de 4 de 2014.] <http://www.bilib.es/fileadmin/descargas/guia/html/node18.html>.

¿Qué es jQuery? [En línea] [Citado el: 5 de febrero de 2014.] <http://www.mexired.com/blog/que-es-jquery/>.

ALTOVA. [En línea] [Citado el: 8 de 4 de 2014.] <http://www.altova.com/es/umodel/uml-deployment-diagrams.html>.

Aragón, Ismaray Morera. *Módulo de Reportes Estadísticos del Sistema de Gestión para la Ingeniería Clínica y Electromedicina.* La Habana : s.n., 2013.

Arias, Yanet Sánchez Hernández y José Manuel Matos. *Reportes Estadísticos del Registro de Partos y Nacimientos.* La Habana : Universidad de las Ciencias Informáticas, 2013.

Comandos básicos para la terminal de Ubuntu . *Comandos básicos para la terminal de Ubuntu.* [En línea] [Citado el: 25 de abril de 2014.] <http://www.ubuntu-guia.com/2009/07/comandos-basicos-de-linux.html>.

Concepto de información-Definición,Significado. [En línea] [Citado el: 16 de enero de 2014.] <http://definicion.de/informacion/#ixzz2mgKE89UF>.

Cortez, Casallas, Gloria y Rubby. *Introduccion a los patrones de software.* Andes : universidad de los Andes, 2010.

Daniela Chueke. Las ventajas de informatizar un hospital desde su apertura. [En línea] 5 de marzo de 2012. [Citado el: 28 de noviembre de 2013.] <http://www.ehealthreporter.com/es/noticia/verNoticia/902>.

Definición de reporte. [En línea] [Citado el: 16 de enero de 2014.] <http://definicion.de/reporte/#ixzz2mzFDUKPG>.

Diccionario de Sinónimos y Antónimos. [En línea] [Citado el: 13 de febrero de 2014.] <http://www.wordreference.com/sinonimos/ciencia>.

Ecured. [En línea] [Citado el: 5 de febrero de 2014.] http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3n.

Ejecutar código Java en PHP y viceversa. [En línea] 28 de julio de 2008. [Citado el: 18 de diciembre de 2013.] <http://gonetil.wordpress.com/2008/07/28/instalar-phpjava-bridge-en-linux/>.

El Proceso Unificado de Desarrollo de Software (RUP). [En línea] [Citado el: 10 de febrero de 2014.] <http://yaqui.mxl.uabc.mx/~molguin/as/RUP.htm>.

Enciclopedia virtual. [En línea] [Citado el: 26 de 4 de 2014.] <http://www.eumed.net/libros-gratis/2008b/390/Modelos%20de%20 analisis.htm>.

ExtJs Comunidad en Español. [En línea] [Citado el: 13 de diciembre de 2013.] <http://extjses.com/>.

Facultad de Ingeniería. [En línea] [Citado el: 2014 de 4 de 25.] <http://www.fing.edu.uy/cpap/cursos/arquitectura-de-software>.

Guerrero, Carlos A. Scielo. *Scielo*. [En línea] 28 de enero de 2013. [Citado el: 2 de abril de 2014.] http://www.scielo.cl/scielo.php?pid=S0718-07642013000300012&script=sci_arttext.

Hernández, Sayda. *El proceso de investigación científica*. La Habana : Editorial Universitaria, 2008.

JasperReports Library. [En línea] [Citado el: 19 de noviembre de 2013.] <https://www.jaspersoft.com/es/node/38063>.

Jesús, Pedro. CodeManía: un espacio para escribir códigos. [En línea] [Citado el: 10 de febrero de 2014.] <http://codemania.cubava.cu/2013/09/que-es-extjs/>.

Kioskea. [En línea] [Citado el: 26 de 4 de 2014.] <http://es.kioskea.net/contents/224-patrones-de-diseno>.

