

**Universidad de Sancti Spíritus “José Martí Pérez”
Facultad de Ingeniería
Carrera Ingeniería Informática**



Título: “Aplicación Web para la gestión y control de datos estadísticos de enfermedades en el Centro de Higiene y Epidemiología en Yaguajay”

Trabajo de diploma para optar por el título de Ingeniería Informática.

Tutor: Ing. Nelsy Muro.

Consultantes: Ing. Pablo Manuel Sánchez Portal.
Lic. Oscar Ulises Melean.

Autor: Omar Estrada Díaz

Junio, 2014

Pensamiento

"He sido un hombre afortunado en la vida: nada me resultó fácil".

Sigmund Freud

Dedicatoria

Quiero dedicar la realización de este trabajo a las personas más importantes que he conocido en toda mi vida: A mi Mamá y a mi Papá, por ser ejemplos bien definidos, guías a las buenas maneras y sobre todo: por enseñarme a conducirme por mí mismo. A mi familia toda: por ser única e irrepetible y a mis amigos por hacerse notar. A todos mis seres queridos. De corazón... Gracias a todos.

Agradecimientos

*Para estar en paz con todos, de una manera sencilla.
Dios mediante quiero agradecer:*

*A mi Mamá, por alimentarme el alma y el espíritu TODOS los días.
A mi Papá por ayudarme a llegar temprano siempre en la vida, sabiendo siempre algo nuevo.*

Al grupo de graduados en el 2014 de Ingeniería Informática, por su apoyo académico y en la vida diaria. Especialmente al grupo de estudio que me enseñó a encontrarle solución a todos los problemas.

A mi compañero de aula, a mi copiloto de viajes y mi socio en los trabajos más difíciles durante muchos años Pedro Alexis López Valdivia.

A los compañeros de mi residencia por compartir en los buenos y malos momentos de la vida universitaria, siempre con una sonrisa en la cara.

A mi tutora Nelsy Muro por darme apoyo, seguridad y oportunidad de llegar hasta el final.

A mis profesores, por permitirme aprender durante estos cinco largos años, y los compañeros recién graduados por aplacar mis dudas, especialmente a Pablo Manuel Sánchez Portal, a Carlos y a Ernesto por darme los minutos que no tenían.

A todos los que han trabajado en el JCCE de Yaguajay por haberme enseñado el mundo de la informática.

A todos los que conocí durante este tiempo, y por quedarse en mi memoria por su manera de ser.

Спасибо товарищи.

Resumen

Las Tecnologías de la Información y las Comunicaciones juegan un rol esencial en los procesos de gestión de la información, permitiendo a las organizaciones tener un control efectivo en su sistema. La presente investigación titulada “*Aplicación Web para la gestión y el control de datos estadísticos*”, se realizó en El Centro de Higiene y Epidemiología de Yaguajay que actualmente presenta problemas en cuanto a la gestión y control sobre los datos estadísticos de las distintas enfermedades que se reconocen en los pacientes de nuestro país. Por esta razón el objetivo de esta investigación es desarrollar una Aplicación Web que contribuya a la gestión y control de datos estadísticos de estas enfermedades. La elaboración del sistema se llevó a cabo utilizando el Proceso Unificado de Desarrollo (RUP), utilizando como herramienta de análisis y diseño: Visual Paradigm. Para la implementación del sistema se utilizó como lenguaje de programación PHP para obtener el máximo rendimiento de y aprovechar todas sus características; para la persistencia de los datos se utilizó PostgreSQL 8.4 y como servidor Web se utilizó el WampServer. Obteniendo como resultado una aplicación Web para contribuir a la gestión y control de datos estadísticos, que se caracteriza por la consistencia, facilidad de acceso y seguridad de la información, presenta una interfaz con un ambiente amigable, legible y de fácil navegación, permitiendo que toda la información se encuentre almacenada en una base de datos, lo cual proporciona información actualizada y detallada relacionada con la gestión del registro de las enfermedades.

Abstract

The Information Technology and Communications play an essential role in the processes of information management, enabling organizations to have effective control on your system. This research entitled "Web Application management and control statistics", was made at The Centre of Hygiene and Epidemiology at Yaguajay city. It's currently presents problems in terms of management and control over the statistics of the various diseases recognized in the patients in our country. Therefore the objective of this research is to develop a Web application that contributes to the management and control of these diseases statistics. In developing the system were carried out several stages which were raised by the Unified Development Process (RUP), using as a tool for analysis and design: Visual Paradigm. To implement the system was used as the programming language PHP to maximize performance and take advantage of all its features; for data persistence and PostgreSQL 8.4 was used as the Web server was used WampServer. Resulting in a Web application to help manage and control statistics, which is characterized by consistency, ease of access and information security, presents an interface with a friendly, readable and easily navigable environment, allowing all the information is stored in a database, which provides updated and detailed management related registry diseases were described by municipal hospitals by the reports that are generated information and also has a system of traces contributes to the consistency and reliability of the information.

Índice

Introducción	1
Capítulo I:	7
1.1 Conceptos básicos asociados al dominio del problema	7
1.2 Gestión	7
1.2.1 ¿Qué es un sistema de gestión?	8
1.3 Control.....	8
1.3.1 Control de datos estadísticos.....	8
1.3.2 Según varios autores:	9
1.3.3 Tipos de Controles:	10
1.4 Definición de Aplicación Web.	11
1.4.1 Ventajas de las aplicaciones web:.....	12
1.4.2 Desventajas.....	12
1.5 Definición de Sitio Web:	13
1.5.1 Ventajas:	13
1.5.2 Desventaja:	14
1.5.3 Otras ventajas:	15
1.5.4 Otras Desventajas:	16
1.6 Tecnologías del lado del cliente	17
1.6.1 HTML.....	17
1.6.2 JavaScript.....	17
1.6.3 CSS	17
1.7 Tecnologías del lado del servidor	18
1.7.1 CGI.....	18
1.7.2 PHP	18
1.7.3 ASP.....	18
1.7.4 JSP.....	18
1.8 Servidores Web.....	19
1.8.1 Apache	19
1.9 Sistemas gestores de base de datos	20
1.9.1 Base de datos (BD):	20
1.9.2 MySQL	20
1.9.3 PostgreSQL.....	21
1.10 Fundamentación de la metodología utilizada:	22
1.10.1 RUP	23
1.10.2 UML.....	23
1.11 ¿Por qué utilizar PHP?.....	24
1.11.1 Velocidad:	24
1.11.2 Estabilidad:	24
1.11.3 Simplicidad:	25
1.12 ¿Por qué utilizar HTML?	26
1.13 ¿Por qué utilizar JavaScript?	26
1.14 ¿Por qué utilizar CSS?	26
1.15 ¿Por qué utilizar PostgreSQL?.....	27
1.16 ¿Por qué utilizar Apache?	27
1.17 Herramientas de desarrollo:.....	28
1.18 Visual Paradigm:	28
1.18.1 Ventajas:	28
1.18.2 Desventajas:	30
1.19 Modelo 2 Capas	30

1.19.1	<i>Desventajas del sistema de dos capas:</i>	30
1.20	Modelo Vista Controlador	31
1.20.1	<i>¿Qué ventajas tiene el uso del patrón MVC?</i>	31
1.20.2	<i>¿TIENE DESVENTAJAS EL MVC?</i>	32
1.21	Bibliotecas	32
1.21.1	<i>FPDF</i>	32
1.21.2	<i>Highcharts</i>	33
1.22	Definición de software libre	34
1.22.1	<i>Características del software libre</i>	35
1.22.2	<i>Diferencias con el open source (código abierto)</i>	38
1.22.3	<i>Pros y contras del uso de software libre</i>	38
1.22.4	<i>Comunidades de desarrollo de software libre</i>	39
1.22.5	<i>Impacto del software libre</i>	39
1.23	Conclusiones	41
Capítulo 2		42
2.1	Introducción:	42
2.2	Modelo del negocio	42
2.2.1	<i>Modelado del negocio</i>	43
2.2.2	<i>Reglas del negocio a considerar</i>	43
2.2.3	<i>Actores del Negocio</i>	43
2.2.4	<i>Trabajadores del negocio</i>	44
2.3	Diagrama de Casos de Uso del Negocio	44
2.3.1	<i>Descripción de los Casos de Uso del negocio</i>	44
2.4	Diagrama de actividades	46
2.5	Modelo de objetos del negocio	46
2.6	Especificación de los requisitos de software.	47
2.6.1	<i>Requerimientos funcionales.</i>	47
2.6.2	<i>Requerimientos funcionales:</i>	47
2.6.3	<i>Requerimientos no funcionales</i>	50
2.7	Descripción del sistema propuesto	53
2.7.1	<i>Modelo de casos de uso del sistema</i>	53
2.7.2	<i>Actores del sistema</i>	53
2.7.3	<i>Casos de uso del sistema</i>	54
2.8	Paquetes	54
2.8.1	<i>Diagrama de casos de uso del sistema por paquetes</i>	55
2.8.2	<i>Descripción de los casos de uso del sistema agrupados por paquetes</i>	55
2.8.3	<i>Seguridad</i>	55
2.8.4	<i>Gestión.</i>	58
2.8.5	<i>Reportes</i>	64
2.9	Diagramas de paquetes del sistema	70
2.10	Conclusiones:	70
Capítulo 3		71
3.1	Diagrama de clases del diseño.	71
3.2	Diseño de la base de datos.	73
3.2.1	<i>Diagrama de clases persistentes</i>	74
3.3	Modelo de datos	74
3.4	Principios de diseño	74
3.4.1	<i>Estándares en la interfaz de la aplicación.</i>	74
3.4.2	<i>Formato de reportes.</i>	75
3.5	Concepción general de la ayuda	75

3.5.1	<i>Tratamiento de errores</i>	75
3.6	Seguridad	76
3.6.1	<i>Estándares de codificación</i>	76
3.7	Modelo de despliegue	76
3.8	Modelo de implementación	77
3.9	Componente	78
3.9.1	<i>Descripción del modelo de componentes</i>	78
3.10	Conclusiones	79
	<i>Conclusiones</i>	80
	<i>Bibliografía</i>	81
	Anexos	1

Introducción

El interés de conocer y registrar el índice de aparición y efecto sobre la humanidad de las enfermedades nace de una necesidad y preocupación constante en las personas a través de los siglos, por registrar y analizar con cifras o datos estadísticos, el comportamiento de la morbilidad y mortalidad de la población; según el conocimiento de su frecuencia, distribución por regiones, grupos de edades, sexo y características socioeconómicas asociadas a cambios en el tiempo, con el fin de investigar los factores que causan las enfermedades y las muertes. Además de evaluar la situación de salud de la población en diversos períodos, así como los resultados de los programas de salud aplicados en los diferentes países. También planear los servicios de salud que requieren la población aplicando medidas para prevenir las enfermedades y muertes prematuras que existían mundialmente.

Creación

Con tal propósito o fin en una reunión efectuada en Viena en 1891 el Instituto Internacional de Estadística, sucesor del Congreso Internacional de Estadística, encargó a un comité dirigido por el sociólogo francés Jacques Bertillon, jefe de los Servicios de Estadística de la ciudad de París, la preparación de una clasificación de causas de defunción, que se aceptara universalmente. Este comité constituido por ilustres estadistas como William Farr, de Inglaterra, Marc D'Espine, de Suiza y el propio Jacques Bertillon, de Francia, comenzarían a trabajar juntos para que en 1893 se adoptara, por primera vez, una clasificación de causas de muerte de uso internacional.

Aplicación

Su primera aplicación fue en América del Norte y ya en 1898 la Asociación Estadounidense de Salud Pública, recomendó que los registros civiles de Canadá, México y Estados Unidos, adoptaran la Clasificación de Bertillon y sugirió además su revisión cada 10 años. El Gobierno de Francia convocó en París en Agosto de 1900, la primera Conferencia Internacional para la revisión de la Lista de Bertillon o Lista Internacional de Causas de defunción, a donde

asistieron delegados de 26 países y el 21 de Agosto de 1900 se adoptó una clasificación detallada de las causas de defunción, que comprendía 179 grupos y una clasificación abreviada que comprendía 35 grupos.

Vigencia en Cuba

La Décima Revisión de la Clasificación Estadística Internacional de Enfermedades y Problemas Relacionados con la Salud, vigente en Cuba desde el primero de Enero del año 2000, gracias a la relación entre la Organización Mundial de la Salud, Organización Panamericana de la Salud, el MINSAP y los logros alcanzados por Cuba en la Medicina ante el resto del Mundo.

Aplicaciones Informáticas de CIE (Clasificación Integral de Enfermedades)

La Organización internacional de Salud (OMS) editó en formato CD-ROM, toda la información de los 3 volúmenes, esta multimedia respeta la estructura de la clasificación y permite realizar búsquedas por diferentes criterios, devuelve la respuesta deseada en el caso de acertada la búsqueda. Esta propuesta presenta algunos inconvenientes como la poca compatibilidad con otros sistemas que puedan hacer uso de sus servicios, la negativa de no poder realizar las actualizaciones de su información a partir de los nuevos cambios que se dicten por la OMS. La solución existente funciona como un libro de consulta y no como una solución informática que permita la integración con otros módulos de salud que necesitan la CIE para gestionar sus procesos informativos. En la estrategia de informatizar el sector de la salud en Cuba, se priorizan áreas claves en la atención médica fundamentalmente de la atención primaria. Ésta en su desenvolvimiento aplica terminologías que no se encuentran en la clasificación internacional. La existencia de la aplicación: Registro de la Clasificación Internacional de Enfermedades y Problemas Relacionados con la Salud (RCIE) la cual está basada en la CIE, sirve como punto de partida para el desarrollo de otras aplicaciones de gestión de información relacionadas con el diagnóstico (Pública, 2013)

Según un informe del Departamento de Estado de los Estados Unidos, en 1957 Cuba ocupó el tercer lugar en América Latina y estaba por delante de países europeos como el Reino Unido o Finlandia en términos de médicos y

odontólogos per cápita. El informe también indica que: "Desgraciadamente, el anuario estadístico de la ONU ya no publica estas estadísticas, lo que no permite comparar al sistema de 1957 con el actual; no obstante, es completamente erróneo caracterizar a la Cuba pre-revolucionaria como un país atrasado en cuanto a la provisión de asistencia sanitaria".

El Banco Mundial en 1950 (época pre-revolucionaria) observó: "Las enfermedades no son un problema serio en Cuba, pero la salud lo es". La mayor parte de los niños de los campos sufrían de parásito intestinal y como la mitad de todos los cubanos se clasificaban de bajo nivel nutricional (hambre). El 60% de los doctores y 80% de las camas de los hospitales estaban en La Habana, nada más había un hospital en Cuba en zonas del campo, 4 de 5 trabajadores campesinos recibían atención médica si la pagaban, que indica que la mayor parte no tenía acceso a los cuidados de salud.

A partir del triunfo de la Revolución en 1959 y la estatalización de la salud, se aplicaron fuertes políticas de innovaciones y ampliación en este campo. En 1958, el país contaba únicamente con 826 enfermeras y auxiliares de enfermería, mientras que en la actualidad la cifra alcanza los 84.232 especialistas, es decir, 101 enfermeras y auxiliares por cada uno que había entonces. En 1959, había solo 8 modestas escuelas de enfermería, mientras que en la actualidad éstos han pasado a ser 24 politécnicos de la salud donde se forman no solo enfermeras, sino también doctores, fisioterapeutas y demás trabajadores relacionados con las tecnologías de la salud. Entre 1980 y 1985, se fundaron instituciones como el Centro de Ingeniería y Biotecnología, el Centro Nacional de Investigaciones Científicas, el Cardiocentro de Cirugía Infantil William Soler (mayor del mundo), el Centro de Inmunoensayo, el Centro de Trasplantes y Regeneración del Sistema Nervioso. Se logró crear un equipo de resonancia magnética con sistema Evalimage para la visualización y análisis termográfico de imágenes y el bisturí láser cubano. En Cuba se comenzaron a realizar trasplantes de riñón hígado, corazón y corazón-pulmón. También se efectuaron aportes a la medicina como la vacuna contra la meningitis meningocócica, el interferón alfa leucocitario humano, el descubrimiento de una sustancia que cura el vitíligo y la obtención del factor de crecimiento epidérmico.

Pero en el caso de prevenir y estudiar el efecto de las enfermedades en la población cubana se crea lo que se conoce como Centros de Higiene y Epidemiología, centro capaz de analizar el comportamiento de las enfermedades a partir de su historial, dado su corredor epidemiológico. Lo que brinda a las autoridades sanitarias la posibilidad de tomar las medidas para combatirlas y erradicarlas, además de su carácter preventivo para evitar males mayores o simplemente su aparición. Nuestro país se ha ido insertando en el desarrollo y utilización de las TIC en la salud, cultura, economía y la sociedad de forma general, potenciando sobre todo la utilización de la informática como herramienta fundamental en la gestión de la información, debido a la eficiencia que las mismas ofrecen en los procesos de las organizaciones. Entre las instituciones interesadas en su utilización está **El Centro de Higiene y Epidemiología** de Yaguajay perteneciente a la provincia de Sancti Spiritus, Cuba. A partir de la necesidad de llevar el control de las incidencias de los casos de las enfermedades que se presentan en nuestro país, el Estado cubano crea lo que se denomina como Centros de Higiene y Epidemiología, con el propósito de poder determinar el origen de las enfermedades y las medidas pertinentes para su eliminación, en ese momento es que el departamento de Centro de Análisis y Cálculo se encarga de llevar un control sobre el número de casos que se registran en los centros hospitalarios, en este se realizan un conjunto de cálculos estadísticos que permiten ver la evolución de las enfermedades, así como en determinado momento poder emitir un pronóstico de su comportamiento, a partir de un boletín semanal se informa a las autoridades pertinentes la evolución de las mismas, además de dar una alerta relacionada con determinado índice, lo que permite saber el grado de peligrosidad de las mismas, siendo definidas estas como éxito, seguridad o alerta. En este caso todos estos cálculos se realizan manualmente, con la dependencia de registros en formatos duros, lo que hace engorroso todo el proceso antes mencionado.

De acuerdo con lo expresado anteriormente, el desarrollo de este trabajo está basado en darle solución al siguiente **problema científico**:

¿Cómo contribuir la gestión y control de datos estadísticos de enfermedades en el Centro de Higiene y Epidemiología en Yaguajay?

A partir del problema se define como **objetivo general**: *Desarrollar una aplicación web para la gestión y control de datos estadísticos de enfermedades en el Centro de Higiene y Epidemiología en Yaguajay.*

Para cumplir el objetivo general se realizan las siguientes **Preguntas Científicas**:

- ¿Qué fundamentos teóricos y metodológicos sustentan la construcción de una aplicación web que contribuya a la gestión control de datos estadísticos de enfermedades en el Centro de Higiene y Epidemiología?
- ¿Cómo diseñar una aplicación web que contribuya a la gestión y control de datos estadísticos de enfermedades en el Centro de Higiene y Epidemiología en Yaguajay?
- ¿Cómo implementar una aplicación web que contribuya a la gestión y control de datos estadísticos de enfermedades en el Centro de Higiene y Epidemiología en Yaguajay?

Para dar respuestas a las preguntas científicas se plantean las siguientes

Tareas de Investigación:

- Determinar que fundamentos teóricos y metodológicos sustentan la construcción de una herramienta informática para a la gestión de datos estadísticos.

- Diseñar una aplicación web que contribuya la *gestión* y control de datos estadísticos de enfermedades en el Centro de Higiene y Epidemiología en Yaguajay.
- Implementar una aplicación Web que contribuya la *gestión* y control de datos estadísticos de enfermedades en el Centro de Higiene y Epidemiología en Yaguajay

Estructura del Documento

Una introducción, para la presentación del trabajo y orientar al lector sobre los aspectos fundamentales, donde se explica con absoluta claridad la situación problemática, problema a resolver, objeto de estudio, campo de acción, objetivo general y preguntas científicas.

Capítulo 1: Fundamentación teórica y metodológica que sustentan la elaboración de una Aplicación Web que contribuya a la gestión y control de datos estadísticos de enfermedades en el Centro de Higiene y Epidemiología en Yaguajay. Aborda los principales conceptos asociados al dominio del problema, aspectos teóricos-conceptuales que fundamentan la creación de una Aplicación Web para el Centro de Higiene y Epidemiología. Se describe el objeto de estudio también incluye un estudio sobre las principales tendencias, metodologías y tecnologías que se usan en la actualidad y las que pueden usarse para la solución del problema. AKI

Capítulo 2. Diseño de un Aplicación Web que contribuya a la gestión y control de datos estadísticos de enfermedades en el Centro de Higiene y Epidemiología en Yaguajay: Se describe el modelo del negocio, identificando los procesos involucrados en él y las reglas que lo caracterizan. Se realiza la descripción del modelo de casos de uso, identificando y describiendo los actores, trabajadores y casos de uso del sistema mediante el diagrama de casos de uso.

Capítulo 3. Implementación de una Aplicación Web que contribuya a la gestión y control de datos estadísticos de enfermedades en el Centro de Higiene y Epidemiología en Yaguajay: Se plantean los detalles relacionados con la implementación del sistema que se propone a través de los diagramas

de clases, el diagrama de clases persistente y el modelo de datos. Se especifican los principios tenidos en cuenta para el diseño de la interfaz de la herramienta propuesta, y la implementación mediante los modelos de despliegue y componentes.

Luego aparecen las conclusiones, las recomendaciones, la bibliografía y los anexos, que se utilizaron para ilustrar este trabajo.