Ledo, Ariel Delgado Ramos y María Vidal. Informática en la salud pública cubana . [En línea] [Citado el: 1 de febrero de 2014.] http://bvs.sld.cu/revistas/spu/vol32_3_06/spu15306.htm.

LibrosWeb. [En línea] 3 de febrero de 2014. http://librosweb.es/symfony_1_4/capitulo_1/symfony_en_pocas_palabras.html.

Malavida. *Malavida*. [En línea] 26 de septiembre de 2009. [Citado el: 29 de enero de 2014.] <http://www.malavida.com/blog/13144/como-desinstalar-programas-en-ubuntu-completamente>.

Master LAMP: La plataforma web libre . [En línea] [Citado el: 12 de diciembre de 2014.] http://www.ciberaula.com/curso/lamp/que_es/.

Mediavilla, Elena. Programación Orientada a Objetos . [En línea] [Citado el: 24 de marzo de 2014.] http://www.ctr.unican.es/asignaturas/MC_OO/Doc/Casos_de_uso.pdf.

Mendoza Navarro, Javier. Diseño del Sistema de Tarjeta de Crédito. *Diseño del Sistema de Tarjeta de Crédito*. [En línea] 2012. [Citado el: 7 de 5 de 2013.] http://sisbib.unmsm.edu.pe/bibvirtualdata/tesis/basic/mendoza_nj/cap5.pdf.

Microsoft Developer Network. [En línea] [Citado el: 8 de 4 de 2014.] <http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.

Mundo Kramer's blog . [En línea] [Citado el: 26 de 4 de 2014.] <http://mundokramer.wordpress.com/2011/05/20/modelo-de-analisis-software/>.

MySQL. *MySQL*. [En línea] 2011. [Citado el: 20 de marzo de 2014.] <http://dev.mysql.com/doc/refman/5.0/es/examples.html>.

National Instruments. [En línea] [Citado el: 8 de 4 de 2014.] <http://www.ni.com/academic/students/learnlabview/esa/debugging.htm>.

Neleste. [En línea] [Citado el: 25 de 4 de 2014.] <http://www.neleste.com/modelo-vista-controlador/>.

Olivares, Yanoski Agneri Martínez Hernández y José Rolando Lafaurie. *Sistema para la generación de reportes en la plataforma alasGRATO.* La Habana : s.n., 2008.

Orteza., Miquel Camps. Aprender a programar PHP desde cero. *Aprender a programar PHP desde cero.* [En línea] [Citado el: 15 de abril de 2014.] <http://betabeers.com/curso/aprender-programar-php-desde-cero-402/>.

PHP Framework Interop Group. [En línea] [Citado el: 8 de 4 de 2014.] <http://www.php-fig.org/psr/psr-2/es/>.

PHP Report Maker. [En línea] [Citado el: 22 de febrero de 2014.] <http://www.softpedia.es/programa-PHP-Report-Maker-65336.html>.

PHP Report Maker. [En línea] 20 de enero de 2014. [Citado el: 18 de febrero de 2014.] <http://www.hkvstore.com/phpreportmaker/>.

Php/ Java brigde. *Php/ Java brigde.* [En línea] [Citado el: 1 de noviembre de 2013.] <http://php-java-bridge.sourceforge.net/pjb/>.

REDMIC. [En línea] [Citado el: 8 de 4 de 2014.] <http://www.redmic.es/web/index.php/modelo-de-datos>.

Reportico-Open Source PHP Report Designer. [En línea] [Citado el: 17 de febrero de 2014.] <http://www.reportico.org/site/index.php>.

Ricardo Mariaca. DynamicReports. [En línea] [Citado el: 18 de febrero de 2014.] <http://www.dynamicreports.org/>.

Salomón, Omar Pérez. Los Joven Club en Fidel Castro. *Cubadebate.* [En línea] [Citado el: 1 de febrero de 2014.] <http://www.cubadebate.cu/especiales/2011/09/15/los-joven-club-en-fidel-castro/>.

Slideshare. Arquitectura de 3 Capas. [En línea] 24 de junio de 2008. [Citado el: 14 de febrero de 2014.] <http://www.slideshare.net/decimo/arquitectura-3-capas>.

SlideShare. *SlideShare.* [En línea] 25 de mayo de 2009. [Citado el: 6 de noviembre de noviembre.] <http://www.slideshare.net/ajdgeniz/aprenda-reportes-con-netbeans-y-ireport>.

Soria, Julio César. Clasificación de los reportes durante el desarrollo de sistemas. . *Clasificación de los reportes durante el desarrollo de sistemas.* . [En línea] 2005. [Citado el: 20 de febrero de 2014.] <http://lobyte.blogspot.com/2006/05/clasificacin-de-los-reportes-durante.html>.

Stackoverflow. *stackoverflow.* [En línea] 2013. [Citado el: 20 de marzo de 2014.] <http://stackoverflow.com/questions/14886781/jqplot-display-button>.

Tedeschi, Nicolás. Microsoft. [En línea] [Citado el: 26 de 4 de 2014.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.

Tipos de Información. [En línea] [Citado el: 20 de febrero de 2014.] <http://10ejemplos.com/tipos-de-informacion>.

Tutorial básico del programador web: PHP desde cero. *Tutorial básico del programador web: PHP desde cero.* [En línea] 2006. [Citado el: 4 de abril de 2014.] http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=70&Itemid=193.

Zamitz, Carlos Alberto Román. TEMAS ESPECIALES DE COMPUTACIÓN. *TEMAS ESPECIALES DE COMPUTACIÓN.* [En línea] [Citado el: 31 de marzo de 2014.] http://profesores.fi-b.unam.mx/carlos/aydoo/sequence_colaboration.html.

Glosario de términos

BD: Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Estadísticas. Ciencia que trata la recolección, clasificación y presentación de los hechos sujetos a una apreciación numérica como base a la explicación, descripción y comparación de los fenómenos.

Framework: Estructura predefinida para la creación de aplicaciones. Puede estar formado por un conjunto de librerías y clases o por una arquitectura que facilita el desarrollo de software.

UML: Lenguaje Unificado de Modelado (*Unified Modeling Language*). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

MINSAP: Ministerio de Salud Pública de la República de Cuba, es el organismo que rige el Sistema Nacional de Salud, dirige y controla la aplicación de la política del Estado y del gobierno en cuanto a la Salud Pública.

PHP: es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Reporte: Informes que organizan y exhiben la información contenida en una base de datos, cuya función es aplicar un formato determinado a los datos para mostrarlos.

Symfony: es un framework construido con varios componentes independientes creados por el proyecto Symfony, para crear aplicaciones y sitios web con PHP.

Anexos

Generar reporte personalizado de equipos médicos

Objetivo	Conocer información relacionada con los equipos médicos según los criterios de búsqueda seleccionados.	
Actores	Especialista.	
Resumen	Inicia cuando el actor accede a la opción Reportes Estadísticos en el menú inicio y presiona el icono que indica la acción Reporte personalizado de equipos médicos, el sistema brinda la posibilidad de introducir los criterios para generar el reporte, el actor introduce dichos, el sistema a partir de los criterios seleccionados y consultando las entidades correspondientes, genera el reporte de información. El actor puede imprimir el reporte, consultarlo o guardarlo según sus necesidades, el caso de uso termina.	
Complejidad	Media.	
Prioridad	Crítico.	
Precondiciones	Debe existir información en el sistema correspondiente a los equipos médicos.	
Postcondiciones	Realizado el reporte personalizado de equipos médicos se generó un informe con la documentación seleccionada por el actor.	
Flujo de eventos		
Flujo básico Generar reporte personalizado de equipos médicos		
	Actor	Sistema
1.	Accede a la opción Reporte personalizado de equipos médicos.	