Capítulo I:

Fundamentación teórica y metodológica que sustentan la elaboración de una Aplicación Web que contribuya a la gestión y control de datos estadísticos de enfermedades en el Centro de Higiene y Epidemiología en Yaguajay.

Aborda los principales conceptos asociados al dominio del problema, aspectos teóricos-conceptuales que fundamentan la creación de una Aplicación Web para el Centro de Higiene y Epidemiología. Se describe el objeto de estudio también incluye un estudio sobre las principales tendencias, metodologías y tecnologías que se usan en la actualidad y las que pueden usarse para la solución del problema.

1.1 Conceptos básicos asociados al dominio del problema

1.2 Gestión

Del latín *gestiō*, el concepto de **gestión** hace referencia a la **acción y a la consecuencia de administrar o gestionar algo**. Al respecto, hay que decir que gestionar es **llevar a cabo diligencias** que hacen posible **la realización de una operación comercial o de un anhelo cualquiera**. Administrar, por otra parte, abarca las ideas de **gobernar, disponer, dirigir, ordenar u organizar** una determinada cosa o situación. (Valda, 2012)

La noción de gestión, por lo tanto, se extiende hacia el conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto. La gestión es

también la **dirección** o admiración de una compañía o de un **negocio**. (Valda, 2012)

1.2.1 *¿Qué es un sistema de gestión?*

Un Sistema de Gestión es un conjunto de etapas unidas en un proceso continuo, que permite trabajar ordenadamente una idea hasta lograr mejoras y su continuidad.

Se establecen cuatro etapas en este proceso, que hacen de este sistema, un proceso circular virtuoso, pues en la medida que el ciclo se repita recurrente y recursivamente, se logrará en cada ciclo, obtener una mejora.

Las cuatro etapas del sistema de gestión son:

1. Etapa de Ideación
2. Etapa de Planeación
3. Etapa de Implementación
4. Etapa de Control

(Vergara, 2011)

1.3 *Control*

La palabra "control", es un galicismo que indica la acción o efecto de controlar. Controlar es un verbo activo que significa: "Examinar y censurar con autoridad suficiente, formando juicio, aprobando o corrigiendo lo que no estuviera en forma o modo debido".

Otra definición dice: "Es el conjunto de actividades que se emprenden, para medir y examinar los resultados obtenidos en el periodo, para evaluarlos y para decidir las medidas correctivas que sean necesarias".

1.3.1 *Control de datos estadísticos.*

El mismo es la aplicación de métodos estadísticos para analizar datos y para estudiar y vigilar la habilidad y el desempeño de un proceso. El control estadístico del proceso se basa en un concepto simple, un diagrama histórico de frecuencia, que consiste en un diagrama que muestra la frecuencia con que ocurren distintos valores o mediciones en los resultados de un proceso. (Borrego, 2010)

El método general es prescriptivo y descriptivo, no es analítico. Al controlar estadísticamente los procesos no se trata de moldear la distribución de datos reunidos en un proceso dado. Lo que se trata es de controlar el proceso con ayuda de reglas de decisión que localicen discrepancias apreciables entre los datos observados y las normas del proceso que se controla. Se dice que un proceso está bajo control estadístico cuando sólo se producen variaciones debidas a causas comunes. En otras palabras el objetivo y razón de ser control Estadístico de Procesos es ayudar a identificar las causas especiales que producen variaciones en el proceso y suministrar información para tomar decisiones.

1.3.2 Según varios autores:

- 1- Es la regulación de las actividades, de conformidad con un plan creado para alcanzar cierto objetivo. (Robert Eckles, 2010).
- 2- El proceso para determinar lo que se está llevando a cabo, valorizándolo y, si es necesario, aplicando medidas correctivas, de manera que la ejecución se desarrolle de acuerdo con lo planeado. (Terry, 2010).
- 3- Consiste en verificar si todo ocurre de conformidad con el plan adoptado, con las instrucciones emitidas y con los principios establecidos. Tiene como fin señalar las debilidades y errores a fin de rectificarlos e impedir que se produzcan nuevamente. (Fayol, 2010)
- 4- Implica la medición de lo logrado en relación con el estándar y la corrección de las desviaciones, para asegurar la obtención de los objetivos de acuerdo con el plan. (Koontz, 2010)

A partir de los criterios mostrados se concluye que a los requerimientos de la investigación el concepto dado por Fayol es que más se ajusta por ser el más abarcador y preciso.

1.3.3 *Tipos de Controles:*

Según (Flanklin, 2013) expone tres tipos de control:

- Control preliminar, este tipo de control tiene lugar antes de que principien las operaciones e incluye la creación de políticas, procedimientos y reglas diseñadas para asegurar que las actividades planeadas serán ejecutadas con prioridad. En vez de esperar los resultados y compararlos con los objetivos es posible ejercer una influencia controladora limitando las actividades por adelantado.
- Control concurrente, este tipo de control tiene lugar durante la fase de la acción de ejecutar los planes e incluye la dirección, vigilancia y sincronización de las actividades según ocurran, en otras palabras, pueden ayudar a garantizar que el plan será llevado a cabo en el tiempo específico y bajo las condiciones requeridas.
- Control de retroalimentación, este tipo de control se enfoca sobre el uso de la información de los resultados anteriores para corregir posibles desviaciones futuras de estándares aceptables. Este tipo de control implica que se han reunido algunos datos, se han analizado y se han regresado los resultados a alguien o a algo en el proceso que se está controlando de manera que puedan hacerse correcciones.

El control es de vital importancia dado que, según (Antonio, 2013):

- 1- Establece medidas para corregir las actividades, de tal forma que se alcancen planes exitosamente.
- 2- Se aplica a todo: a las cosas, a las personas, y a los actos.
- 3- Determina y analiza rápidamente las causas que pueden originar desviaciones, para que no se vuelvan a presentar en el futuro.
- 4- Localiza a los lectores responsables de la administración, desde el momento en que se establecen medidas correctivas.
- 5- Proporciona información acerca de la situación de la ejecución de los planes, sirviendo como fundamento al reiniciarse el proceso de planeación.
- 6- Reduce costos y ahorra tiempo al evitar errores.

Su aplicación incide directamente en la racionalización de la administración y consecuentemente, en el logro de la productividad de todos los recursos de la empresa.

1.4 Definición de Aplicación Web.

Son muchas las definiciones que se pueden encontrar de sobre Aplicación Web, se trata de poner las más importantes dada su terminación y maneras de exponer el contenido del mismo. En la ingeniería software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web, y en la que se confía la ejecución de la aplicación al navegador. (SUPERIOR, Definición Y Tipos de Aplicaciones Web., 2010)

Son aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador. En concreto, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web. (Bardallo, Aplicaciones Web, 2012)

Aplicación web según (Celaya, 2006). Ver anexo # 1.

Las aplicaciones web son soluciones informáticas o como su nombre lo dice “aplicaciones” que nos permiten interactuar con la información y a las cuales podemos acceder a través de una conexión a internet, desde cualquier lugar del mundo, sin necesidad de instalarlas previamente en nuestra pc; solamente necesitamos contar con un navegador web, llámese Mozilla, Internet Explorer, Chrome (Vera, 2010)

En este caso se usa como referencia el concepto dado por el MINISTERIO DEL PODER POPULAR PARA LA EDUCACIÓN SUPERIOR, ya que es implícito a la hora de dar una denominación más acercada al concepto Aplicación Web.

1.4.1 *Ventajas de las aplicaciones web:*

- Al ejecutarse a través de los navegadores, se puede acceder a ellas a través de cualquier computadora en la que se cuente con internet o se encuentre conectada a una intranet.
- Desde el punto de vista del usuario, no es necesario instalar ningún software en la computadora, por lo que no hay que preocuparse por costos de licencias o actualizaciones.
- Las actualizaciones las realiza el desarrollador en su servidor y por ende cada vez que nos conectemos tendremos la última versión disponible.
- No hay incompatibilidades con los sistemas operativos porque todo se maneja en el navegador.
- No ocupan espacio en el disco duro porque se ejecutan a través de la web.
- Nos consumen pocos recursos de hardware porque las tareas se realizan en otro ordenador.

(Vera, 2010)

En el caso del centro donde se desarrolla la aplicación existen las condiciones óptimas y necesarias para usarla como la opción de mejor desempeño a la hora de ayudar en su propósito.

1.4.2 *Desventajas*

- Las aplicaciones web requieren navegadores web totalmente compatibles para funcionar. Incluso muchas veces requieren las extensiones apropiadas y actualizadas para operar
- Muchas veces requieren una conexión a internet para funcionar, si la misma se interrumpe, no es posible utilizarla más. De todas maneras, en ocasiones, pueden ser descargadas e instaladas localmente su uso offline.
- Muchas no son de código abierto, perdiendo flexibilidad.
- La aplicación web desaparece si así lo requiere el desarrollador o si el mismo se extingue. Las aplicaciones tradicionales, en general, pueden seguir usándose en esos casos.
- El usuario, en general, no tiene libertad de elegir la versión de la

aplicación web que quiere usar. Un usuario podría preferir usar una versión más antigua, hasta que la nueva sea probada.

- En teoría, el desarrollador de la aplicación web puede rastrear cualquier actividad que el usuario haga. Esto puede traer problemas de privacidad. (Pacheco, 2013)

1.5 Definición de Sitio Web:

Un sitio web es un conjunto de páginas web ordenadas jerárquicamente bajo una misma dirección web (URL). Este grupo de páginas se entrelazan entre sí por medio de los denominados Hipervínculos — también conocidos como links o enlaces — lo que en resumen da como resultado un sitio web. (Karim, Sitios Web, 2011)

Un sitio web debe ser entendido primero como un medio de comunicación. Y en segundo lugar como una secuencia programática que se genera y visualiza mediante un computador. Si entendemos a un sitio web como un medio de comunicación debemos compararlo más con un canal de televisión que con una hoja impresa. Si se explotan correctamente las capacidades de Internet, su sitio web debería transformarse en la sucursal virtual de su empresa o institución. (Cuesta, 2011)

Básicamente, los sitios web se componen y estructuran de la siguiente manera: Ver Anexo # 2. Según (Suez, 2013).

A pesar de la pequeña diferencia que existe entre las definiciones su toma como punto de vista investigativo de este trabajo el concepto de Gabriel Cuesta (Cuesta, 2011) por la manera que enfoca el uso del Sitio Web.

1.5.1 Ventajas:

- 1- Administración sencilla.** Para gestionar el contenido de su sitio web puede hacer uso de un Panel de Administración que en comparación con el resto de los demás CMS resulta uno de los más sencillos que existen
- 2- Edición Wysiwig** Es una manera ágil y sencilla de editar o crear nuevo contenido. Wysiwig es "lo que ve es lo que escribe". Con dicho editor

puede escribir su artículo con el tipo de letra que desee, asignar color, medidas, tablas, insertar imágenes flotantes, etc.

- 3- **Editor HTML.** Si el editor Wysiwig es insuficiente hay la opción de escribir código HTML directamente donde puede hacer uso de la mayoría de etiquetas permitidas en XHTML y HTML5, posiciones relativas, absolutas, etc.
- 4- **Facilidad para insertar script externos.** Por ejemplo plugins de JQuery, Addthis, botones Facebook, twitter y otros
- 5- **Multiusuario.** Es decir muchas personas pueden gestionar el contenido del sitio web con niveles de permisos y privilegios
- 6- **Plantillas.** Puede cambiar la apariencia si perder el contenido de la página con solo cambiar de plantilla.
- 7- **Facilidad de plantillas.** La forma de crear una plantilla es relativamente más sencilla que la ofrecida por otros CMS

1.5.2 *Desventaja:*

- 1- **Actualización constante.** El administrador de la página web siempre debe estar atento a las actualizaciones disponibles. Requiere que el sitio esté constantemente actualizado para evitar agujeros o vulnerabilidades que siempre surgen ya que el código fuente es de dominio público. Afortunadamente siempre hay toda una comunidad detrás así como el propio equipo Wordpress por lo cual siempre hay publicaciones de parches o correcciones al aplicativo.
- 2- **Está orientado más a blog.** Para un sitio web que tendrá pocas páginas quizás no es la mejor alternativa, ya que uno de los aspectos con mayor fuerza de Wordpress es la publicación de artículos, opiniones, ideas, críticas, o simplemente de temáticas de diversa índole. Un sitio web hecho en Wordpress es aconsejable que tenga una sección de publicaciones para sacar el mejor provecho de Wordpress.
- 3- **Acabado diferente, edición HTML.** Para darle a una página web hecha en Wordpress un acabado diferente con bloques flotantes se debe hacer uso de CSS y HTML el editor WYSIWIG resulta limitado.
- 4- **Consume más recursos.** Wordpress usa más recursos que una página hecha en HTML puro, esto es por las consultas realizadas a la base de

datos, y la ejecución de scripts en PHP. Pero una manera de agilizar la carga de la página y disminuir el consumo de recursos es usar un complemento o plugin para manejar la cache, y generar paginas estáticas para la lectura del visitante, ejemplo de ellos es Súper Cache, Total Cache y muchos más que se pueden encontrar en el portal de wordpress.org

5- Requiere plugins adicionales para mejorar SEO. Las plantillas por defecto de Wordpress no traen herramientas para indicar meta etiquetas importantes como son descripción, keyword, posición geográfica y otros. Estos recursos SEO se pueden conseguir mediante plugins o modificando ligeramente el código de la plantilla.

6- Mayor cantidad de plugins mayor consumo de recursos. Si bien existen miles de plugin para Wordpress, no es aconsejable abusar de ellos, ya que reducen la eficiencia de su sitio web, lo hace más lento y si consume muchos recursos lo más probable es que su cuenta de Hosting compartido quede cancelada o sufra cuelgues intempestivos.

(A, 2013)

1.5.3 *Otras ventajas:*

1.- Administración del contenido del sitio

- Una vez armado el diseño del sitio web, podemos comenzar a añadir contenido directamente dentro de las secciones del sitio web. Dependiendo del CMS utilizado, utilizar su sitio web autoadministrable puede resultar en un verdadero dolor de cabeza si no se dispone del tiempo necesario para aprender a utilizar las herramientas básicas de edición.

- La gestión del contenido por lo general se realiza con editores de texto basados en WYSIWYG (del inglés What You See Is What You Get, lo que ves es lo que obtienes), lo que significa que, a medida que vemos lo que escribimos y añadimos en el editor, será lo que obtendremos como resultado final en el sitio web.

2.- Bajo costo de mantención.

En el momento que ocupamos con fluidez la administración de nuestro sitio web, ya no tendrá que gastar por mantener su sitio web al día. De manera fácil

podrá actualizar el contenido de su sitio, modificar y añadir secciones sin la necesidad de tener un experto que lo ayude en esta tarea.

3.- Sitio web escalable

Dependiendo del CMS utilizado, existe una gran comunidad de desarrolladores y programadores tras de él, lo que le permitirá agregar más funciones y módulos a su sitio web sin mayores problemas. Los módulos podrán variar desde simples galerías de imágenes hasta modelos avanzados para la integración de redes sociales en su sitio web.

4.- Cambio de diseño

En el caso de necesitar una renovación del diseño de su sitio web, podrá elegir entre una variedad de plantillas gratuitas para su sitio web. Estas plantillas sólo afectan cómo se ve su página, el contenido del sitio continuará siendo el mismo. De todas formas, en caso de querer realizar un cambio de diseño, lo mejor es asesorarse por un experto en el CMS que esté utilizando.

5.- Variedad en el soporte

En el caso que necesite cambiar funciones del código de su CMS, no estará amarrado a un sólo proveedor de servicios. Al ser sistemas ampliamente utilizados por una gran cantidad de diseñadores y programadores web, podrá elegir a la persona adecuada para llevar a cabo su proyecto.

1.5.4 *Otras Desventajas:*

1.- Vulnerabilidades

Al ser un sistema que se basa en la utilización de bases de datos, si su sitio web o servidor no cuenta con la protección y medidas necesarias en contra de ciber ataques, es muy probable que su página web sea hackeada. Se recomienda utilizar contraseñas fuertes en sus cuentas de usuarios, realizar copias de seguridad de su sitio regularmente, y contar con un proveedor de hosting confiable.

2.- Curva de Aprendizaje

Antes de empezar a añadir contenido de su sitio web, deberá tomarse el tiempo de leer y aprender a utilizar de manera correcta las funciones de su página. Por lo general, si empezamos a cambiar funciones que no sabemos ni para qué

sirven, lo más probable es que terminemos con un sitio desarmado o, peor aún, totalmente inutilizable. (prpdesigns(BLOG), 2013)

Una vez explicado la diferencia entre Aplicación Web y Sitio Web, y analizado sus ventajas y desventajas dadas por distintos autores, su puede llegar a la conclusión que es más acertado la idea de usar una Aplicación Web como medio para desarrollar esta investigación

1.6 Tecnologías del lado del cliente

Las tecnologías del lado del cliente se encuentran insertadas en la página HTML del cliente y son interpretadas y ejecutadas por el navegador. Por lo que, su correcta funcionalidad depende del soporte de la versión del navegador que se está utilizando. (Velasco, 2010)

1.6.1 HTML

Lenguaje de etiquetas e hipertexto (HTML, por sus siglas en inglés, HiperText Markup Language), en donde hipertexto hace referencia a la capacidad del lenguaje para la utilización de los conocidos hipervínculos. Este lenguaje está compuesto por etiquetas o marcas (tags en inglés), y gracias a ellas es posible dar forma a todos los componentes de una página. (Alvarez, 2010)

1.6.2 JavaScript

JavaScript es un lenguaje que permite la programación de pequeños scripts, pero también se pueden crear programas grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. (Alvarez, 2010)

Con JavaScript se pueden realizar acciones dentro del ámbito de una página web, crear efectos especiales y definir interactividades con el usuario. (Alvarez, 2010)

1.6.3 CSS

Hojas de estilo en cascada (CSS por sus siglas en inglés, Cascading Style Sheets). CSS, es una tecnología que nos permite crear Páginas Web de una manera más exacta. Se pueden hacer muchas cosas que no se podía hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos y colores entre otras cosas. (Alvarez, 2010)

1.7 Tecnologías del lado del servidor

Las tecnologías del lado del servidor pueden o no estar insertadas dentro de la página HTML, ya que las cuales son interpretadas y ejecutadas por el servidor. (Torre, 2012)

1.7.1 CGI

Es el sistema más antiguo que existe para la programación de las páginas dinámicas del servidor. En la actualidad se usa poco por diversas razones entre las cuales podemos destacar la dificultad con la que se desarrollan los programas y la pesada carga que representa para el servidor que los ejecuta. (Torre, 2012)

1.7.2 PHP

PHP es el acrónimo de Hipertext Preprocesor (Preprocesador de Hipertexto). Es un lenguaje de programación del lado del servidor de código abierto, independiente de la plataforma, rápido, con una gran librería de funciones y mucha documentación. Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. (Alvarez, 2010)

1.7.3 ASP

El lenguaje ASP fue desarrollado por Microsoft para el funcionamiento junto a su servidor Internet Information Server. Su funcionamiento es sencillo, cuando un cliente solicita una página ASP al servidor, está es interpretada por el mismo en lugar de enviarla directamente. Como resultado, el navegador recibe una página en código HTML puro, con los comandos ASP ya procesados. Es un lenguaje muy poderoso, sobre todo si vamos a utilizar herramientas de Microsoften conjunto, como SQL Server. (Alvarez, 2010)

1.7.4 JSP

Páginas de Servidor Java (JSP, por sus siglas en inglés, Java Server Pages), es una tecnología orientada a crear páginas Web con programación en Java.