2.		Brinda la posibilidad de introducir los criterios para generar el reporte.
3.	Introduce los datos que considera como criterios para generar el reporte.	
4.		<p>Permite:</p> <ul style="list-style-type: none"> • Ordenar el reporte por uno de los atributos correspondientes a equipos médicos, de manera ascendente o descendente. “Ordenar el reporte ascendente o descendientemente por un atributo.”
5.		<p>Permite exportar los resultados obtenidos. Ver Descripción de CUS “Exportar resultados a PDF.”</p> <p>En caso de que no halla datos a mostrar. Ver Flujo Alterno “No hay datos a mostrar”.</p>
6.		El caso de uso termina.
Flujos alternos		
Nº 1 “Ordenar el reporte ascendente o descendientemente por un atributo.”		
	Actor	Sistema
4	Selecciona un atributo del reporte para ordenarlo ascendente o descendientemente.	
5		Ordena el reporte ascendente o descendientemente por el atributo

		seleccionado.
6		Regresa al paso 5 del Flujo Normal de Eventos .
Nº 2 “No hay datos a mostrar.”		
	Actor	Sistema
1.	Selecciona la opción generar reporte.	
2.		Muestra un mensaje de notificación: “La búsqueda obtuvo 0 resultados”.
Relaciones	CU incluidos	
	CU extendidos	
Requisitos no funcionales		
Asuntos pendientes	No existen	

Tabla 16: Generar reporte personalizado de equipos médicos

Generar reporte gráfico estado técnico y disponibilidad técnica por unidad de subordinación nacional

Objetivo	Conocer los datos relacionados con el estado técnico y disponibilidad técnica en las instituciones de subordinación nacional.
-----------------	---

Actores	Especialista.	
Resumen	Inicia cuando el actor accede a la opción Reportes Estadísticos en el menú inicio y presiona el icono que indica la acción Reporte gráfico estado técnico y disponibilidad técnica por subordinación, el sistema brinda la posibilidad de escoger los criterios para generar el reporte, el actor toma su decisión, el sistema a partir de los criterios seleccionados y consultando las entidades correspondientes, genera el reporte gráfico. El actor puede consultarlo o guardarlo según sus necesidades, el caso de uso termina	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	Debe existir información en el sistema correspondiente a las instituciones de subordinación nacional.	
Postcondiciones		
Flujo de eventos		
Flujo básico Generar reporte gráfico estado técnico y disponibilidad técnica por unidad de subordinación nacional		
	Actor	Sistema
1.	Accede a la opción Reportes Estadísticos en el menú inicio y presiona el icono que indica la acción Reporte gráfico estado técnico por subordinación.	
2.		Muestra los tipos de subordinación existentes y permite seleccionar como criterio subordinación nacional.

3.	Selecciona la opción subordinación nacional.	
4.		Muestra las unidades de salud por subordinación nacional existentes.
5.	Selecciona las unidades de salud que desee graficar y la opción Generar Reporte.	
6.		Genera la gráfica de estado técnico y disponibilidad técnica por unidad de subordinación nacional.
7.		<p>Permite:</p> <ul style="list-style-type: none"> • A través de la gráfica generada en el eje de las x se muestran los valores correspondientes a la disponibilidad técnica en las instituciones de subordinación nacional. • Una vez generado el reporte gráfico, el sistema brinda la opción de salvar la imagen. “Guardar Imagen.” En caso de que no halla datos a mostrar. Ver Flujo Alterno “No hay datos a mostrar”.
8.		El caso de uso termina.
Flujos alternos		
Nº 1 “Guardar Imagen.”		

	Actor	Sistema
1.	Selecciona guardar imagen.	
2.		Despliega una imagen exacta de la gráfica generada.
3.	Selecciona salvar imagen presionando clic derecho.	
4.		Guarda la imagen en formato png.
Nº 2 “No hay datos a mostrar.”		
	Actor	Sistema
1.	Selecciona la opción generar reporte.	
2.		Muestra un mensaje de notificación: “La búsqueda obtuvo 0 resultados”.
Relaciones	CU incluidos	
	CU extendidos	
Requisitos funcionales	no	
Asuntos pendientes	No existen	

Tabla 17: Generar reporte gráfico estado técnico y disponibilidad técnica por subordinación nacional

Generar reporte gráfico disponibilidad técnica por provincia

Objetivo	Conocer los datos relacionados con la disponibilidad técnica en las provincias	
Actores	Especialista.	
Resumen	Inicia cuando el actor accede a la opción Reportes Estadísticos en el menú inicio y presiona el icono que indica la acción Reporte gráfico disponibilidad técnica por provincia, el cual brinda la posibilidad de escoger los criterios para generar el reporte, el actor toma su decisión, el sistema a partir de los criterios seleccionados y consultando las entidades correspondientes, genera el reporte gráfico. El actor puede consultarlo o guardarlo según sus necesidades, el caso de uso termina	
Complejidad	Alta.	
Prioridad	Crítico.	
Precondiciones	Debe existir información en el sistema correspondiente a los equipos	
Postcondiciones		
Flujo de eventos		
Flujo básico Generar reporte gráfico disponibilidad técnica por provincia		
	Actor	Sistema
1.	Accede a la opción Reportes Estadísticos en el menú inicio y presiona el icono que indica la acción Reporte gráfico disponibilidad técnica por provincia.	

2.		Brinda la posibilidad de escoger los criterios para generar el reporte.
3.	Escoge los datos que considera como criterios para generar el reporte.	
4.		<p>Permite:</p> <ul style="list-style-type: none"> • Una vez generado el reporte gráfico, el sistema brinda la opción de salvar la imagen. “Guardar Imagen.” Ver Flujo Alterno “No hay datos a mostrar”.
5.		El caso de uso termina.
Flujos alternos		
Nº 1 “Guardar Imagen.”		
	Actor	Sistema
1	Selecciona guardar imagen.	
.		
2		Despliega una imagen exacta de la gráfica generada.
.		
3	Selecciona salvar imagen presionando clic derecho.	
.		
4		Guarda la imagen en formato png.
.		
Nº 2 “No hay datos a mostrar.”		
	Actor	Sistema

1.	Selecciona la opción generar reporte.	
2.	Muestra un mensaje de notificación: “La búsqueda obtuvo 0 resultados”.	
Relaciones	CU incluidos	
	CU extendidos	
Requisitos no funcionales		
Asuntos pendientes	No existen	