Con JSP podemos crear aplicaciones Web que se ejecuten en varios servidores

es Web, de múltiples plataformas, ya que Java es multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar script de servidor en sintaxis Java, por lo cual las JSP se pueden escribir con el editor HTML/XML (Alvarez, 2010)

1.8 Servidores Web

Un servidor Web es un programa que implementa el protocolo HTTP (HyperText Transfer Protocol). Este protocolo está diseñado para transferir lo que se llama hipertextos, páginas Web o páginas HTML, textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. Entre los servidores más utilizados en Internet se encuentran Apache e Internet Information Services (IIS). (Alvarez, 2010)

Internet Information Server Windows XP incorpora en su versión Profesional, el servidor Internet Information Server 5.1, o IIS, es el servidor Web estándar del sistema operativo Windows. Este servidor no puede ejecutarse sobre Windows 95, 98 o Millennium; para estos casos existe una opción más sencilla llamada Servidor Web Personal que básicamente permite configurar un servidor para pruebas o para una pequeña Intranet. Entre los servicios que ofrece el Internet Information Server se encuentran servidor Web (o HTTP), servidor FTP y servidor SMTP. (Torre, 2012)

1.8.1 Apache

Apache es uno de los servidores Web más difundido y utilizado por su estabilidad, seguridad y facilidad de expansión, aparte de su costo, ya que Apache es software libre, esto significa que se distribuye por medio de una licencia gratuita y de código abierto, no hay que pagar licencias para utilizarlo y, además se dispone de su código fuente para modificar y adaptar el servidor a sus necesidades. Apache se encuentra disponible para una gran cantidad de sistemas operativos, entre los que se encuentran Linux, BeOS, FreeBSD, HP UNIX, Mac OS X, OS/2, QNX, Solaris, Rhapsody, Sun OS, UNIXware y Windows, entre otros. (Torre, 2012)

1.9 Sistemas gestores de base de datos

Sistema de gestión de bases de datos (SGBD) es el software que permite la utilización y/o la actualización de los datos almacenados en una o varias bases de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez. (Mato García, 2006)

Un SGBD tiene como objetivo fundamental suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado. (Avila, 2013)

Hay muchas formas de organizar las bases de datos, pero deben cumplir un conjunto de objetivos generales todos los SGBD, de modo que faciliten el proceso de diseño de aplicaciones y que los tratamientos sean más eficientes y rápidos, dando la mayor flexibilidad posible a los usuarios, entre los cuales se encuentran la independencia de los datos y los programas de aplicación, minimización de la redundancia, integración y sincronización de las bases de datos, integridad de los datos, seguridad y recuperación, facilidad de manipulación de la información y control centralizado. (Avila, 2013)

1.9.1 Base de datos (BD):

Se puede definir como el conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en una computadora, por lo que se puede considerar como una colección de datos variables en el tiempo. (Avila, 2013)

Existen varios SGBD entre los cuales podemos mencionar:

1.9.2 MySQL

Existe una gran competencia en el mundo de las base de datos cliente/servidor, se compite por la prestación más rápida, más segura, más confiable y más robusta, MySQL no se queda atrás en estos aspectos, es desarrollado por la empresa MySQLAB, de origen sueco, se encuentra disponible en forma gratuita bajo la Licencia General Publica GNU(GNU General Public License, GPL), lo cual significa que cualquiera puede utilizar

este producto sin costo alguno y además tener acceso a su código fuente para modificarlo y adaptarlo a otras necesidades y proyectos. MySQL AB también vende licencias comerciales del producto para aquellas empresas que necesitan soporte o no están de acuerdo con la licencia GPL. (Avila, 2013)

1.9.3 *PostgreSQL*

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto Relacionales (ORDBMS) que a partir de 1977 ha sido desarrollado de varias formas. Comenzó como un proyecto denominado Ingres en la Universidad Berkeley de California. Ingres fue más tarde desarrollado comercialmente por la Relational Technologies/Ingres Corporation. (Avila, 2013)

PostgreSQL es considerado como el sistema de bases de datos, descendiente de dominio público y código abierto más avanzado del mundo ya que posee una serie de características entre las cuales podemos mencionar (Avila, 2013)

- Relacional PostgreSQL aproxima los datos a un modelo objeto relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multiversión, soporte multiusuario, optimización de consultas, herencia, y arreglos.
- Altamente extensible PostgreSQL soporta operadores, funcionales métodos de acceso y tipos de datos definidos por el usuario. Soporte SQL comprensivo PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92. Integridad referencial PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos. API flexible. La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.
- Lenguajes procedurales como PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar

Perl, Python, o TCL como lenguaje procedural embebido. La tecnología Cliente/Servidor PostgreSQL usa una arquitectura de procesos por usuario como cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

1.10 *Fundamentación de la metodología utilizada:*

La tendencia en el software lleva a la construcción de sistemas más grandes y más complejos. Esto es debido en parte al hecho de que los computadores son más potentes cada año, y los usuarios esperan más de ellos. Esta tendencia también se ha visto afectada por el uso creciente de Internet para el intercambio de todo tipo de información. Al ver cómo se pueden mejorar los productos crece la expectativa de un software más sofisticado, que este mejor adaptado a nuestras necesidades, pero esto, a su vez, hace el software más complejo. También queremos reducir el tiempo que empleamos en desarrollar software. (Cazarez, 2011)

Conseguirlo, sin embargo, es difícil por lo que se necesita una forma coordinada de trabajar, se necesita un proceso que integre las múltiples facetas del desarrollo, se necesita un método común un proceso que:

Proporciona una guía para ordenar las actividades de un equipo según (Cazarez, 2011):

- Dirige las tareas de cada desarrollador por separado y del equipo como un todo.
- Especifique los artefactos que deben desarrollarse Ofrezca criterios para el control y la medición de los productos y actividades del proyecto. Se puede decir que la presencia de un proceso bien definido y bien gestionado es la diferencia entre proyectos hiperproductivos y otros que fracasan.
- Proceso Unificado Racional (RUP) El Proceso Unificado Racional está equilibrado por ser el producto final de tres décadas de desarrollo y uso práctico. Su desarrollo como producto ha seguido un camino desde el Proceso Objectory (primera publicación en 1987) pasando por el

Proceso Objectory de Rational (publicado en 1997) hasta el Proceso Unificado de Rational (publicado en 1998)

1.10.1 RUP

El Proceso Unificado Racional es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado Racional es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. (Cazarez, 2011)

El Proceso Unificado Racional, es dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental. Proceso que se organiza en cuatro fases: inicio, elaboración, construcción y transición, se estructura en torno a cinco flujos de trabajo fundamentales: recopilación de requisitos, análisis, diseño, implementación y pruebas.

(Cazarez, 2011)

El Proceso Unificado Racional utiliza el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) para preparar todos los esquemas de un sistema software. (Cazarez, 2011)

1.10.2 UML

El Lenguaje Unificado de Modelado, es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. Pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. Los conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan

generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. (Cazarez, 2011)

Un artefacto es una pieza de información tangible que es creada, modificada y usada por los trabajadores al realizar actividades; representa un área de responsabilidad, y es candidata a ser tenida en cuenta para el control de la configuración. Puede ser un modelo, un elemento de un modelo o un documento. (Cazarez, 2011)

Fundamentación del lenguaje, gestores de bases de datos y de otros softwares y hardware que serán utilizados:

1.11 ¿Por qué utilizar PHP?

Se decide utilizar PHP porque es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo Web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor Web, tomando el código en PHP como su entrada y creando páginas Web como salida. Puede ser desplegado en la mayoría de los servidores Web y en casi todos los sistemas operativos y plataformas sin costo alguno, además posee diferentes características entre las cuales podemos mencionar velocidad, estabilidad, seguridad y simplicidad.

1.11.1 Velocidad:

No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, está listo para ser utilizado.

1.11.2 Estabilidad:

La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad

de ejecuciones. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable, Seguridad: El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo

1.11.3 *Simplicidad:*

Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente. Entre las ventajas adicionales de utilizar PHP tenemos: PHP corre en (casi) cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de Unix, Windows (95,98,NT,ME,2000,XP) y Macs. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al Sistema Operativo.

La sintaxis de PHP es similar a la del C, por esto cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP. Entre los lenguajes del tipo C incluimos al Java y JavaScript, de hecho mucha de la funcionalidad del PHP se la debe al C en funciones como `fread()` o `strlen()`, así que muchos programadores se sentirán como en casa.

PHP es completamente expandible. Está compuesto de un sistema principal (escrito por Zend), un conjunto de módulos y una variedad de extensiones de código. Muchas interfaces distintas para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTPD. Otra alternativa es configurarlo como módulo CGI. Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos. Siempre podrás disponer de ODBC para situaciones que lo requieran. Una gran variedad de módulos cuando un programador PHP necesite una interface para una librería en particular, fácilmente podrá crear una API para esta. Algunas de las que ya vienen implementadas permiten manejo de gráficos, archivos PDF, Flash, Cybercash, calendarios, XML, IMAP, POP, etc. Rapidez. PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz. Está completamente escrito en C, así que se ejecuta rápidamente utilizando poca

memoria. PHP es Open Source, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no estás forzado a pagar actualizaciones anuales para tener una versión que funcione.

1.12 ¿Por qué utilizar HTML?

Se decide utilizar HTML ya que es un lenguaje muy sencillo que se basa en el uso de etiquetas, además nos permite presentar la información a través de los navegadores y se utiliza para marcar los contenidos, es decir, para designar lo que es un párrafo, lo que es un titular o lo que es una lista de elementos entre otras cosas.

1.13 ¿Por qué utilizar JavaScript?

Se decide utilizar JavaScript ya que permite ejecutar secuencias de comandos en el mismo navegador del usuario, se pueden realizar cálculos rápidos y complejos y hasta controlar la mayor parte de los elementos del navegador. También se pueden ejecutar acciones como enviar mensajes, crear calendarios y validar los formularios para evitar errores y no recargar la página antes de enviarlos al servidor aspecto de suma importancia

1.14 ¿Por qué utilizar CSS?

Se decide utilizar CSS ya que es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es de vital importancia para crear páginas Web complejas. Las ventajas que presentan la separación de los contenidos y su presentación son numerosas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo, a los cuales se les llama también “documentos semánticos”. Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

1.15 ¿Por qué utilizar PostgreSQL?

PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multiversión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike). PostgreSQL ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema

- Clases
- Herencia
- Tipos
- Funciones

Otras características que aportan potencia y flexibilidad adicional:

- Restricciones (Constraints)
- Disparadores (triggers)
- Reglas (rules)
- Integridad transaccional

1.16 ¿Por qué utilizar Apache?

El servidor Apache es un servidor de código abierto, corre sobre varias plataformas Unix (BSD, GNU/Linux, etcétera), Windows entre otras. Entre las características que presenta se encuentran, mensajes de error altamente configurables y bases de datos de autenticación y negociado de contenido. Es uno de los servidores Web más utilizados a nivel mundial, multiplataforma, posee infinidad de paquetes y módulos que permiten trabajar con gran cantidad de lenguajes de programación Web, así como intérpretes de SQL y otras funciones, permite transacciones seguras mediante SSL (Secure Socket Layer) y contiene soporte para Hosts virtuales.

1.17 Herramientas de desarrollo:

- **Macromedia Dreamweaver** es una de las herramientas más utilizadas por los web masters que trabajan con aplicaciones visuales, ya que se adapta increíblemente a las necesidades de todo tipo de profesional del diseño Web, tanto para los que prefieren programar el código directamente en el editor de texto como para los que gustan de una metodología de trabajo totalmente visual. Dreamweaver ha incorporado desde la versión MX, todo el poder de otra herramienta de Macromedia que ya había ganado gran aceptación: HomeSite es un editor de texto especialmente diseñado para trabajar con documentos Web como HTML, PHP, ASP, CSS o JavaScript, entre otros. HomeSite fue añadido a la Vista código de Dreamweaver, con lo que se ganó muchísimo en comodidad y velocidad a la hora de escribir. (Alvarez, <http://www.desarrolloweb.com>, 2011)

El área visual de Dreamweaver también tiene grandes ventajas, sobre todo se destaca en el manejo de capas y, desde la versión 4, en la construcción de tablas utilizando la herramienta tablas y celdas de disposición.

(Alvarez, <http://www.desarrolloweb.com>, 2011)

1.18 Visual Paradigm:

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. También proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

1.18.1 Ventajas:

- Presenta licencia gratuita y comercial. Es fácil de instalar y actualizar y compatible entre ediciones.
- Fácil de usar, muy intuitivo.
- Multiplataforma.
- Soporte de UML versión 2.1.

- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- Modelado colaborativo con CVS y Subversión (control de versiones).
- Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse) a través de XMI.
- Ingeniería de ida y vuelta.
- Ingeniería inversa - Código a modelo, código a diagrama.
- Ingeniería inversa Java, C++, Esquemas XML, XML, NET exe/dll, CORBA IDL.
- Generación de código - Modelo a código, diagrama a código.
- Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Diagramas EJB - Visualización de sistemas EJB.
- Generación de código y despliegue de EJB - Generación de beans para el desarrollo y despliegue de aplicaciones.
- Diagramas de flujo de datos.
- Soporte ORM - Generación de objetos Java desde la base de datos.
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- Generador de informes.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- Importación y exportación de ficheros XMI.
- Integración con Visio – Dibujo de diagramas UML con plantillas (stencils) de Microsoft Visio.
- Editor de figuras.
(JHERSON, 2013)

1.18.2 Desventajas:

No permite su uso en versiones comerciales sin tener la licencia, ejemplo de ellos es su marca de agua en cada diagrama.

(JHERSON, 2013)

1.19 Modelo 2 Capas

La arquitectura tradicional cliente/servidor también es conocida como arquitectura de dos capas. Requiere una interfaz de usuario que se instala y se ejecuta en una PC o estación de trabajo y envía solicitudes a un servidor para ejecutar operaciones complejas. Por ejemplo, una estación de trabajo utilizada como cliente puede ejecutar una aplicación de interfaz de usuario que interroga a un servidor central de bases de datos. (Valladolid, 2010)

El desarrollo de aplicaciones en un ambiente de dos capas funciona adecuadamente, pero no es necesariamente lo más eficiente. Las herramientas para el desarrollo con dos capas son robustas y ampliamente evaluadas. Las técnicas de ingeniería de software de prototipo se emplean fácilmente. Las soluciones de dos capas trabajan bien en ambientes no dinámicos estables, pero no se ejecutan bien en organizaciones rápidamente cambiantes. (Valladolid, 2010)

1.19.1 Desventajas del sistema de dos capas:

- Las desventajas de la arquitectura cliente/servidor según (Valladolid, 2010):
 - 1- Los ambientes de dos capas requieren control excesivo de las versiones y demandan esfuerzo de distribución de la aplicación cuando se les hacen cambios. Esto se debe al hecho de que la mayoría de la aplicación lógica existe en la estación de trabajo del cliente.
 - 2- La seguridad del sistema en un diseño de dos capas es compleja y a menudo requiere administración de las bases de datos; esto es debido al número de dispositivos con acceso directo al ambiente de esas bases de datos.
 - 3- Las herramientas del cliente y de la base de datos, utilizadas en diseños de dos capas, constantemente están cambiando. La dependencia a

largo plazo de cualquier herramienta, puede complicar el escalamiento futuro o las implementaciones.

1.20 Modelo Vista Controlador

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica del 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. Este patrón de diseño se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

(Practices, 2013)

Aunque de forma algo simplista, podríamos definir MVC como un patrón arquitectural que describe una forma de desarrollar aplicaciones software separando los componentes en tres grupos (o capas):

- El **Modelo** que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- La **Vista**, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.
- El **Controlador**, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

MVC son las siglas de Modelo-Vista-Controlador, y se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

1.20.1 ¿Qué ventajas tiene el uso del patrón MVC?

Como siempre esto de enumerar ventajas es algo subjetivo, por lo que puede que pienses que falta o sobra alguna. En un primer asalto podríamos aportar las siguientes:

- Clara separación de responsabilidades entre interfaz, lógica de negocio y de control, que además provoca parte de las ventajas siguientes.

- Facilidad para la realización de pruebas unitarias de los componentes, así como de aplicar desarrollo guiado por pruebas (TDD)
- Simplicidad en el desarrollo y mantenimiento de los sistemas.
- Reutilización de los componentes.
- Facilidad para desarrollar prototipos rápidos.
- Sencillez para crear distintas representaciones de los mismos datos.
- Los sistemas son muy eficientes, y a la postre más escalables.

Pero bueno, también se pueden citar algunos inconvenientes:

- Tener que ceñirse a una estructura predefinida, lo que a veces puede incrementar la complejidad del proyecto. De hecho, hay problemas que son más difíciles de resolver, o al menos cuestan algo más de trabajo, respetando el patrón MVC.
- Al principio puede cierto esfuerzo adaptarse a esta filosofía, sobre todo a desarrolladores acostumbrados a otros modelos más cercanos al escritorio, como Webforms.
- La distribución de componentes obliga a crear y mantener un mayor número de ficheros. (Aguilar, 2013)

1.20.2 *¿TIENE DESVENTAJAS EL MVC?*

El MVC es una concepción de programación, separando el código ejecutor (modelo), del código de presentación (vista) y del código que gestiona (controlador). En sí mismo no tiene desventajas, ya que es un concepto genérico. Se puede decir que el modelo no tiene en cuenta la documentación, por ejemplo, pero cualquier programador que use el MVC puede aplicar un desarrollo exquisito de documentación sin salirse del MVC. (Casablanca, MVC, 2010)

1.21 *Bibliotecas*

1.21.1 *FPDF*

FPDF es una clase escrita en PHP que permite generar documentos PDF directamente desde PHP, es decir, sin usar la biblioteca PDFlib. La F de FPDF

significa Free (gratis y libre): puede usted usarla para cualquier propósito y modificarla a su gusto para satisfacer sus necesidades.

FPDF tiene otras ventajas:

- Funciones de alto nivel. Esta es una lista de sus principales características:
 - Elección de la unidad de medida, formato de página y márgenes.
 - Gestión de cabeceras y pies de página.
 - Salto de página automático.
 - Salto de línea y justificación del texto automáticos.
 - Admisión de imágenes (JPEG, PNG y GIF).
 - Colores.
 - Admisión de fuentes TrueType, Type1 y codificación.

Compresión de página. FPDF no necesita de ninguna extensión para PHP (excepto zlib para activar la compresión y GD para soporte a GIF) y funciona con PHP4 ($\geq 4.3.10$) y PHP5.

(TEMIQUEL, 2013)

1.21.2 *Highcharts*

En Internet lo más importante es tener siempre presente las estadísticas de nuestro negocio en todo momento para cotejar valores en alza y baja.

Esta vez se quiere presentar una librería Javascript que dependiendo de la finalidad que queramos darle, nos puede servir de mucho. Se trata de highcharts JS una librería javascript desde la que podemos implementar multitud de opciones para estadísticas, desde la típica gráfica de barras con histórico a complejas composiciones para detallar valores. Cuenta con multitud de tipos de gráficas, entre las que encontramos:

- Line & Scatter Charts
- Area Charts
- Column & Bar Charts
- Pie Charts
- Dynamic Charts
- Combination Charts
- Angular gauge

(WordPress, 2013)

Highcharts es una biblioteca de gráficos escrito en JavaScript puro, que ofrece una forma fácil de añadir gráficos interactivos a una aplicación web.

Es compatible con cualquier navegador que soporte Javascript, por lo cual funciona con Windows, Linux, Mac, Android, iOS, Windows Phone, Firefox, Chrome, Safari, Internet Explorer, así como en un Smartphone, en una Tablet, Highcharts incluye una gran colección tipos de gráficos que podemos utilizar, desde gráficos de torta, columna de datos, gráficos lineales, series de tiempo, gráficos dinámicos, animaciones y otro tipo de gráficos complejos por ejemplo áreas con puntos faltantes Mencionada la información anterior se recurre a una herramienta como lo es Highcharts, para incluir en la plataforma CRM la opción de registros y reportes estadísticos facilitándole al usuario el análisis y la interpretación de la información almacenada en el sistema

(VELASCO GIRALDO y SAA VIAFARA, 2012)

Highcharts es una biblioteca de gráficos escrito en JavaScript puro, que ofrece una forma fácil de añadir gráfico s interactivos a su sitio web o aplicación web. Highcharts actualmente soporta gráficos de línea, área, columna, barra, circular, dispersión, medidores angulares, polares.

(Muñoz, 2010)

En este caso se toma como punto de referencia en la investigación el concepto de Diego Fernando y Alejandro (VELASCO GIRALDO y SAA VIAFARA, 2012) por muy abarcador en la idea que se quiere expresar.

1.22 Definición de software libre

El software libre (free software), es aquel que una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

Para estudiarlo y modificarlo la distribución del Software Libre debe incluir el código fuente, característica fundamental.

El software libre suele estar disponible gratuitamente, pero no hay que asociar software libre a software gratuito, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así y, aunque conserve su carácter de libre, puede ser vendido comercialmente.

Software gratis o gratuito (conocido usualmente como freeware)

Donde en algunos casos es posible encontrar el código fuente pero solo para efectos ilustrativos o educativos, sin la posibilidad de ser modificado, regularmente el software gratuito ofrece las aplicaciones ya compiladas y listas para usarse.

1.22.1 Características del software libre

De acuerdo a la definición, el software es libre si se garantizan las 4 libertades siguientes:

Libertad 0	Libertad 1	Libertad 2	Libertad 3
La libertad de usar el programa, con cualquier propósito (privado, educativo, público, comercial, militar, etc.)	La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (para lo cual es necesario poder acceder al código fuente)	La libertad de distribuir copias, con lo que puedes ayudar a quien sea	La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie

El derecho a utilizar: El primer derecho o libertad, el que trata sobre el derecho a utilizar software, puede sorprender. ¿Acaso no tiene la gente derecho a utilizar un programa que ha comprado en una tienda, por ejemplo? La respuesta es que cuando una persona "compra" un programa de ordenador que no es software libre (programas privativos, los llamamos) por lo general no dispone del derecho de utilización ilimitada que nosotros consideramos inevitable: El usuario está limitado a utilizar el programa para determinados objetivos (prohibido usar este programa de forma comercial) o en determinados sitios (prohibido usar este programa en el país X y el país Y) o en un número determinado de máquinas (prohibido usar este programa en más de una máquina al mismo tiempo). Estas restricciones son muy habituales cuando hablamos de software privativo, y pensamos que son inadmisibles. Por eso hacemos énfasis en el derecho a utilizar: es vulnerado constantemente por las compañías y empresas que fabrican y distribuyen software privativo.

El derecho a entender: Por otra parte tenemos la segunda libertad para el usuario: el derecho a entender cómo funcionan los programas que nos distribuyen, y a adaptarlo a nuestras necesidades. De nuevo, este derecho fundamental se ve vulnerado cuando de software privativo hablamos: por lo general, el software privativo se distribuye en forma de ejecutables (equivalentes a los ficheros ".exe" en entornos Windows) sin que le acompañe el código fuente correspondiente. El código fuente de un programa es su forma entendible y modificable por un programador. En cierto modo, es como el manual de instrucciones de algún aparato. Si compramos un mando a distancia universal, ajustado de fábrica a nuestro televisor, y no viene con instrucciones, no seremos capaces de utilizarlo en nuestro otro televisor, el de la cocina. Por contra, si tenemos acceso a las instrucciones, podremos sintonizar el mando con tantos televisores como queramos. En eso consiste adaptar los artefactos que utilizamos a nuestras necesidades. En el caso de los programas de ordenador, para ello necesitamos el código fuente.