Tabla 18: Generar reporte gráfico disponibilidad técnica por provincia

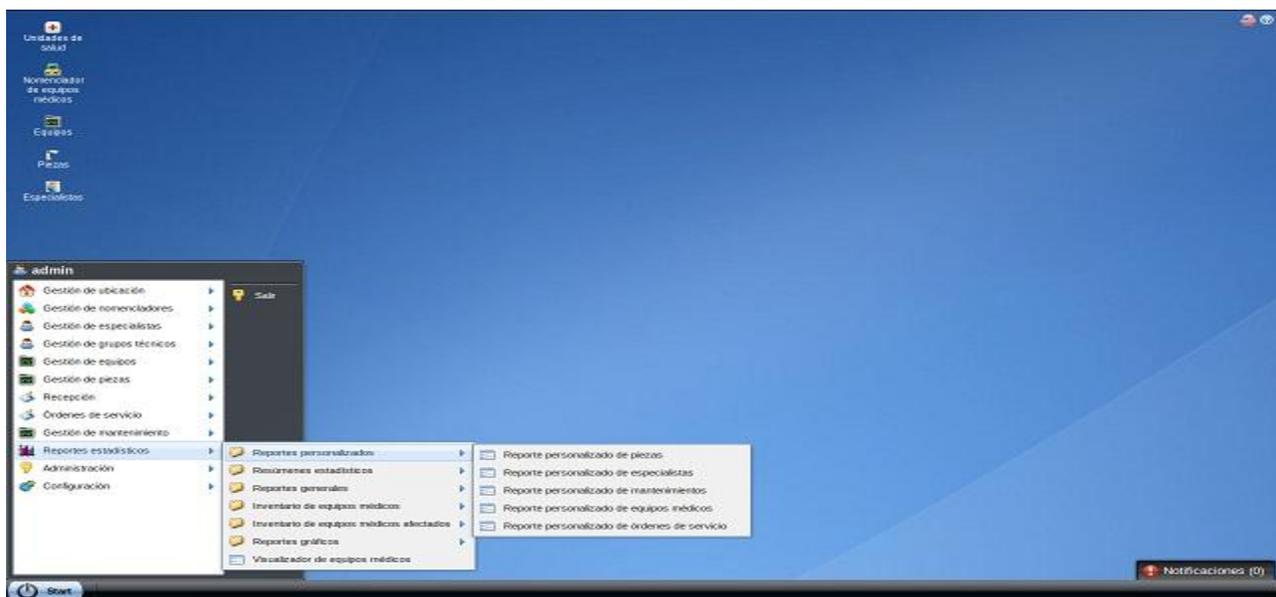


Figura 20: Interfaz Principal módulo Reportes Estadísticos v2.0.