El derecho a distribuir: De aquí nos vamos al tercer derecho o libertad: el derecho a distribuir programas de ordenador de forma gratuita o, alternativamente, cobrando algo a cambio de hacerlo. Este es tal vez el punto que mejor comprenden los usuarios de programas privativos cuando quieren aprender qué es el software libre. Es natural, ya que la industria del software privativo hace continuos esfuerzos para intentar convencer a la sociedad de que copiar programas de ordenador es algo que no debe hacerse. El Software Libre, por el contrario, incluye el poder ser copiado sin necesidad de grandes recursos (con una unidad de grabación basta) y la característica peculiar de que la copia no pierde calidad respecto al original no es algo malo: por el contrario, es casi lo mejor que tiene el software. Copiar programas de ordenador y distribuirlos es algo que beneficia a la sociedad. Es de sentido común. Realizar copias de programas privativos es algo ilegal en la mayoría de los países. Por eso el sentido del software libre: es perfectamente legal copiarlo. De esta forma tanto el usuario como la sociedad se benefician, y nadie sale perdiendo (la copia original no funciona peor por haber hecho una o millones de copias). Es importante un detalle: el software libre no tiene por qué ser gratis. Es perfectamente posible distribuir software libre a cambio de dinero.

Así es como pueden ganarse la vida los programadores y distribuidores. Ahora bien, eso no justifica el hecho de vulnerar los derechos de la gente que paga por obtener una copia del programa: el usuario puede distribuir sus propias copias, cobrando por ello si lo desea.

El derecho a mejorar: El último derecho o libertad fundamental, el derecho a mejorar el software y distribuir las mejoras, es tal vez el que más controversia genera. Obviamente, el usuario de software privativo no puede mejorar los programas que utiliza: aunque quisiera y supiera hacerlo, por lo general no tiene acceso al código fuente. Y aunque lo tuviera (puede distribuirse el código fuente y no obstante no ser software libre) sería ilegal modificar ese código fuente. Sin embargo, el software libre siempre se distribuye con su código fuente, y además es totalmente legal modificarlo. La objeción típica a este derecho suele ser: "bien, tengo el código fuente. Pero YO no sé modificarlo para mejorarlo. ¿De qué me sirve entonces tenerlo?". La respuesta a esa pregunta (muy habitual y muy comprensible) es que para mejorar o modificar un programa no es necesario hacerlo uno mismo. De la misma forma que una persona puede llevar su coche a un taller a que se repare (o se le introduzcan algunas mejoras) y además puede hacerlo al taller que más le guste (el fabricante del coche no puede poner restricciones) el usuario de software debe poder contratar a un profesional (o a una empresa, o pedir a su amigo el informático que lo haga gratis) que mejore el software libre que ha comprado o que ha descargado de Internet. Si el mismo usuario compra o descarga software privativo y quiere mejorarlo, entonces depende totalmente de la compañía que le vendió (o proporcionó vía Internet) el programa. Si la compañía decide que las mejoras deseadas por el usuario no son rentables o no son convenientes por alguna otra razón, no las incorporará al programa. El resultado es que con software privativo el usuario siempre pierde. Con software libre, por el contrario, el usuario es libre y no está atado a ninguna compañía ni empresa. Además, el usuario puede distribuir el programa que incorpora sus mejoras gratuitamente o por un pago (aquí aplica el tercer derecho explicado anteriormente). Y otra cosa importante: el usuario también tiene derecho a no distribuir sus mejoras si no quiere. Una persona puede descargar o comprar

software libre, introducirle mejoras, y no redistribuir ni hacer públicas dichas mejoras.

1.22.2 *Diferencias con el open source (código abierto)*

La principal diferencia entre los términos "open source" y "free software" es que éste último tiene en cuenta los aspectos éticos y filosóficos de la libertad, mientras que el "open source" se basa únicamente en los aspectos técnicos. Aunque en la práctica el software Open Source y el software libre comparten muchas de sus licencias, la FSF opina que el movimiento Open Source es filosóficamente diferente del movimiento del software libre. Apareció en 1998 con un grupo de personas, entre los que cabe destacar a Eric S. Raymond y Bruce Perens, que formaron la Open Source Initiative (OSI). Ellos buscaban darle mayor relevancia a los beneficios prácticos del compartir el código fuente, e interesar a las principales casas de software y otras empresas de la industria de la alta tecnología en el concepto. Mucha gente reconoce el beneficio cualitativo del proceso de desarrollo de software cuando los desarrolladores pueden usar, modificar y redistribuir el código fuente de un programa. El movimiento del software libre hace especial énfasis en los aspectos morales o éticos del software, viendo la excelencia técnica como un producto secundario deseable de su estándar ético. El movimiento Open Source ve la excelencia técnica como el objetivo prioritario, siendo la compartición del código fuente un medio para dicho fin. Por dicho motivo, la FSF se distancia tanto del movimiento Open Source como del término "Open Source". Puesto que la OSI sólo aprueba las licencias que se ajustan a la OSD (Open Source Definition), la mayoría de la gente lo interpreta como un esquema de distribución, e intercambia libremente "open source" con "software libre". Aun cuando existen importantes diferencias filosóficas entre ambos términos, especialmente en términos de las motivaciones para el desarrollo y el uso de tal software, raramente suelen tener impacto en el proceso de colaboración.

1.22.3 *Pros y contras del uso de software libre*

- Costo
- Seguridad
- Soporte

- Popularidad
- Variedad
- Documentación
- Escalabilidad
- Personalización

1.22.4 *Comunidades de desarrollo de software libre*

- Grupos de desarrollo por todo el mundo
- Modelos interdisciplinarios (todas las áreas)
- Participación activa y reactiva
- Fundaciones creadas específicamente para la promoción y desarrollo de proyectos de SL (superar problema de supervivencia y fondeo)
- Organismos gubernamentales orientados y dedicados a esta nueva forma de hacer tecnología

1.22.5 *Impacto del software libre*

- Gobiernos
- Entidades educativas (Inclusión de materias obligatorias en carreras de Ingeniería e informática, Argentina, Chile, México y España)
- Soluciones empresariales (Servidores 75% bajo alguna versión de Linux)
- Iglesia (Primeros esfuerzos, incipientes comunidades de desarrollo, esfuerzos aislados que deberían encontrar la forma de unirse, Francia (Cristux), Bibliux (USA), Aleluya (MX).

1. **Cuba** donde el gobierno ha establecido una indicación oficial para introducir de manera progresiva el software libre y en particular el Linux y en el que la red de Salud Pública, Infomed, fue pionera en su uso. Cuba, como otros muchos países, está apostando por la informatización de la sociedad a partir del uso del software libre en un esfuerzo por socializar y masificar el uso de estas tecnologías

2. **China**, con su acuerdo con Sun Microsystems para distribuir millones de Java Desktop (una distribución de Linux basada en GNOME y especialmente bien integrada con java)

3. **Brasil**, con una actitud generalmente positiva, y, por ejemplo, con el desarrollo de los telecentros
4. **En España**, distintos gobiernos regionales están desarrollando sus propias distribuciones no sólo para uso administrativo sino también académico. Así tenemos LinEx en Extremadura, GuadaLinux en Andalucía, LliureX en La Comunidad Valenciana, Molinux en Castilla-La Mancha, MAX en La Comunidad de Madrid y Trisquel en la Comunidad de Galicia, LInkcat en Cataluña y Melinux en Melilla por el momento. Estas distribuciones están basadas en Debian, Ubuntu y Suse.
5. **Venezuela** donde el presidente Chávez decretó el uso prioritario del software libre y GNU/Linux en toda la administración pública, incluyendo ministerios y oficinas gubernamentales y se está fomentando la investigación y el desarrollo de software libre. Decreto 3390. En Venezuela se discute una ley sobre Tecnologías de la Información que incluye al Software Libre.
6. **Chile**, donde el Ministerio de Educación y la Universidad de la Frontera (ubicada en Temuco) crearon EduLinux, una distribución que hoy está en más de 1500 escuelas chilenas y funcionando en más de un 90% de las bibliotecas chilenas. Actualmente las Fuerzas Armadas chilenas están planificando la creación de una distribución militar que interconecte a las ramas de la defensa chilena. El gobierno de ese país aprobó el uso del software libre en la administración pública, anulando así un contrato previo con Microsoft para el mantenimiento de las redes y de los equipos en escuelas y bibliotecas chilenas.
7. **República Dominicana**, promociona el uso y proliferación del Software libre en el campo educativo y científico. Dispone de dos fundaciones, una en la capital de Santo Domingo y la otra en la ciudad de Santiago. Codigolibre.org
8. **Argentina**: En su momento el Gobierno Nacional apoyo el desarrollo del software libre, a través del portal SoftwareLibre.gov.ar. El Gobierno de la Ciudad de Buenos Aires utiliza Ututo en sus escuelas, como enrutador: UtutoR. También se puede destacar en el proyecto Munix, de la ciudad de Rosario. Este proyecto (actualmente con 300 puestos de trabajos migrados) reemplaza el software propietario en las estaciones de trabajo obsoletas por software libre, reutilizando dichas estaciones de trabajo como clientes delgados. Este

proyecto se lleva a cabo utilizando Debian GNU/Linux y está basado en Linux Terminal Server Project.

9. **México** el Gobierno del Distrito Federal dentro de sus políticas y lineamientos en materia de Informática da preferencia al uso del Software Libre. La Delegación Tlalpan crea la distribución Gobierno GDF/Linux en conjunto con el proyecto del Laboratorio de Informática y Software Libre (LINFOSOL) con el que se pretende investigar y experimentar con el software libre, buscando aportar datos que ayuden a una futura migración al software libre.

Todo lo antes explicado aquí hace referencia a (Foundation, 2014)

1.23 Conclusiones

Se describieron los conceptos básicos asociados al dominio del problema, identificando la necesidad que existe de la aplicación de las Tecnologías de la Información y las Comunicaciones en la APPP de Sancti Spíritus, con el objetivo de contribuir a la gestión de la información en el control de equipos informáticos a través de una aplicación Web, utilizando como metodología RUP que utiliza UML como lenguaje de modelado, como lenguajes de programación del lado del cliente HTML, CSS y JavaScript, y del lado del servidor PHP basado en el patrón de diseño MVC, para lo cual resultan apropiadas las herramientas Visual Paradigm, Macromedia Dreamweaver, pgAdmin, SQL Manager 2007 for PostgreSQL y Help & Manual, como SGBD PostgreSQL y servidor Web Apache.

Capítulo 2

Diseño de una Aplicación Web que contribuya al proceso de gestión de la información del fondo de tierras ociosas en Cuba.

2.1 *Introducción:*

En este capítulo se exponen las reglas del negocio a tener en cuenta para el diseño del sistema, una caracterización de los actores y trabajadores del mismo, sus casos de uso, conformándose el diagrama de casos de uso, la descripción de estos en formato expandido, y se muestran los diagramas de actividades y del modelo objetos. Además se describe de modo general el funcionamiento del sistema diseñado. Se definen los requerimientos funcionales y no funcionales, se realiza la descripción del modelo de casos de uso del sistema, basándose en los actores y los casos de uso.

2.2 *Modelo del negocio*

Un proceso de negocio es un conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido. Cada proceso de negocio tiene sus entradas, funciones y salidas. Las entradas son requisitos que deben tenerse antes de que una función pueda ser aplicada. Cuando una función es aplicada a las entradas de un método, tendremos ciertas salidas resultantes. Es una colección de actividades estructurales relacionadas que producen un valor para la organización, sus inversores o sus clientes. Es, por ejemplo, el proceso a través del que una organización ofrece sus servicios a sus clientes. Un proceso de negocio puede ser parte de un proceso mayor que lo abarque o bien puede incluir otros procesos de negocio que deban ser incluidos en su función. En este contexto un proceso de negocio puede ser visto a varios niveles de granularidad. El enlace entre procesos de negocio y generación de valor lleva a algunos practicantes a ver los procesos de negocio como los flujos de trabajo que efectúan las tareas de una organización (WorldPress, 2010)

2.2.1 Modelado del negocio

El modelo de Casos de Uso del Negocio es el encargado de describir los procesos de una empresa utilizando los casos de uso y los actores, en correspondencia, a su vez, con los procesos del negocio y los clientes. (WorldPress, 2010)

2.2.2 Reglas del negocio a considerar

Las reglas de negocio son condiciones, estándar o reglas que deben ser cumplidas y controladas por la organización durante el flujo del proceso. Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. (Bizagi, 2010)

Partiendo de lo planteado anteriormente fueron identificadas las siguientes reglas:

- El usuario que solicite la información debe ser trabajador Centro de Higiene y Epidemiología.
- Para tener acceso a la información y herramientas que ofrece el sistema es necesario que la persona se encuentre registrada como usuario del mismo.
- La única persona que tiene autorización para eliminar contenido o cuentas de usuario del sistema así como de crear un usuario con privilegios administrativos es el director del Centro de Higiene y Epidemiología.
- Para la inserción de datos en el sistema va a ser realizada por el trabajador destinado a esta tarea dentro del propio Centro de Higiene y Epidemiología y estos datos deben cumplir con la estructura para el tipo de información que se desea insertar.

2.2.3 Actores del Negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. (WorldPress, 2010)

Nombre del actor	Justificación
Epidemiólogo	Es quien se encarga de emitir las alertas pertinentes según el informe dado por el Bioestadístico a partir del análisis del estado de las enfermedades.

Tabla 1 Actores del negocio

2.2.4 *Trabajadores del negocio*

Un trabajador es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores y manipulando entidades. (Booc & Rumbaugh, 2006)

Nombre del trabajador	Justificación
Bioestadístico	Es quien consulta, revisa y gestiona la Información de las enfermedades en el diario de la semana estadística.

Tabla 2. Trabajadores del Negocio.

2.3 *Diagrama de Casos de Uso del Negocio*

Un diagrama de casos de uso representa gráficamente a los procesos del negocio como casos de usos y su interacción con los actores. (WorldPress, 2010)

Para tener una visión general de los procesos del negocio en la entidad se confecciona el diagrama de Casos de Uso del Negocio.

Figura 1 Diagrama de Casos de Uso del Negocio Recibir Información. Ver Anexo # 3.

2.3.1 *Descripción de los Casos de Uso del negocio*

Luego de identificar el proceso del negocio es necesario explicarlo detalladamente. A continuación se muestra la descripción del mismo.

CU 1	Recibir Información
Actores	Epidemiólogo
Propósito	Obtener Información
Resumen	
<p>El caso de uso se inicia cuando los centros hospitalarios del municipio le entregan la información semanal referente a las enfermedades al epidemiólogo, siendo este el encargado de entregarle dicha información al bioestadístico, una vez que este trabajador posee esta documentación, la inserta en el documento Excel “Semana estadística”, y comienza a realizar los cálculos para determinar el corredor endémico de las enfermedades en cuestión, una vez terminada esta acción, el bioestadístico realiza un informe y se lo entrega al epidemiólogo. De esta manera termina el caso de uso.</p>	
Casos de uso asociados:	-
Flujo de Trabajo	
Acción del actor	Respuesta del negocio
	1- El caso de uso se inicia cuando los centros hospitalarios del municipio le entregan la información semanal referente a las enfermedades al bioestadístico.
	2- La inserta en el documento Excel “Semana estadística”.
	3- Comienza a realizar los cálculos para determinar el corredor endémico de las enfermedades en cuestión.
	4-Una vez terminada esta acción, el bioestadístico realiza un informe y se lo entrega al epidemiólogo.

5- El epidemiólogo recibe el reporte de las enfermedades dadas por el bioestadístico, y termina el caso de uso.	
Prioridad:	Alta
Mejoras:	- Todos los cálculos utilizados en el corredor se realizan de forma automática.
Cursos alternos:	-

Tabla 3 Descripción del Caso de Uso del Negocio: Recibir Información.

2.4 Diagrama de actividades

El diagrama de actividad es un grafo que contiene los estados en que puede hallarse la actividad a analizar. Cada estado de la actividad representa la ejecución de una sentencia de un procedimiento, o el funcionamiento de una actividad en un flujo de trabajo. En resumen describe un proceso que explora el orden de las actividades que logran los objetivos del negocio. (Booc & Rumbaugh, 2006).

Para un mejor entendimiento del proceso en cuestión se realiza el Diagrama de Actividad correspondiente al Caso de Uso del Negocio Solicitar Información. El diagrama de actividad se presenta a continuación.

Figura 2 Diagrama de Actividades, Caso de Uso recibir información ver Anexo # 4.

2.5 Modelo de objetos del negocio

Un modelo de objetos del negocio es un modelo interno a un negocio. Describe como cada caso de uso del negocio es llevado a cabo por parte de un conjunto

de trabajadores que utilizan un conjunto de entidades del negocio y unidades de trabajo. (Booc & Rumbaugh, 2006)

Figura 3 Diagrama Modelo de objetos: Caso de uso Recibir Información.
Ver anexo # 5.

2.6 Especificación de los requisitos de software.

2.6.1 Requerimientos funcionales.

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo. (Booc & Rumbaugh, 2006)

Teniendo en cuenta lo planteado, el sistema propone los siguientes requerimientos funcionales:

2.6.2 Requerimientos funcionales:

1- Iniciar sesión.

1.1 Entrar al sistema

2- Gestionar usuarios.

2.1 Crear nuevo usuario.

2.2 Modificar usuario

2.3 Eliminar usuario.

3- Restaurar contraseña.

4- Gestionar provincia

4.1 Crear un registro de nueva provincia

4.2 Modificar el registro de una provincia

4.3 Eliminar el registro de una provincia

5- Gestionar municipio

5.1 Crear un registro de un nuevo municipio

5.2 Modificar el registro de un municipio

5.3 Eliminar el registro de un municipio

6- Gestionar Área de salud

6.1 Registrar Área de salud

6.2 Modificar Área de salud

6.3 Eliminar Área de salud

7- Gestionar Enfermedades.

7.1 Registrar enfermedad

7.2 Modificar enfermedad

7.2 Eliminar enfermedad

8- Gestionar datos de Casos por Enfermedades

8.1 Registrar caso.

8.2 Modificar caso.

8.3 Eliminar caso

9- Gestionar Tasa por Enfermedades

9.1 Calcular Tasa.

10-Gestionar datos estadísticos acumulados por enfermedades.

10.1 Calcular los datos estadísticos acumulados por enfermedades

11-Gestionar Datos Estadísticos de la Semana

11.1 Registrar Semana

11.2 Modificar Semana

11.3 Eliminar Semana

12-Gestionar Datos Estadísticos de la población

12.1 Registrar nuevos datos de la población

12.2 Modificar datos de la población.

12.3 Eliminar datos de la población.

13-Gestionar prevalencia acumulada

13.1 Calcular la prevalencia acumulada

14- Gestionar la Transformación logarítmica de las tasas

14.1 Calcular la transformación logarítmica de las tasas

15- Gestionar Cálculo de medias. Desviación estándar e intervalos de confianza al 95%

15.1 Calcular la media. Calcular desviación estándar e intervalos de confianza al 95%.

16- Gestionar transformación a unidades originales, menos 1 y cálculo de casos

16.1 Calcular la transformación a unidades originales, menos 1. Calcular los casos.

17- Gestionar Cálculo de las diferencias

17.1 Calcular las diferencias

18- Gestionar Cálculo de tasas y suma 1. Acumulado

18.1 Calcular las tasas y sumar 1 a los acumulados.

19-Gestionar la Transformación logarítmica de las tasas. Acumulado.

19.1 Calcular la transformación logarítmica de las tasas a los acumulados.

20-Gestionar Cálculo de medias. Desviación estándar e intervalos de confianza 95%. Acumulado.

20.1 Calcular las medias. Desviación Estándar e intervalos de confianza a los acumulados.

21-Gestionar la Transformación a unidades originales, menos 1 y cálculo de casos. Acumulado.

21.1 Calcular la transformación a unidades originales, menos 1 y cálculo de casos a los acumulados.

22-Gestionar el Cálculo de las diferencias. Acumulado.

22.1 Calcular las diferencias acumuladas.

23-Gestionar reportes

23.1 Mostrar enfermedades

23.2 Mostrar casos y tasas por trimestres

- 23.3 Mostrar corredor endémico acumulativo por enfermedades
- 23.4 Mostrar cálculo de tasas y suma por enfermedades.
- 23.5 Mostrar Cálculo de la Transformación logarítmica de las tasas
- 23.6 Mostrar Cálculo de medias. DE e IC 95%
- 23.7 Mostrar transformación a unidades originales, menos 1 y cálculo de casos.
- 23.8 Mostrar Cálculo de las diferencias
- 23.9 Mostrar reporte gráfico del Corredor endémico de Enfermedades según prevalencia acumulada
- 23.10 Mostrar reporte gráfico en la tendencia del comportamiento de las enfermedades, según su tipo: Éxito, Seguridad o Alerta.

24-Cerrar sesión.

2.6.3 *Requerimientos no funcionales*

Requerimientos de Apariencia o Interfaz Externa

- La interfaz estará diseñada de modo tal que el usuario pueda tener en todo momento el control de la aplicación, lo que le permitirá ir de un punto a otro dentro de ella con gran facilidad. Se cuidará porque la aplicación sea lo más interactiva posible.

Requerimientos de Usabilidad

- El producto está orientado para ser usado por diferentes tipos de usuarios, entre los que se encuentran aquellas personas que tengan poca experiencia en el uso de la computadora y que sólo necesitarían un ligero entrenamiento sobre el funcionamiento de los principales elementos de una interfaz estándar en el ambiente del sistema operativo Windows (uso del Mouse, manejo de menús, botones y cuadros de texto).

Requerimientos de Rendimiento

- El sistema propuesto debe ser rápido en el procesamiento de la información así como a la hora de dar respuesta a la solicitud de los usuarios, además debe permitir el acceso simultáneo a los datos por diferentes usuarios.

Todo esto depende en gran medida del uso que se le da a los recursos que se disponen en el modelo Cliente/Servidor y de la velocidad de las consultas en la Base de Datos. El sistema deberá recuperarse en un corto período de tiempo ante cualquier falla.

Requerimientos de Soporte

- Se requiere un servidor de bases de datos con soporte para grandes volúmenes de información, velocidad de procesamiento y tiempo de respuesta rápido en accesos concurrentes. Se documentará la aplicación para garantizar su soporte.
- El administrador tendrá la responsabilidad de mantener actualizada la aplicación.
- El sistema deberá permitir futuras mejoras y nuevas opciones que se le quieran incorporar.

Requerimientos de Portabilidad

- La plataforma seleccionada para desarrollar la aplicación es Windows, pero puede ser ejecutada desde otras plataformas que soporten el lenguaje PHP, por ejemplo Linux.

Requerimientos de Seguridad

- No se permitirán accesos sin autorización al sistema, para ello se definirá una política de usuarios con roles y privilegios diferentes para que la información pueda ser consultada por las personas autorizadas. Para ello el sistema contará con cuatro tipos de usuarios, el administrador, el investigador, el director y los usuarios, que según el rol que tengan asignado en el sistema será el acceso a la parte de la información que le corresponde.
- Debe garantizar la conectividad e integridad de los datos almacenados a través de la red. Esto está garantizado por el Sistema Operativo.
- Debe garantizar la confidencialidad para proteger la información de acceso no autorizado. Esto estará garantizado por el Sistema Gestor de Base de Datos.

Confiabilidad

➤ Es importante que el sistema presente un mecanismo de respuesta rápida ante fallos y que en caso de ocurrencia se minimicen las pérdidas de información.

Requerimientos de Ayuda y Documentación en Línea

➤ El sistema contará con una ayuda que explicará de manera clara y sencilla al usuario todas las funcionalidades del sistema y esta estará disponible en todas las ventanas del sitio.

Requerimientos de Software

➤ Se debe disponer de un sistema operativo compatible. El sistema debe poderse ejecutar en entornos Windows y/o Linux (Multiplataforma).

➤ Del lado del servidor se utilizará Wamp como servidor Web, PHP 5.3.5 o superior como lenguaje de programación, Postgres SQL como sistema gestor de base de datos para garantizar la integridad y consistencia de la información.

➤ Del lado del cliente se requiere un navegador que interprete las funciones básicas de Java Script, se recomienda Mozilla Firefox 3.0 o superior.

Requerimientos de Hardware

Para el desarrollo y puesta en práctica del proyecto se requieren máquinas con los siguientes requisitos:

- Procesador de 400MHz y 128MB de RAM.
- Requiere 35 MB de espacio de disco duro.
- Navegador Web.
- Periférico de comunicación: Módem y tarjeta de red.

Las computadoras situadas en los puestos de trabajo de los usuarios requerirán como mínimo:

- Un Procesador Pentium III

- 128 Mb de Memoria RAM
- Deben estar conectadas en red con el servidor a través de una tarjeta de red de 100 Mbps
- Teclado y Mouse.

2.7 Descripción del sistema propuesto

2.7.1 Modelo de casos de uso del sistema

El modelo de casos de uso permite que los desarrolladores del software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. Describe lo que hace el sistema para cada tipo de usuario (Jacobson et al., 2006).

2.7.2 Actores del sistema

Un actor no es más que un rol que juega un usuario de Caso de Uso cuando interaccionan con estos casos de uso. Los actores representan a terceros fuera del sistema que colaboran con el mismo. Una vez que se han identificado los actores del sistema, se ha identificado el entorno externo del sistema. (Jacobson, Booch, & Rumbaugh, 2006).

Actor del Sistema	Justificación
Administrador	Es el encargado de mantener la seguridad del sistema a través del control de los diferentes accesos, pues es quién tiene todos los permisos para trabajar con el software y entre otras cosas es el responsable de registrar, modificar, eliminar usuarios o restaurar sus contraseñas.
Bioestadístico	Consulta, inserta y revisa la Información de acuerdo con los permisos que tenga.
Epidemiólogo	Es el encargado de revisar toda la información emitida en los reportes de las enfermedades y sus cálculos estadísticos y su función es dar el parte correspondiente a las autoridades de salud.

Tabla 4 Actores del sistema.

2.7.3 Casos de uso del sistema

Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo la relación y la generalización son relaciones. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona una respuesta a eventos que se producen en el mismo. En este tipo de diagrama intervienen algunos conceptos nuevos: un actor es una entidad externa al sistema que se modela y que puede interactuar con él; un ejemplo de actor podría ser un usuario o cualquier otro sistema. (mastermagazine, 2010)

Las relaciones entre casos de uso y actores pueden ser las siguientes:

- Un actor se comunica con un caso de uso.
- Un caso de uso extiende otro caso de uso.
- Un caso de uso usa otro caso de uso.

Dado el número de casos de uso, se introducen paquetes para lograr una mejor organización de los elementos en grupos. Sobre los paquetes del sistema, se asegura que subdividir los casos de uso en paquetes resulta de mucha ayuda en la modelación de cualquier sistema informático. Los paquetes son un mecanismo de organización de elementos que subdividen el modelo en otros más pequeños que colaboran entre sí. (mastermagazine, 2010)

2.8 Paquetes

Término que denota un mecanismo de propósito general para organizar en grupos los elementos. Los paquetes se pueden anidar dentro de otros paquetes. Un sistema puede corresponder a un único paquete de alto nivel, con todo el resto del modelo contenido recursivamente en él. En

un paquete pueden aparecer tanto elementos del modelo como diagramas. (Booc & Rumbaugh, 2006)

Como el modelo de casos de uso es grande, es decir, contiene un gran número de casos de uso y/o actores, también puede ser útil introducir paquetes en el modelo.

2.8.1 *Diagrama de casos de uso del sistema por paquetes*

Es un diagrama que muestra las relaciones existentes entre actores y casos de uso dentro de un sistema. (Booc & Rumbaugh, 2006)

Se realiza el diagrama de casos de uso del sistema por paquetes, en los anexos se muestran los diagramas de casos de uso del sistema correspondiente al paquete “Seguridad”, “Gestionar Información” y “Reportes”.

2.8.2 *Descripción de los casos de uso del sistema agrupados por paquetes*

2.8.3 *Seguridad*

En el paquete Seguridad se tienen los siguientes Casos de uso:

1. Iniciar sesión.
2. Gestionar usuarios.
3. Cambiar contraseña.
4. Cerrar sesión.

CU 1	Iniciar sesión
Actores	Bioestadístico, administrador
Propósito	Acceder a la aplicación

Resumen	
El caso de uso inicia cuando el Bioestadístico o el Administrador desea acceder a la aplicación para lo cual tiene que introducir su nombre de usuario y contraseña. El caso de uso finaliza cuando se verifica que los datos son correctos permitiendo la entrada al mismo según los permisos que tenga asignados..	
Referencia	1
Prototipo	Anexo # 6

Tabla 2.7 Descripción del caso de uso del sistema “Iniciar sesión”.

CU 2	Gestionar usuario
Actores	Administrador
Propósito	Registrar, modificar, eliminar usuario, asignar usuario a la entidad o restaurar contraseña.
Resumen	
El caso de uso se inicia cuando el Administrador procede a registrar, modificar, eliminar usuario, asignar usuario a la entidad o restaurar contraseña. Los usuarios son registrados según su nombre, el rol que desempeña y se le asigna una contraseña. Para modificar un usuario se selecciona y se procede a realizar las modificaciones necesarias. Para eliminar un usuario se debe seleccionar el usuario que se desea borrar. Para asignar un usuario a la entidad a la cual pertenece se procede seleccionando el usuario y la entidad. Para restaurar la contraseña de un usuario se selecciona y se introduce la nueva contraseña. El caso de uso finaliza cuando se ha completado una de las operaciones que se describe anteriormente.	
Referencia	2

2.8 Descripción del caso de uso del sistema “Gestionar Usuario”.

CU 3	Cambiar contraseña
Actores	Administrador
Propósito	Cambiar la contraseña anterior.
Resumen	
<p>El caso de uso se inicia cuando el Bioestadístico o el Administrador desean cambiar su contraseña, para lo cual debe introducir su nombre de usuario, contraseña actual y la nueva contraseña. El caso de uso finaliza cuando la contraseña es cambiada satisfactoriamente. Para asignar un usuario a la entidad a la cual pertenece se procede seleccionando el usuario y la entidad. Para restaurar la contraseña de un usuario se selecciona y se introduce la nueva contraseña. El caso de uso finaliza cuando se ha completado una de las operaciones que se describe anteriormente.</p>	
Referencia	3

Tabla 2.9 Descripción del caso de uso del sistema “Cambiar contraseña”.
Caso de uso: #3 Cambiar contraseña.

CU 4	Cerrar sesión.
Actores	Bioestadístico o el Administrador.
Propósito	Salir de la aplicación.
Resumen	
<p>El caso de uso se inicia cuando el Bioestadístico Supervisor, Técnico de Informática de la APPP, Responsable de Seguridad Informática de la APPM, Jefe de Seguridad Informática de la APPP o el Administrador selecciona la opción de cerrar sesión. El caso de uso finaliza cuando se cierra la sesión mostrando la página de inicio.</p>	
Referencia	24

2.10 Descripción del caso de uso del sistema “Cerrar Sesión”.

Gestión.

En el paquete Gestión se tienen los siguientes casos de uso:

- 1 Gestionar provincia.
- 2 Gestionar municipio.
- 3 Gestionar Área de salud.
- 4 Gestionar Enfermedades.
- 5 Gestionar Caso.
- 6 Gestionar Tasa.
- 7 Gestionar Semana.
- 8 Gestionar prevalencia acumulada.
- 9 Gestionar la Transformación logarítmica de las tasas.
- 10 Gestionar Cálculo de medias. Desviación estándar e intervalos de confianza al 95%
- 11 Gestionar transformación a unidades originales, menos 1 y cálculo de casos.
- 12 Gestionar Cálculo de las diferencias
- 13 Gestionar reportes

CU 5	Gestionar provincia.
Actores	Administrador.
Propósito	Registrar, eliminar o modificar una provincia determinada.
Resumen	El caso de uso se inicia cuando el Administrador procede a registrar, modificar o eliminar una provincia. Para registrar se introduce el nombre la provincia. Cuando se desea modificar se selecciona la provincia y se realiza la modificación. Para eliminar simplemente se escoge la provincia y se procede a eliminar. El caso de uso finaliza cuando se ha realizado una de las operaciones descritas anteriormente.
Referencia	5
Prototipo	Anexo 7

2.10 Descripción del caso de uso del sistema "Gestionar Provincia".

CU 6	Gestionar municipio.
Actores	Administrador.
Propósito	Registrar, eliminar o modificar un municipio.
Resumen	
El caso de uso se inicia cuando el Administrador procede a registrar, modificar o eliminar un municipio. Para registrar se introduce el nombre del municipio. Cuando se desea modificar se selecciona el municipio y se realiza la modificación. Para eliminar simplemente se escoge el municipio y se procede a eliminar. El caso de uso finaliza cuando se ha realizado una de las operaciones descritas anteriormente.	
Referencia	6

2.11 Descripción del caso de uso del sistema “Gestionar Municipio”.

CU 7	Gestionar área de salud.
Actores	Administrador.
Propósito	Registrar, eliminar o modificar un área de salud.
Resumen	
El caso de uso se inicia cuando el Administrador procede a registrar, modificar o eliminar un área de salud. Para registrar se introduce el nombre del área de salud. Cuando se desea modificar se selecciona el área de salud y se realiza la modificación. Para eliminar simplemente se escoge el área de salud y se procede a eliminar. El caso de uso finaliza cuando se ha realizado una de las operaciones descritas anteriormente.	
Referencia	7

2.12 Descripción del caso de uso del sistema “Gestionar Área de Salud”.

CU 8	Gestionar enfermedad.
Actores	Usuario.
Propósito	Registrar, eliminar o modificar una enfermedad.
Resumen	
El caso de uso se inicia cuando el Usuario procede a registrar, modificar o eliminar alguna enfermedad. Para registrar se introduce el nombre de la	

enfermedad. Cuando se desea modificar se selecciona la enfermedad y se realiza la modificación. Para eliminar simplemente se escoge el nombre de la enfermedad y se procede a eliminar. El caso de uso finaliza cuando se ha realizado una de las operaciones descritas anteriormente.

Referencia	8
Prototipo	Anexo 8

2.13 Descripción del caso de uso del sistema “Gestionar enfermedad”.

CU 9	Gestionar caso por enfermedades.
Actores	Usuario.
Propósito	Registrar, eliminar o modificar un caso.
Resumen	
El caso de uso se inicia cuando el Usuario procede a registrar, modificar o eliminar algún caso. Para registrar se introduce el nombre del caso. Cuando se desea modificar se selecciona el nombre del caso y se realiza la modificación. Para eliminar simplemente se escoge el nombre del caso y se procede a eliminar. El caso de uso finaliza cuando se ha realizado una de las operaciones descritas anteriormente.	
Referencia	9
Prototipo	Anexo 9

2.14 Descripción del caso de uso del sistema “Gestionar Casos por enfermedades”.

CU 10	Gestionar tasa por enfermedades.
Actores	Usuario.
Propósito	Registrar, eliminar o modificar tasa por enfermedades.
Resumen	
El caso de uso se inicia cuando el Usuario procede a calcular la tasa de	

incidencia de una enfermedad. Una vez introducido los datos referente a la enfermedad se usa una fórmula matemática y se obtiene la tasa. El caso de uso termina cuando se calcula correctamente la tasa de incidencia de las enfermedades introducidas.	
Referencia	10

2.15 Descripción del caso de uso del sistema “Gestionar Tasa por enfermedades”.

CU 11	Gestionar semana de incidencia.
Actores	Usuario.
Propósito	Registrar, eliminar o modificar la semana de incidencia.
Resumen El caso de uso se inicia cuando el Usuario procede a registrar, modificar o eliminar alguna semana. Para registrar se introduce el número id del dato. Cuando se desea modificar se selecciona el id de dato poblacional y se realiza la modificación. Para eliminar simplemente se escoge el id del dato y se procede a eliminar. El caso de uso finaliza cuando se ha realizado una de las operaciones descritas anteriormente.	
Referencia	11

2.16 Descripción del caso de uso del sistema “Gestionar Semana”.

CU 12	Gestionar datos estadísticos de la población.
Actores	Usuario.
Propósito	Registrar, eliminar o modificar los datos estadísticos de la población.
Resumen El caso de uso se inicia cuando el Usuario procede a registrar, modificar o eliminar algún dato estadístico de la población. Para registrar se introduce el número de la semana. Cuando se desea modificar se selecciona el número de la semana y se realiza la modificación. Para eliminar simplemente se escoge la semana en cuestión y se procede a eliminar. El caso de uso	

finaliza cuando se ha realizado una de las operaciones descritas anteriormente.

Referencia	12
-------------------	-----------

2.17 Descripción del caso de uso del sistema “Gestionar Población”.

CU 13	Gestionar la Transformación logarítmica de las tasas
Actores	Usuario.
Propósito	Calcular la transformación logarítmica de las tasas.
Resumen	
El caso de uso se inicia cuando el Usuario procede a calcular la transformación logarítmica de las tasas de una enfermedad. Una vez introducido los datos referente a la enfermedad se usa una fórmula matemática y se obtiene la transformación logarítmica de las tasas. El caso de uso termina cuando se calcula transformación logarítmica de las tasas correctamente.	
Referencia	13
Prototipo	Anexo 10.

2.18 Descripción del caso de uso del sistema “Gestionar Tranf Log Tasa”.

CU 14	Gestionar Cálculo de medias. Desviación estándar e intervalos de confianza al 95%.
Actores	Usuario.
Propósito	Gestionar Cálculo de medias. Desviación estándar e intervalos de confianza al 95%.
Resumen	
El caso de uso se inicia cuando el Usuario procede al cálculo de medias, Desviación estándar e intervalos de confianza al 95%.de una enfermedad. Una vez introducido los datos referente a la enfermedad se usa una fórmula matemática y se obtiene el resultado del cálculo de medias. Desviación estándar e intervalos de confianza al 95%.de una enfermedad. El caso de uso termina cuando se calcula correctamente las medias y la desviación estándar e intervalos de confianza al 95%.de dicha enfermedad.	
Referencia	14
Prototipo	Anexo 11

--	--

2.19 Descripción del caso de uso del sistema “Gestionar Calc Media, DE e IC”.

CU 15	Gestionar transformación a unidades originales, y cálculo de casos.
Actores	Usuario.
Propósito	Calcular la transformación a unidades originales, menos 1 y cálculo de casos.
Resumen	
El caso de uso se inicia cuando el Usuario procede Gestionar transformación a unidades originales, menos 1 y el cálculo de casos de una enfermedad. Una vez introducido los datos referente a la enfermedad se usa una fórmula matemática y se obtiene la transformación a unidades originales, menos 1 y cálculo de casos. El caso de uso termina cuando se calcula correctamente la transformación a unidades originales, menos 1 y el cálculo de casos.	
Referencia	15
Prototipo	Anexo 12

2.20 Descripción del caso de uso del sistema “Gestionar Tranf a U. originales”.

CU 17	Gestionar Cálculo de las diferencias
Actores	Usuario.
Propósito	Cálculo de las diferencias.
Resumen	
El caso de uso se inicia cuando el Usuario procede a calcular las diferencias de una enfermedad. Una vez introducido los datos referente a la enfermedad, se usa una fórmula matemática y se obtiene el cálculo de las diferencias. El caso de uso termina cuando se calcula las diferencias entre las enfermedades correctamente.	
Referencia	16
Prototipo	Anexo 13

--	--

2.21 Descripción del caso de uso del sistema “Gestionar Calc. Diferencias”.

CU 19	Gestionar Cálculo de la prevalencia
Actores	Usuario.
Propósito	Cálculo de la prevalencia de las enfermedades.
Resumen	
El caso de uso se inicia cuando el Usuario procede al cálculo de la prevalencia de las enfermedades. Una vez introducido los datos referente a la enfermedad, se usa una fórmula matemática y se obtiene el cálculo de la prevalencia de las enfermedades. El caso de uso termina cuando se obtiene el cálculo de la prevalencia de las enfermedades correctamente.	
Referencia	17

2.22 Descripción del caso de uso del sistema “Gestionar Calc. Prevalencia”.

CU 20	Gestionar Cálculo de tasas y suma 1.
Actores	Usuario.
Propósito	Cálculo de tasas y suma 1.
Resumen	
El caso de uso se inicia cuando el Usuario procede al cálculo de tasas y suma 1 en las enfermedades. Una vez introducido los datos referente a la enfermedad, se usa una fórmula matemática y se obtiene el cálculo de tasas y suma 1 en las enfermedades. El caso de uso termina cuando se obtiene el cálculo de tasas y suma 1 en las enfermedades correctamente.	
Referencia	18

2.23 Descripción del caso de uso del sistema “Gestionar Calc. Tasa y Suma 1”.

2.8.4 *Reportes*

En el paquete **Reportes** se tienen los siguientes casos de uso:

- 1 Mostrar enfermedades.
- 2 Mostrar casos y tasas por trimestres
- 3 Mostrar corredor endémico acumulativo por enfermedades
- 4 Mostrar de tasas y suma por enfermedades.
- 5 Mostrar Cálculo de la Transformación logarítmica de las tasas
- 6 Mostrar Cálculo de medias. DE e IC 95%
- 8 Mostrar transformación a unidades originales, menos 1 y cálculo de casos.
- 7 Mostrar Cálculo de las diferencias
- 8 Mostrar reporte gráfico del Corredor endémico de Enfermedades según prevalencia acumulada
- 9 Mostrar reporte gráfico en la tendencia del comportamiento de las enfermedades.