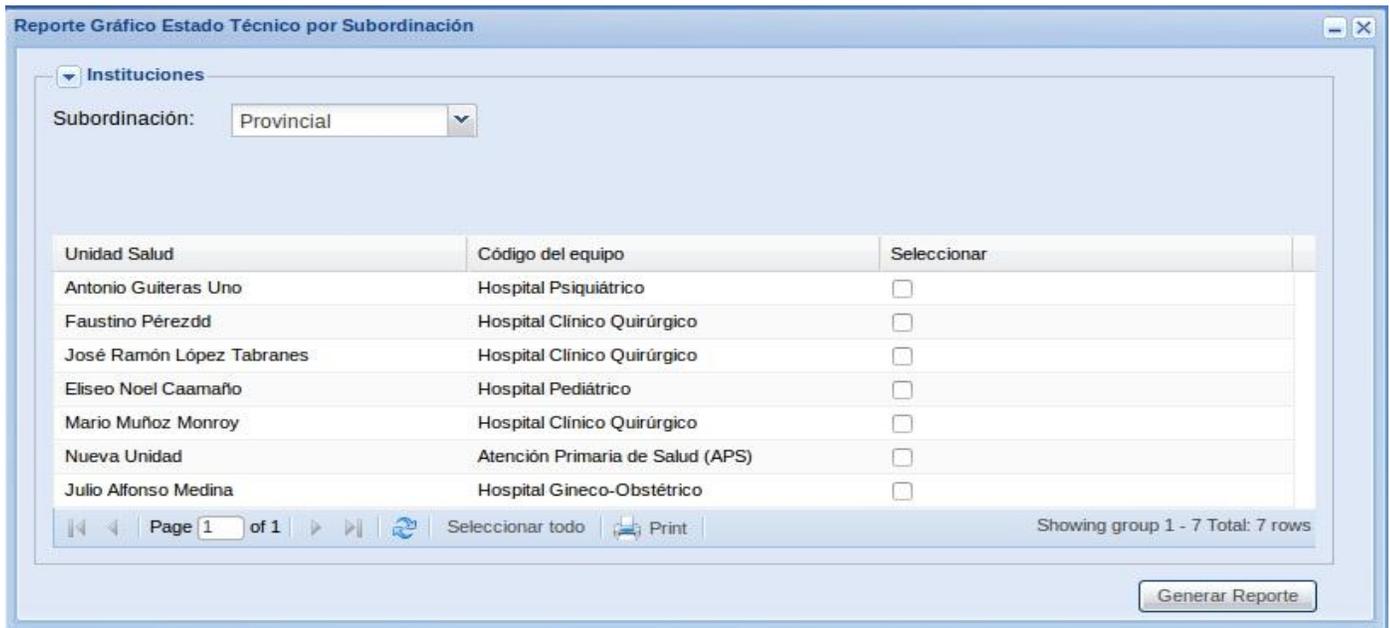


Figura 21: Interfaz Reporte Gráfico Estado Técnico por Subordinación

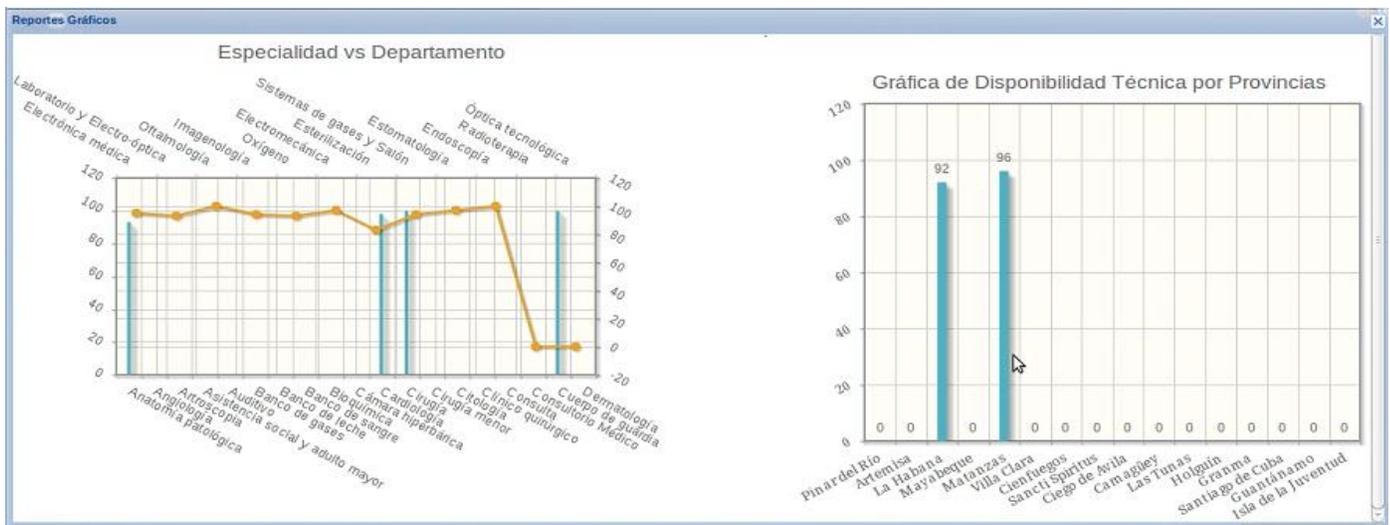


Figura 22: Interfaz Reportes Gráficos

Guía de Observación. Aspectos a tener en cuenta:

Objetivo: Decidir que mecanismo incorporar al módulo Reportes Estadísticos v2.0 para utilizar las funciones gráficas en los reportes.

Escala de Evaluación		
Mal	Regular	Bien

Condiciones de trabajo:

- Puesto de trabajo (mesas, sillas).
- Computadoras con accesos a Internet para documentarse en caso necesario.
- Climatización del laboratorio.
- Iluminación.
- Limpieza y organización.

Relaciones interpersonales:

- Entre los miembros del departamento y los estudiantes involucrados en el trabajo de diploma.

Dinámica de trabajo:

- Necesidad de la integración en el nuevo módulo de un mecanismo que erradique las deficiencias existentes
- Escoger que tipo de herramienta es el más factible para añadir a la aplicación
- Estudio en el ámbito internacional y nacional de los sistemas que poseen implícitos mecanismos con funciones gráficas.
- Ventajas y desventajas a mediano y largo plazo de la selección realizada.
- Resultados preliminares de la elección desarrollada.