CU 21	Mostrar enfermedades
Actores	Usuario.
Propósito	Mostrar listado de enfermedades.
Resumen	
El caso de uso inicia cuando el Usuario necesita tener información detallada de las enfermedades reportadas por los centros hospitalarios, seleccionando la opción correspondiente a "Listado de Enfermedades". El caso de uso finaliza cuando se muestra el listado de todas las enfermedades.	
Referencia	23.1

2.24 Descripción del caso de uso del sistema "Mostrar Enfermedades".

CU 22	Mostrar casos y tasas por trimestres.
Actores	Usuario.
Propósito	Mostrar casos y tasas por trimestres.
Resumen	
El caso de uso inicia cuando el Usuario necesita tener información detallada de los casos y tasas por trimestres, seleccionando la opción correspondiente a "Listado de Casos y Tasas por Trimestres". El caso de	

uso finaliza cuando se muestra el listado de casos y tasas por trimestres.	
Referencia	23.2

2.25 Descripción del caso de uso del sistema “Mostrar Casos y Tasas”.

CU 23	Mostrar corredor endémico acumulativo por enfermedades.
Actores	Usuario.
Propósito	Mostrar corredor endémico acumulativo por enfermedades.
Resumen	
El caso de uso inicia cuando el Usuario necesita tener información detallada del corredor endémico acumulativo por enfermedades, seleccionando la opción correspondiente a “Listado de corredores endémico acumulativo por enfermedades”. El caso de uso finaliza cuando se muestra el listado de los corredores endémico acumulativo por enfermedades.	
Referencia	23.4

2.26 Descripción del caso de uso del sistema “Corredor Endémico Acumulativo”.

CU 24	Mostrar tasas y suma por enfermedades.
Actores	Usuario.
Propósito	Mostrar tasas y suma por enfermedades.
Resumen	
El caso de uso inicia cuando el Usuario necesita tener información detallada del corredor endémico acumulativo por enfermedades, seleccionando la opción correspondiente a “Listado de corredores endémico acumulativo por enfermedades”. El caso de uso finaliza cuando se muestra el listado de los corredores endémico acumulativo por enfermedades.	
Referencia	23.5

2.27 Descripción del caso de uso del sistema “Mostrar Tasas y suma por enfermedades”.

CU 25	Mostrar cálculo de la Transformación logarítmica
--------------	--

	de las tasas.
Actores	Usuario.
Propósito	Mostrar cálculo de la Transformación logarítmica de las tasas.
Resumen	
El caso de uso inicia cuando el Usuario necesita tener información detallada del cálculo de la Transformación logarítmica de las tasas por enfermedades, seleccionando la opción correspondiente a “Listar Cálculo de la Transformación logarítmica de las tasas por enfermedades”. El caso de uso finaliza cuando se muestra el listado del cálculo de la Transformación logarítmica de las tasas.	
Referencia	23.6

2.28 Descripción del caso de uso del sistema “Mostrar Tranf Log de Tasas”.

CU 26	Mostrar cálculo de medias. Desviación estándar e intervalos de confianza al 95%.
Actores	Usuario.
Propósito	Mostrar cálculo de medias. Desviación estándar e intervalos de confianza al 95%.
Resumen	
El caso de uso inicia cuando el Usuario necesita tener información detallada del cálculo de medias. Desviación estándar e intervalos de confianza al 95%. por enfermedades, seleccionando la opción correspondiente a “Listar Cálculo de la Transformación logarítmica de las tasas por enfermedades”. El caso de uso finaliza cuando se muestra el listado del cálculo de la Transformación logarítmica de las tasas.	
Referencia	23.7

2.29 Descripción del caso de uso del sistema “Mostrar Calculo de Medias, DE e IC”.

CU 27	Mostrar transformación a unidades originales, menos 1 y cálculo de casos.
Actores	Usuario.
Propósito	Mostrar transformación a unidades originales, menos 1 y cálculo de casos.
Resumen	
El caso de uso inicia cuando el Usuario necesita tener información detallada de la transformación a unidades originales, menos 1 y cálculo de casos por enfermedades, seleccionando la opción correspondiente a “Listar transformación a unidades originales, menos 1 y cálculo de casos”. El caso de uso finaliza cuando se muestra el listado de la transformación a unidades originales, menos 1 y cálculo de casos.	
Referencia	23.8

2.30 Descripción del caso de uso del sistema “Mostrar Tranf a unidades”.

CU 28	Mostrar cálculo de las diferencias.
Actores	Usuario.
Propósito	Mostrar cálculo de las diferencias.
Resumen	
El caso de uso inicia cuando el Usuario necesita tener información detallada del cálculo de las diferencias por enfermedades, seleccionando la opción correspondiente a “Listar cálculo de las diferencias por enfermedades”. El caso de uso finaliza cuando se muestra el listado del cálculo de las diferencias por enfermedades.	
Referencia	23.9

2.31 Descripción del caso de uso del sistema “Mostrar Calc de Diferencias”.

CU 29	Mostrar reporte gráfico del Corredor endémico por Enfermedades
Actores	Usuario.
Propósito	Mostrar reporte gráfico del Corredor endémico de Enfermedades

Resumen	
El caso de uso inicia cuando el Usuario necesita ver el reporte gráfico del Corredor endémico de Enfermedades, seleccionando la opción correspondiente a “Mostrar reporte gráfico del Corredor endémico de Enfermedades”. El caso de uso finaliza cuando se muestra reporte gráfico del Corredor endémico de Enfermedades.	
Referencia	23.10

2.32 Descripción del caso de uso del sistema “Mostrar Corredor Endémico”.

CU 30	Mostrar reporte gráfico en la tendencia del comportamiento de las enfermedades, según su tipo: Éxito, Seguridad o Alerta.
Actores	Usuario.
Propósito	Mostrar reporte gráfico en la tendencia del comportamiento de las enfermedades, según su tipo: Éxito, Seguridad o Alerta.
Resumen	
El caso de uso inicia cuando el Usuario necesita ver el reporte gráfico en la tendencia del comportamiento de las enfermedades, según su tipo: Éxito, Seguridad o Alerta por enfermedades, seleccionando la opción correspondiente a “Mostrar reporte gráfico en la tendencia del comportamiento de las enfermedades, según su tipo: Éxito, Seguridad o Alerta por Enfermedades”. El caso de uso finaliza cuando se muestra el reporte gráfico en la tendencia del comportamiento de las enfermedades, según su tipo: Éxito, Seguridad o Alerta por enfermedades.	
Referencia	23.11

2.33 Descripción del caso de uso del sistema “Mostrar Tendencia de enfermedades”.

2.9 *Diagramas de paquetes del sistema*

Figura 4 Relación entre los paquetes del sistema. Ver Anexo 14

Figura 5 Diagrama de Casos de Uso: Paquete Gestión de la información. Ver anexo 15

Figura 6 Diagrama de Casos de Uso: Paquete Seguridad. Ver Anexo 16

Figura 7 Diagrama de Caso de Uso: Paquete Reportes. Ver Anexo 17

2.10 *Conclusiones:*

En este capítulo se realizó un análisis para conocer cómo funciona el negocio en el Centro de Higiene y Epidemiología, se estudiaron los procesos que lo conforman, lo que permitió modelarlo mediante casos de uso, actores, trabajadores y reglas. Se definieron los principales requerimientos funcionales y no funcionales del sistema, así como los actores y su relación con estos, se definieron los paquetes de casos de uso que permiten agruparlos según su funcionalidad y por último se realiza una descripción de cada caso de uso.

Capítulo 3

Construcción de una Aplicación Web para la gestión y control de datos estadísticos de enfermedades en el Centro de Higiene y Epidemiología.

3.1 Introducción. En el diseño modelamos el sistema y encontramos su forma (incluida la arquitectura) para que soporte los requisitos funcionales, no funcionales y otras restricciones que se le suponen. (Booc & Rumbaugh, 2006)

Los objetivos que perseguimos en el diseño son: realizar el diagrama de clases de diseño, diseñar la base de datos, definir los principios de diseño y realizar el modelo de despliegue. En la implementación empezamos con el resultado del diseño e implementamos el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. (Booc & Rumbaugh, 2006)

El objetivo de la implementación es: Realizar el modelo de implementación.

3.1 Diagrama de clases del diseño.

Un diagrama de clases es una presentación gráfica de la vista estática, que muestra una colección de elementos declarativos (estáticos) del modelo, como clases, tipos, y sus contenidos y relaciones. Puede mostrar una vista de un paquete y contener símbolos de paquetes anidados. Contiene ciertos elementos materializados de comportamiento, como operaciones, pero cuya dinámica está representada en otros diagramas, como diagramas de estados o diagramas de colaboración. (Booc & Rumbaugh, 2006)

Un diagrama de clases de diseño representa las especificaciones de las clases e interfaces software en una aplicación. Donde podemos encontrar: (Laman, 1999).

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Información acerca del tipo de los atributos.
- Navegabilidad.
- Dependencias.

Las clases de diseño de los diagramas de clases de diseño solo muestran las definiciones de las clases software en lugar de los conceptos del mundo real.

Diagrama de Clases de Diseño
Paquete Seguridad
Diagrama de clases de diseño CU: "Iniciar sesión".
Diagrama de clases de diseño CU: "Gestionar usuarios".
Diagrama de clases de diseño CU: "Cambiar contraseña".
Diagrama de clases de diseño CU: "Cerrar sesión".

Diagrama de Clases de Diseño
Paquete Gestión
Diagrama de clases de diseño CU: "Gestionar provincia".
Diagrama de clases de diseño CU: "Gestionar municipio".
Diagrama de clases de diseño CU: "Gestionar Área de salud".
Diagrama de clases de diseño CU: "Gestionar Enfermedades".
Diagrama de clases de diseño CU: "Gestionar Caso".
Diagrama de clases de diseño CU: "Gestionar Tasa".
Diagrama de clases de diseño CU: "Gestionar Semana".
Diagrama de clases de diseño CU: "Gestionar prevalencia acumulada".
Diagrama de clases de diseño CU: "Gestionar la Transformación logarítmica de las tasas".
Diagrama de clases de diseño CU: "Gestionar Cálculo de medias. Desviación estándar e intervalos de confianza al 95%".
Diagrama de clases de diseño CU: "Gestionar transformación a unidades originales, menos 1 y cálculo de casos".

Diagrama de clases de diseño CU: “Gestionar Cálculo de las diferencias”.

Diagrama de clases de diseño CU: “Gestionar reportes”.

Diagrama de Clases de Diseño

Paquete reportes

Diagrama de clases de diseño CU: “Mostrar casos y tasas por trimestres”.

Diagrama de clases de diseño CU: “Mostrar corredor endémico acumulativo por enfermedades”.

Diagrama de clases de diseño CU: “Mostrar de tasas y suma por enfermedades.”.

Diagrama de clases de diseño CU: “Mostrar Cálculo de la Transformación logarítmica de las tasas”.

Diagrama de clases de diseño CU: “Mostrar Cálculo de medias. DE e IC 95%”.

Diagrama de clases de diseño CU: “Mostrar transformación a unidades originales, menos 1 y cálculo de casos.”.

Diagrama de clases de diseño CU: “Mostrar Cálculo de las diferencias”.

Diagrama de clases de diseño CU: “Mostrar reporte gráfico del Corredor endémico de Enfermedades según prevalencia acumulada”.

Diagrama de clases de diseño CU: “Mostrar reporte gráfico en la tendencia del Comportamiento de las enfermedades”.

3.2 *Diseño de la base de datos.*

Las bases de datos necesitan de una definición de su estructura que le permitan almacenar datos, reconocer el contenido y recuperar la información. La estructura tiene que ser desarrollada para la necesidad de las aplicaciones que la usarán, esto ayuda a realizar un proceso del negocio para alcanzar un valor agregado para el cliente. La puesta en práctica de la base de datos es el paso final en el desarrollo de aplicaciones de soporte del negocio. Tiene que conformarse con los requisitos del proceso del negocio, que es la primera abstracción de la vista de la base de datos.

3.2.1 *Diagrama de clases persistentes*

El diagrama del modelo lógico de datos o diagrama de clases persistentes, muestra las clases capaces de mantener su valor en el espacio y en el tiempo. (Booc & Rumbaugh, 2006)

Ver Anexo # 22

3.3 *Modelo de datos*

El modelo físico de datos o modelo de datos, representa la estructura o descripción física de las tablas de la base de datos y se obtiene a partir del modelo lógico de datos. (Booc & Rumbaugh, 2006).

Ver Anexo # 23

3.4 *Principios de diseño*

El diseño de una aplicación Web es un trabajo difícil ya que debe cumplir de la interfaz de un sitio Web, el formato de los reportes, la concepción de la ayuda y el tratamiento de excepciones tiene gran influencia en el éxito o fracaso del mismo. A continuación se describen los principios de diseño seguidos para el desarrollo de la aplicación en cuestión.

3.4.1 *Estándares en la interfaz de la aplicación.*

Se presenta una interfaz amigable, atractiva, sencilla, agradable, legible y de fácil uso, para lo cual se tienen en cuenta varios aspectos, sobre todo relacionados con tipografías, colores y navegación. Se han utilizado principalmente tonalidades de azul y blanco por ser suaves y refrescantes. El vocabulario utilizado es de fácil comprensión para el usuario, sin emplear palabras técnicas de informática. El tipo de letra utilizada es Times New Roman, el tamaño de letra que predomina es de 12 píxeles para mantener la uniformidad de la aplicación. La navegación guía al usuario de una página a otra. La entrada de información por parte de los usuarios se realiza a través de formularios. Las interacciones con el

sistema se basan fundamentalmente en selecciones de menú, en sus distintas formas y en acciones sobre elementos de código visual como botones, y mensajes.

3.4.2 *Formato de reportes.*

Los reportes serán concebidos utilizando un formato de letra clara, legible y con colores agradables a la vista del usuario, para garantizar su correcta visualización y la calidad de su impresión. Cada reporte presenta un encabezado que lo identifica y seguidamente se muestra la información organizada en forma de tabla para una mejor comprensión, y será mostrado en la pantalla como un documento de formato PDF, igual que su respectivo archivo para su envío a las autoridades sanitarias.

3.5 *Concepción general de la ayuda*

La ayuda constará en gran parte de la explicación funcional de la aplicación, con el objetivo de guiar al usuario y proporcionar información necesaria, con explicaciones sencillas y de fácil comprensión. Formada por un menú general que contiene todas las funcionalidades de la aplicación. El usuario podrá navegar por cada una de ellas de forma sencilla, pues está descrita de un modo fácil de entender.

3.5.1 *Tratamiento de errores*

En el sistema propuesto se evitan, minimizan y tratan los posibles errores, con el fin de garantizar la integridad y confiabilidad de la información, ya que muchas veces el usuario teclea datos y en otras ocasiones selecciona elementos, para reducir las posibilidades de introducir información errónea se realizan las validaciones del lado del cliente mediante JavaScript evitando hacer peticiones innecesarias al servidor, los errores se le comunican al usuario mostrando mensajes claros, precisos y de fácil comprensión.

3.6 Seguridad

Para garantizar la seguridad y protección de la información se tienen en cuenta diferentes medidas entre las cuales podemos mencionar:

Garantizar el control de los diferentes accesos, sólo pueden acceder a la aplicación los usuarios registrados, ya que se requiere un nombre de usuario y contraseña. Encriptar las contraseñas almacenadas en la base de datos, para lo cual se utiliza un algoritmo de encriptación de un solo sentido, es decir que si alguien puede acceder a ellas no puede descifrarlas.

Se define una política de usuarios con privilegios de acuerdo al rol que desempeña, lo que asegura que la información pueda ser consultada de acuerdo a su nivel de acceso. Almacenar en la base de datos las trazas de los usuarios que registran, modifican o eliminan información correspondiente a los equipos informáticos y sus componentes.

3.6.1 Estándares de codificación

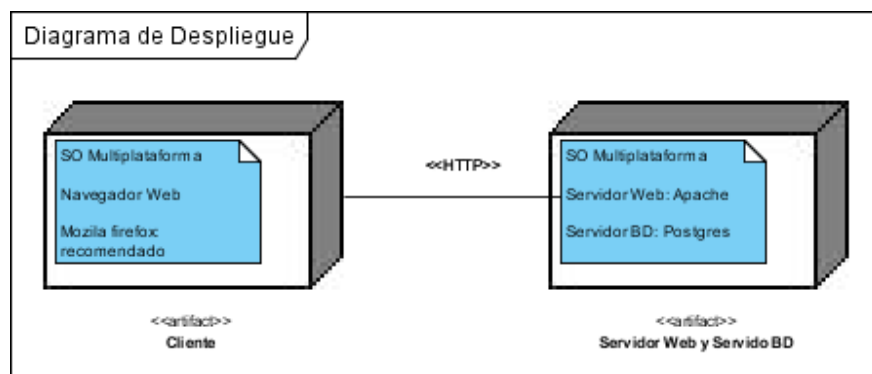
Establecer estándares de codificación es una buena forma de programación ya que un código comprensible permite reducir el tiempo y esfuerzo a la hora de realizar alguna modificación al mismo, por lo que establecer un estilo de programación es de suma importancia, algunos de los aspectos a tener en cuenta son: utilizar nombres descriptivos, el uso de una notación coherente y comentarios informativos. Las clases, nombres de variables, nombre de funciones, entre otros son cortos, claros y describen su propósito. Los operadores utilizados se separan por un espacio antes y uno después de estos y en caso necesario se añade un comentario al código para ayudar a entender su objetivo.

3.7 Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño. (Booc & Rumbaugh, 2006)

En el modelo de despliegue podemos observar lo siguiente: (Booc & Rumbaugh, 2006)

- Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar.
- Los nodos poseen relaciones que representan medios de comunicación entre ellos, tales como internet, intranet, bus y similares.
- El modelo de despliegue puede describir configuraciones de red, incluidas las configuraciones para pruebas y para simulación.
- La funcionalidad (los procesos) de un nodo se definen por los componentes que se distribuyen sobre ese nodo.
- El modelo de despliegue en sí mismo representa una correspondencia entre la arquitectura software y la arquitectura del sistema (el hardware).
- Teniendo en cuenta la definición anterior se realiza el diagrama de despliegue.



3.8 Modelo de implementación

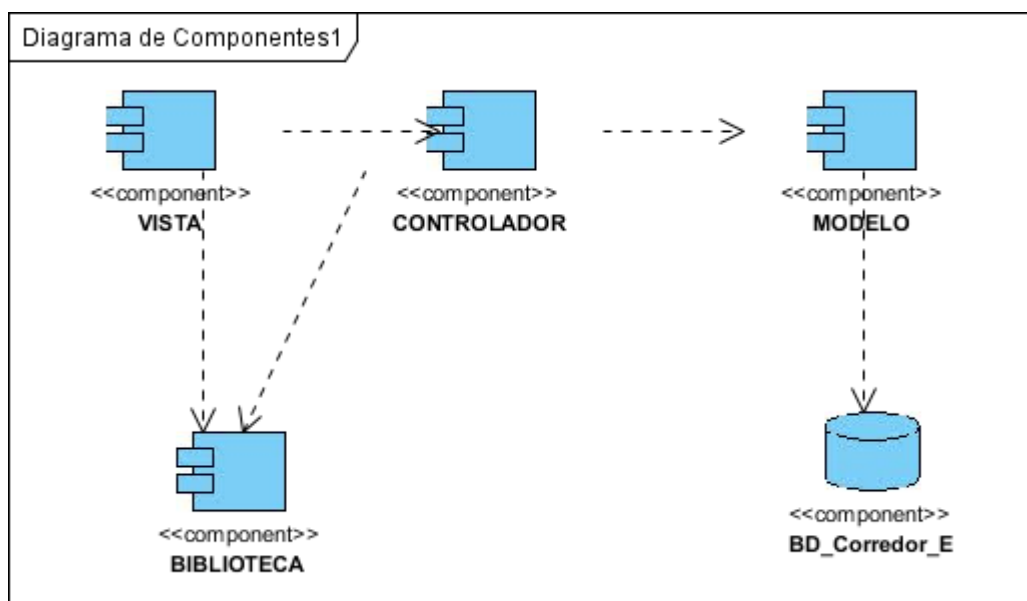
El modelo de implementación describe cómo los elementos de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, etc. Describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros. (Booc & Rumbaugh, 2006)Capítulo 3

3.9 Componente

Al empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño, se le denomina componente. (Booc & Rumbaugh, 2006)

Entre los estereotipos de componentes estándar encontramos los siguientes: (Booc & Rumbaugh, 2006)

- <<executable>> es un programa que puede ser ejecutado en un nodo.
- <<file>> es un fichero que contiene código fuente o datos.
- <<library>> es una librería estática o dinámica.
- <<table>> es una tabla de base de datos.
- <<document>> es un documento.



3.9.1 Descripción del modelo de componentes

- **Vista**: Paquete de componentes que agrupa los ficheros de interfaz de la aplicación o sea la vista codificada y mantiene la presentación final de la aplicación de cara al usuario.

- **Controlador:** Paquete de componentes que agrupa los ficheros de las clases controladoras, hace de enlace entre el modelo, la vista y cualquier otro recurso que se tenga que procesar en el servidor.
- **Modelo:** Paquete de componentes que agrupa los ficheros de las clases que tienen que ver con el acceso a la base de datos. En el modelo mantendremos encapsulada la complejidad de la base de datos y creamos funciones para recibir, insertar, actualizar o borrar información de las tablas. Al mantenerse todas las llamadas a la base de datos en un mismo código, desde otras partes del programa podremos invocar las funciones que necesitemos del modelo y éste se encargará de procesarlas.
- **BD_APPP:** Representa la base de datos de la aplicación para la cual se utilizó como gestor de base de datos o sistema de gestión de base de datos (SGBD o DBMS) PostgreSQL.
- **Librerías:** Paquete de componentes que agrupa las librerías utilizadas por la aplicación.

3.10 Conclusiones

En el diseño se realizó el diagrama de clases de diseño, se diseñó la base de datos, se describieron los principios de diseño seguidos, específicamente, estándares de la interfaz de usuario, el formato de reportes, el tratamiento de errores, la concepción general de la ayuda y los estándares de codificación y se confeccionó el modelo de despliegue. En la implementación se desarrolló el modelo de implementación donde se confeccionó el modelo de componentes.

Conclusiones

- 1- Los fundamentos teóricos metodológicos evidenciaron que para contribuir a la gestión y control de datos estadísticos de enfermedades en el Centro de Higiene y Epidemiología resulta oportuno la creación de una aplicación Web y para el desarrollo de la misma resultan apropiadas las herramientas: Visual Paradigm, Macromedia Dreamweaver, pgAdmin, SQL Manager 2007 for PostgreSQL y Help & Manual, como SGBD PostgreSQL y servidor Web Apache.
- 2- El diseño de la aplicación Web se basó en la utilización de la metodología RUP y el lenguaje UML, partiendo de la descripción de los procesos del negocio y la modelación del negocio, así como los requerimientos funcionales y no funcionales.
- 3- La aplicación Web se desarrolló utilizando como lenguaje de programación, del lado del cliente HTML, CSS y JavaScript, y del lado del servidor PHP basado en el patrón de diseño MVC, permitiendo la reutilización y mantenimiento del código, obteniendo como resultado una aplicación con una interfaz amigable con grandes facilidades de uso, así como la concepción general de la ayuda que orienta al usuario a cerca de las funcionalidades y de cómo trabajar con éstas, contribuyendo a la gestión y control de datos estadísticos de enfermedades en el Centro de Higiene y Epidemiología.

Bibliografía

- Booc, J., & Rumbaugh. (2006). *Conceptos generales de la Ingeniería del Software*. España: Nuevas Ediciones.
- A, P. V. (3 de agosto de 2013). *Ventajas y Desventajas de Wordpress*. Recuperado el 16 de octubre de 2013, de Ventajas y Desventajas de Wordpress: <http://www.plusvela.com/publicaciones/wordpress/ventajas-y-desventajas-de-wordpress>
- Aguilar, J. M. (3 de mayo de 2013). *ASP.NET MVC 2: Quince cuestiones que deberías conocer*. Recuperado el 16 de octubre de 2013, de ASP.NET MVC 2: Quince cuestiones que deberías conocer: <http://www.variablenotfound.com/2010/05/aspnet-mvc-2-quince-cuestiones-que.html#p1>
- Alvarez, M. A. (1 de enero de 2010). <http://www.desarrolloweb.com>. Recuperado el 22 de enero de 2014, de <http://www.desarrolloweb.com>: <http://www.desarrolloweb.com/articulos/que-es-html.html>
- Alvarez, M. A. (1 de enero de 2011). <http://www.desarrolloweb.com>. Recuperado el 22 de enero de 2014, de <http://www.desarrolloweb.com>: <http://www.desarrolloweb.com/articulos/332.php>
- Antonio, M. (28 de diciembre de 2013). www.importancia.org. Recuperado el 8 de enero de 2017, de www.importancia.org: www.importancia.org/control.php
- Avila, K. (13 de marzo de 2013). <http://www.cavsi.com>. Recuperado el 22 de enero de 2014, de <http://www.cavsi.com>: <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>
- Bardallo, C. (9 de octubre de 2012). *Aplicaciones Web*. Obtenido de Aplicaciones Web.
- Bardallo, C. (9 de octubre de 2012). *Aplicaciones Web*. Recuperado el 16 de octubre de 2013, de Aplicaciones Web: <http://appweblm.blogspot.com/2012/10/aplicaciones-web-definicion.html>

- Bizagi. (15 de junio de 2010). *http://help.bizagi.com*. Recuperado el 9 de enero de 2014, de http://help.bizagi.com:help.bizagi.com/bpmsuite/es/reutilizar_reglas_de_negocio.htm
- Borrego, D. J. (7 de julio de 2010). *BuenasTareas.com*. Recuperado el 2014 de enero de 8, de buenastareas.com
- Casablanca, A. C. (16 de julio de 2010). *MVC*. Recuperado el 16 de octubre de 2013, de *MVC*: <http://pormeterse.wordpress.com/2009/07/16/mvc-ventajas-e-inconvenientes/>
- Casablanca, A. C. (16 de julio de 2013). *MVC*. Recuperado el 16 de octubre de 2013, de *MVC*: <http://pormeterse.wordpress.com/2009/07/16/mvc-ventajas-e-inconvenientes/>
- Cazarez, c. (21 de noviembre de 2011). *http://iswitson.blogspot.com*. Recuperado el 22 de enero de 2014, de [http://iswitson.blogspot.com: http://iswitson.blogspot.com/2011/11/metodologia-rup.html](http://iswitson.blogspot.com:http://iswitson.blogspot.com/2011/11/metodologia-rup.html)
- Celaya, C. L. (3 de septiembre de 2006). *Descripción y Aplicaciones de la Web*. Recuperado el 16 de octubre de 2013, de *Descripción y Aplicaciones de la Web*: <http://zenweb20.blogspot.com/>
- Cuesta, G. (12 de febrero de 2011). *Definiciones apropiadas de Sitio web*. Recuperado el 16 de octubre de 2013, de *Definiciones apropiadas de Sitio web*: http://imaginar.org/iicd/index_archivos/TUS4/buscadores.pdf
- Fayol, H. (9 de septiembre de 2010). *definicion de control*. Recuperado el 8 de enero de 2014, de itescoingenieriaenadministracion.wikispaces.com: <http://itescoingenieriaenadministracion.wikispaces.com>
- Flanklin, T. &. (4 de abril de 2013). *www.sites.upiicsa.ipn.mx*. Recuperado el 8 de enero de 2014, de [www.sites.upiicsa.ipn.mx: www.sites.upiicsa.ipn.mx: http://www.sites.upiicsa.ipn.mx/polilibros/portal/Polilibros/P_proceso/ADMINISTRACION_II_TRANSPORT_Eva_Cruz_Maldonado/Polilibro/Unidad%205/5.3.htm](http://www.sites.upiicsa.ipn.mx:www.sites.upiicsa.ipn.mx: http://www.sites.upiicsa.ipn.mx/polilibros/portal/Polilibros/P_proceso/ADMINISTRACION_II_TRANSPORT_Eva_Cruz_Maldonado/Polilibro/Unidad%205/5.3.htm)
- Foundation, F. S. (4 de enero de 2014). *http://www.gnu.org/*. Recuperado el 9 de enero de 2010, de [http://www.gnu.org/ http://www.gnu.org/ http://www.gnu.org/philosophy/free-sw.es.html](http://www.gnu.org: http://www.gnu.org/ http://www.gnu.org/philosophy/free-sw.es.html)
- JHERSON, D. (1 de mayo de 2013). *herramientascascomparaciones*. Recuperado el 22 de enero de 2014, de

- herramientascasecomparaciones:
<http://herramientascasecomparaciones.blogspot.com/>
- Karim, G. (3 de octubre de 2011). *Sitios Web*. Recuperado el 16 de octubre de 2013, de Sitios Web: <http://www.wevxs.com/inegocios/sitios-web/que-es-un-sitio-web/>
- Karim, G. (03 de Octubre de 2011). *Sitios Web* . Recuperado el 16 de Octubre de 2013, de Sitios Web : <http://www.wevxs.com/inegocios/sitios-web/que-es-un-sitio-web/>
- Koontz, H. (9 de septiembre de 2010). *definicion del control*. Recuperado el 8 de enero de 2014, de itescoingenieriaenadministracion.wikispaces.com: <http://itescoingenieriaenadministracion.wikispaces.com>
- Linux, R. H. (23 de abril de 2013). *Tienda Linux*. Recuperado el 16 de octubre de 2013, de Tienda Linux: <http://www2.tiendalinux.com/docs/manuales/redhat/rhl-rg-es-7.1/s1-ldap-procon.html>
- mastermagazine. (14 de junio de 2010). <http://www.mastermagazine.info>. Recuperado el 23 de enero de 2014, de <http://www.mastermagazine.info>: <http://www.mastermagazine.info/termino/4184.php>
- Muñoz, X. (16 de mayo de 2010). Aplicacion Web para el control estadístico de los delitos en Guayaquil. Guayaquil, Guayaquil, Ecuador.
- Pacheco, R. (23 de enero de 2013). *Características de una aplicacion web*. Recuperado el 16 de octubre de 2013, de Características de una aplicacion web: <http://culturacion.com/2010/01/caracteristicas-de-una-aplicacion-web/>
- Practices, B. M. (14 de enero de 2013). *Modelo Vista Controlador*. Recuperado el 16 de octubre de 2013, de Modelo Vista Controlador: <http://es.cyclopaedia.net/wiki/Modelo-Vista-Controlador>
- prpdesigns(BLOG). (5 de julio de 2013). *Sitios Web Autoadministrables*. Recuperado el 16 de octubre de 2013, de Sitios Web Autoadministrables: <http://www.prpdesigns.cl/blog/consejos-sitios-web/ventajas-y-desventajas-de-sitios-web-autoadministrables/>

- Pública, R. E. (1 de octubre de 2013). *http://scielo.isciii.es*. Recuperado el 10 de febrero de 2014, de <http://scielo.isciii.es>: http://scielo.isciii.es/scielo.php?pid=S1135-57272003000500015&script=sci_arttext
- Robert Eckles, R. C. (9 de septiembre de 2010). *Definición de control*. Recuperado el 2014 de enero de 8, de itescoingenieriaenadministracion: <http://itescoingenieriaenadministracion.wikispaces.com>
- ss. (16 de julio de 2013). *Como hacer gráficos estadísticos*. Recuperado el 16 de octubre de 2013, de Como hacer gráficos estadísticos: <http://www.portalscreen.com/2013-01/highchart-libreria-javascript-graficas-estadisticas.html>
- Suez, G. (1 de julio de 2013). *Estructura y composición de un sitio web*. Recuperado el 16 de octubre de 2013, de Estructura y composición de un sitio web: <http://magazine.joomla.org/es/ediciones-anteriores/julio-2013/item/1398-estructura-y-composicion-de-un-sitio-web>
- SUPERIOR, M. D. (s.f.).
- SUPERIOR, M. D. (5 de octubre de 2010). *Definición Y Tipos De Aplicaciones Web*. Recuperado el 8 de enero de 2013, de Definición Y Tipos De Aplicaciones Web.: <http://www.buenastareas.com/ensayos/Definici%C3%B3n-y-Tipos-De-Aplicaciones-Web/317130.html>
- SUPERIOR, M. D. (5 de Mayo de 2010). *Definición Y Tipos De Aplicaciones Web*. Obtenido de Definición Y Tipos De Aplicaciones Web.
- SUPERIOR, M. D. (5 de mayo de 2010). *Definición Y Tipos De Aplicaciones Web*. Recuperado el 16 de octubre de 2013, de Definición Y Tipos De Aplicaciones Web.: <http://www.buenastareas.com/ensayos/Definici%C3%B3n-y-Tipos-De-Aplicaciones-Web/317130.html>
- Systems, Q. (14 de mayo de 2013). *LDAP*. Recuperado el 16 de octubre de 2013, de LDAP: <http://www.qnap.com/mx/index.php?lang=es-mx&sn=6380>

- TEMIQUEL, A. (1 de noviembre de 2013). *¿Qué es FPDF?* Recuperado el 16 de octubre de 2013, de *¿Qué es FPDF?*: <http://www.alfredotemiquel.com.mx/que-es-fpdf/>
- Terry, G. R. (9 de septiembre de 2010). *definicion de control*. Recuperado el 8 de enero de 2014, de <http://itescoingenieriaenadministracion.wikispaces.com>: <http://itescoingenieriaenadministracion.wikispaces.com>
- Torre, A. d. (3 de octubre de 2012). <http://adelat.org>. Recuperado el 22 de enero de 2014, de <http://adelat.org>: http://adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html
- UML, D. (22 de mayo de 2010). <http://egdamar877.blogspot.com>. Recuperado el 28 de enero de 2014, de <http://egdamar877.blogspot.com>: <http://egdamar877.blogspot.com/2009/05/expocicion.html>
- Valda, J. C. (10 de diciembre de 2012). *definicion.de*. Recuperado el 21 de mayo de 2014, de <http://definicion.de/gestion/>: <http://jcvalda.wordpress.com/2012/12/10/definicion-y-caracteristicas-de-los-indicadores-de-gestion-empresarial/>
- Valladolid, U. d. (15 de febrero de 2010). <http://www.infor.uva.es>. Recuperado el 15 de enero de 2014, de <http://www.infor.uva.es>: http://www.infor.uva.es/~fdiaz/sd/2005_06/doc/SD_TE02_20060305.pdf
- VELASCO GIRALDO, D. F., & SAA VIAFARA, A. (3 de JUNIO de 2012). DISEÑO DE UNA PLATAFORMA CRM INTEGRADA CON ASTERISK. SANTIAGO DE CALI, CALI, ESPAÑA.
- Velasco, J. e. (10 de septiembre de 2010). <http://www.prograweb.com.mx>. Recuperado el 22 de enero de 2014, de <http://www.prograweb.com.mx>: <http://www.prograweb.com.mx/pweb/0202ladoCliente.html>
- Vera, H. S. (10 de septiembre de 2010). *Aplicaciones Web*. Recuperado el 16 de octubre de 2013, de *Aplicaciones Web*: <http://www.e-lemental.com.pe/que-es-una-aplicacion-web/>
- Vergara, G. (31 de marzo de 2011). <http://mejoratugestion.com>. Recuperado el 21 de mayo de 2014, de <http://mejoratugestion.com>:

<http://mejoratugestion.com/mejora-tu-gestion/que-es-un-sistema-de-gestion/>

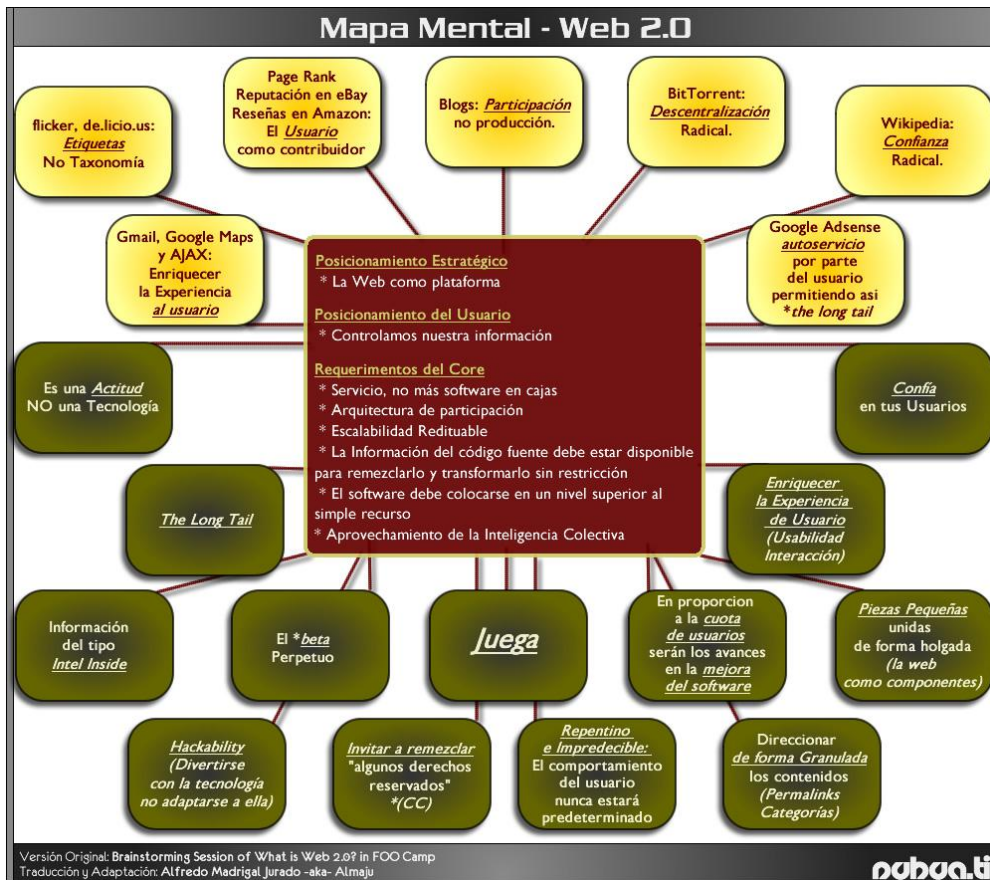
WordPress. (16 de Septiembre de 2013). *Como hacer gráficos estadísticos*. Recuperado el 16 de Octubre de 2013, de Como hacer gráficos estadísticos: <http://www.portalscreen.com/2013-01/highchart-libreria-javascript-graficas-estadisticas.html>

WorldPress. (27 de septiembre de 2010). <http://ingti.wordpress.com/>. Recuperado el 9 de enero de 2014, de <http://ingti.wordpress.com/>: <http://ingti.wordpress.com/2011/09/27/definicion-de-proceso-de-negocio/>

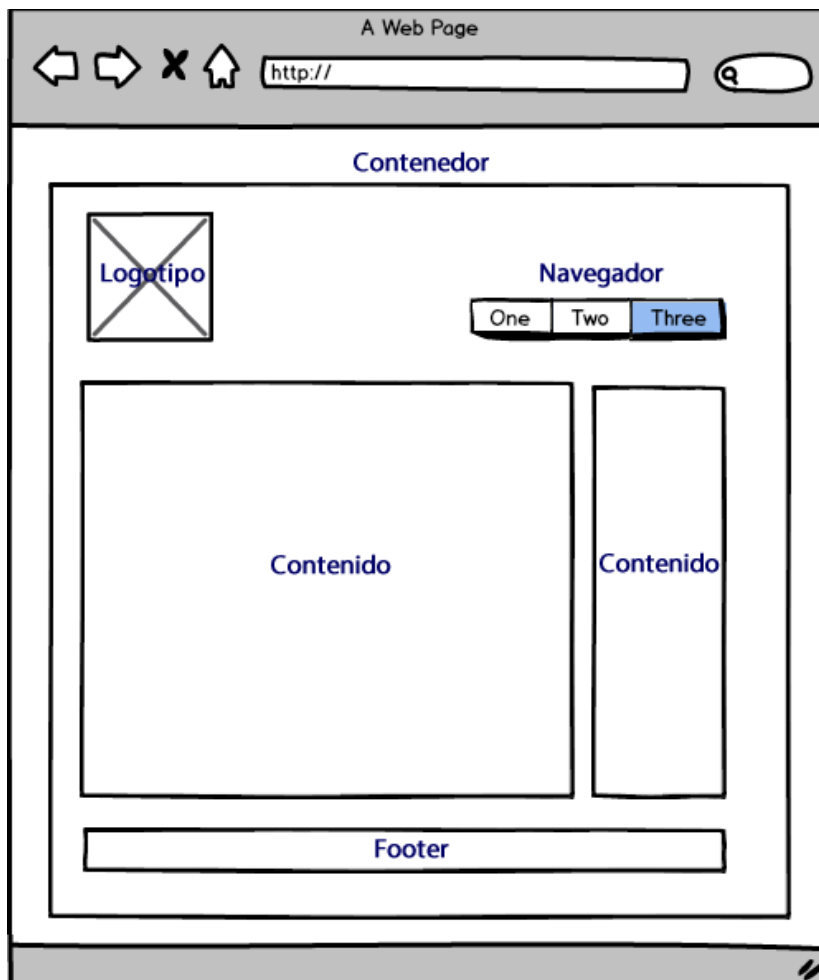
WorldPress2. (s.f.).

Anexos

1- Aplicación web. (Celaya, 2006)

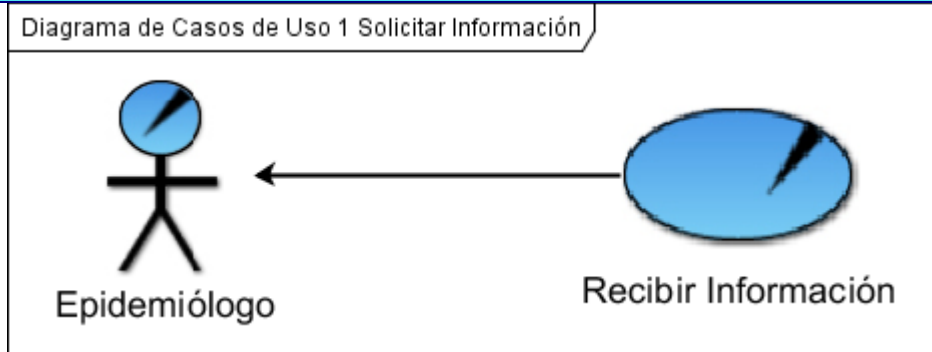


2- Sitio web. (Suez, 2013).

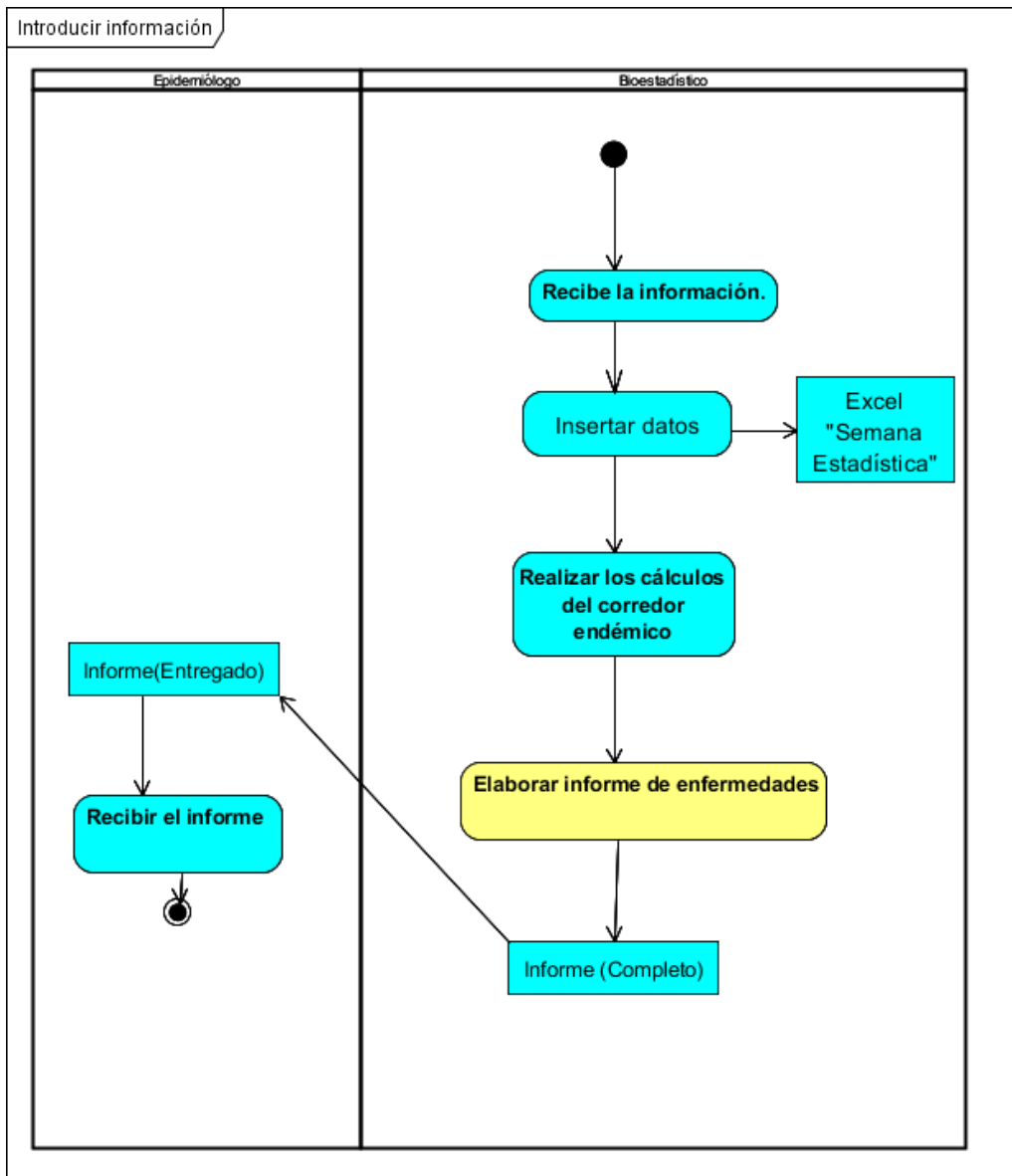


3- **Figura 1 Diagrama de Casos de Uso del Negocio Solicitar Información.**

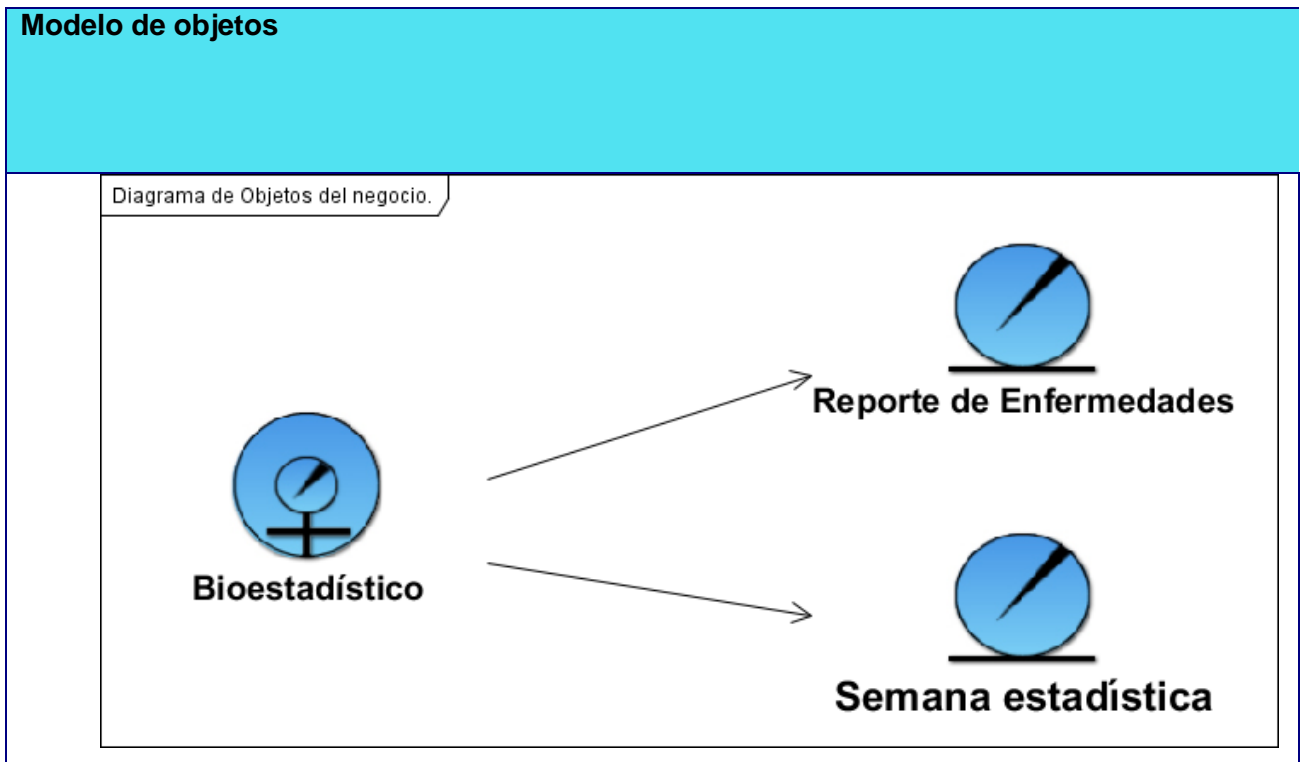
Diagrama de Casos de Uso del Negocio



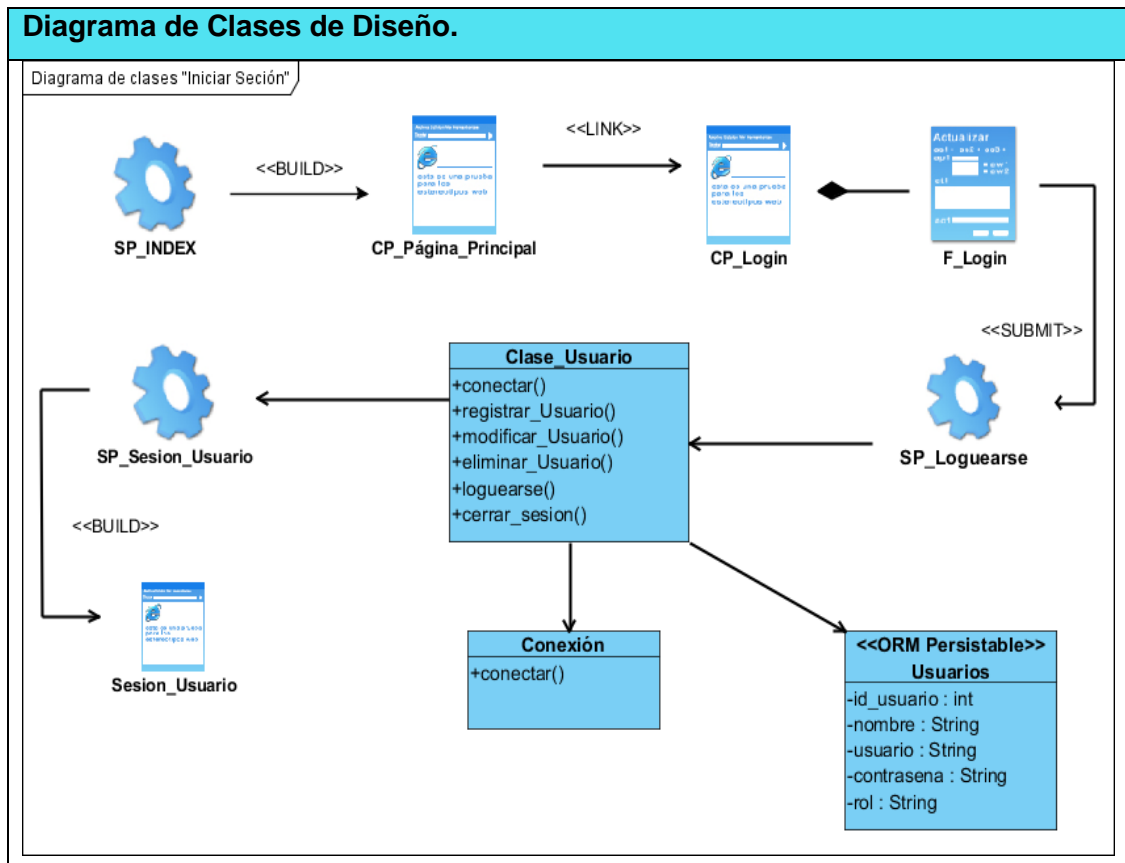
4- **Figura 2 Diagrama de Actividades.**



5- Figura 3 Diagrama Modelo de objetos.

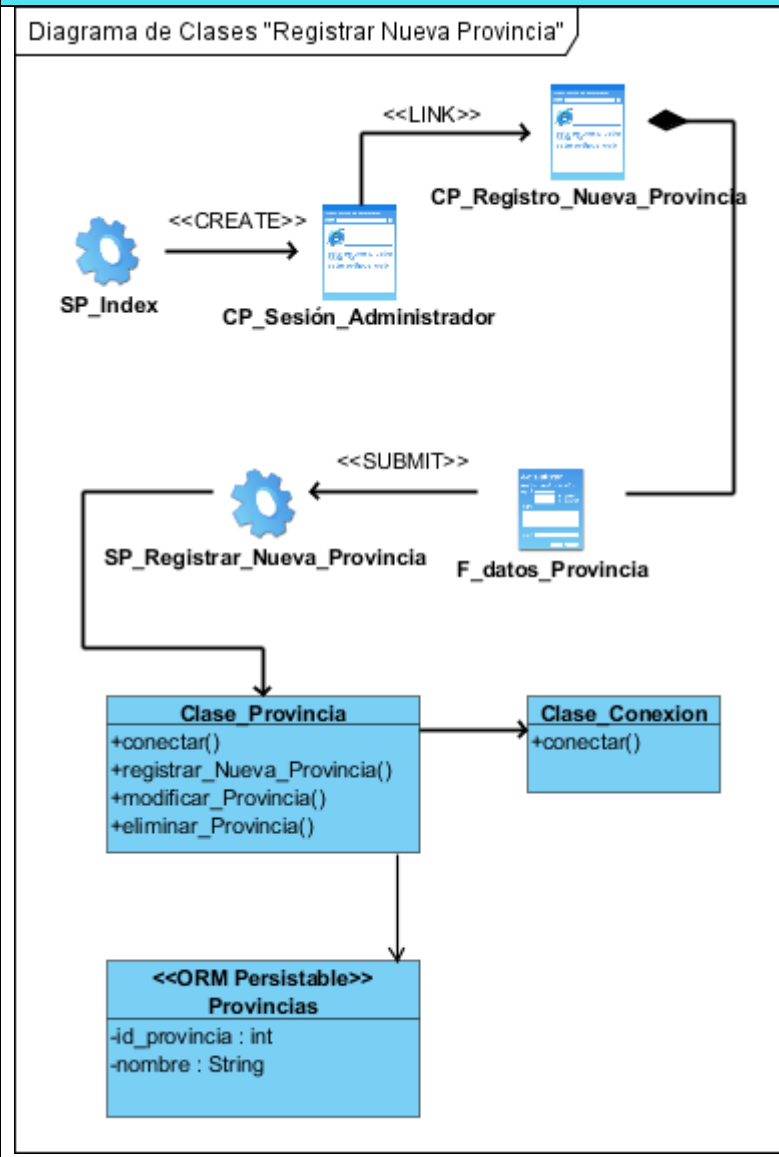


6- Diagrama de Clases. Iniciar Sesión.



7- Diagrama de Clases. Agregar Provincia

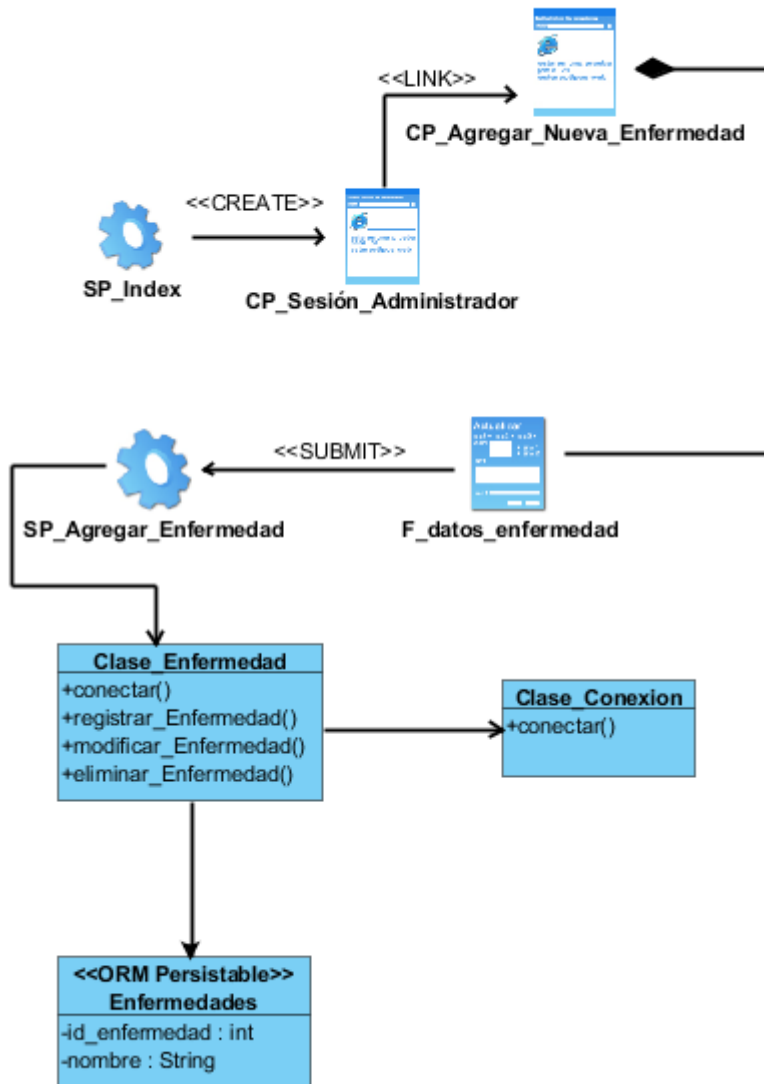
Diagrama de Clases de Diseño.



8- Diagrama de Clases. Agregar enfermedad

Diagrama de Clases de Diseño.

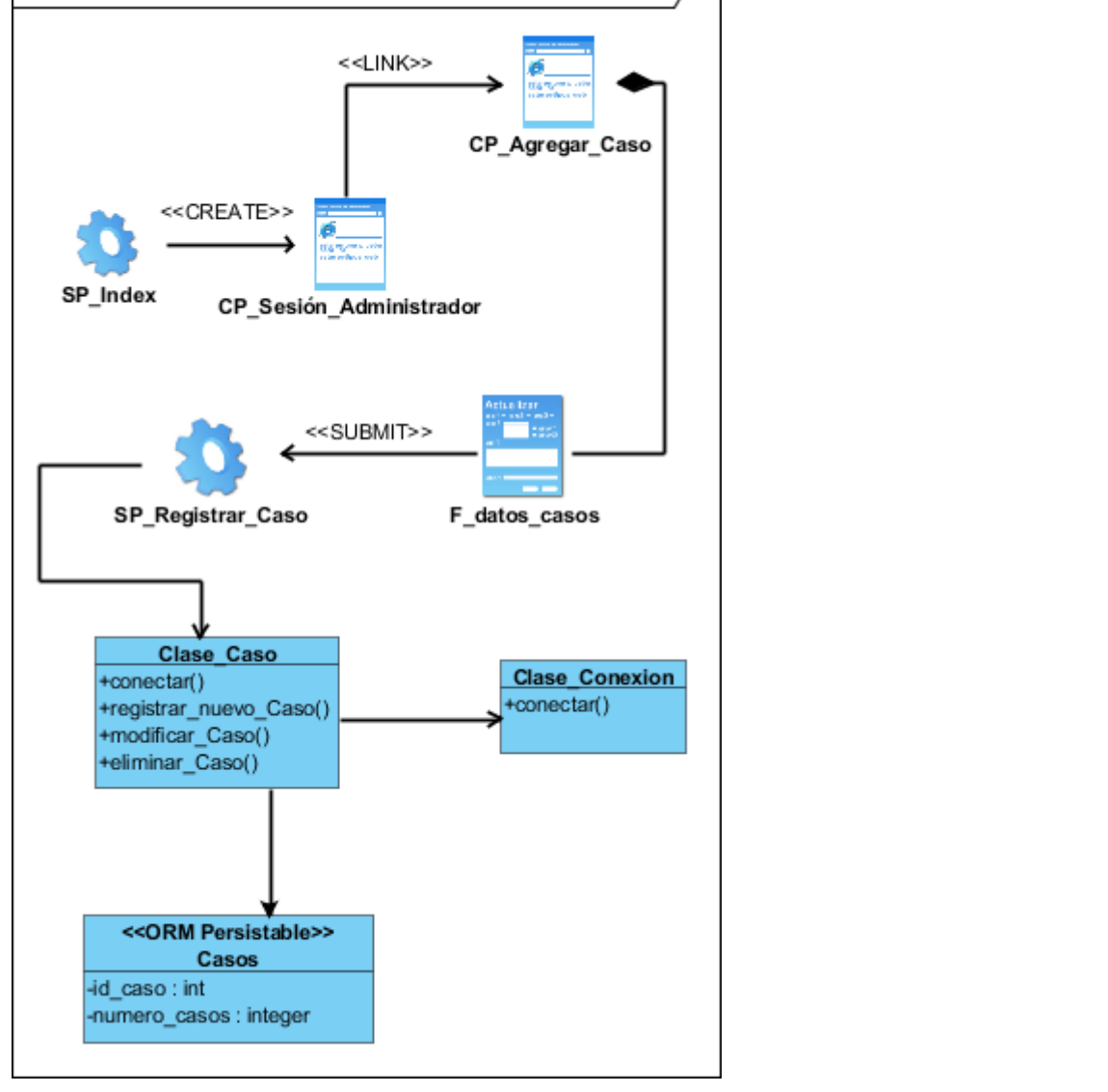
Diagrama de Clases "Agregar Nueva Enfermedad"



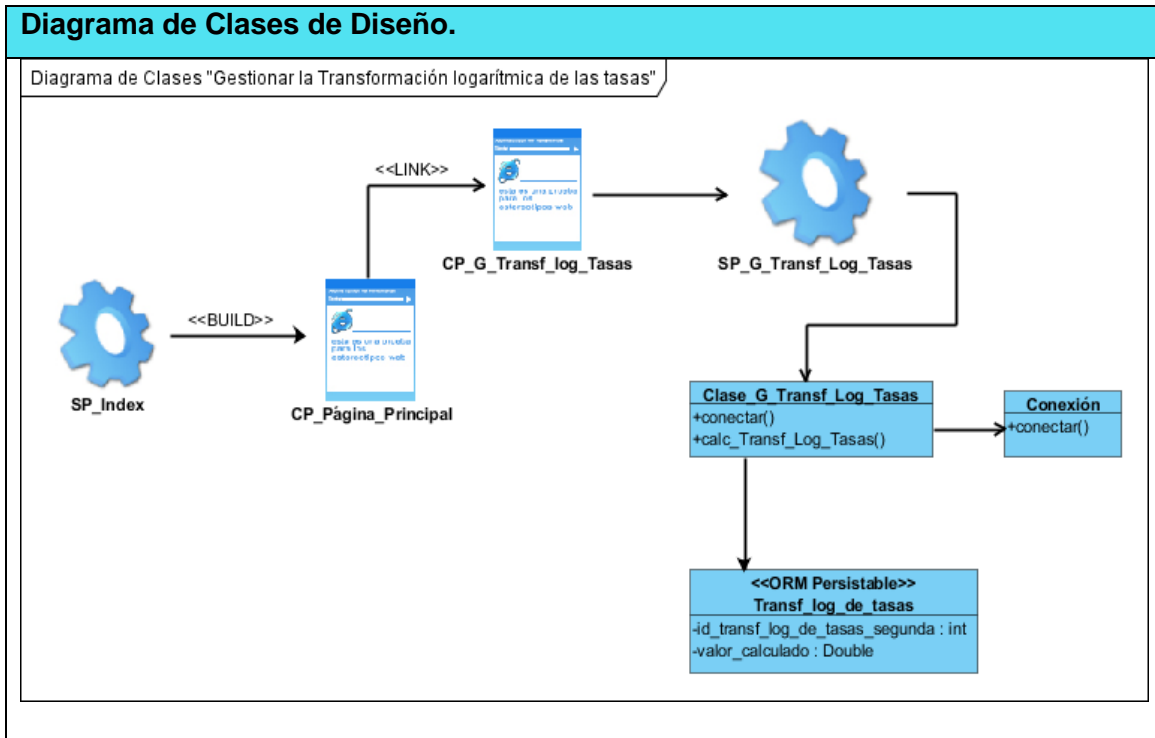
9- Diagrama de Clases. Agregar nuevo caso por enfermedad.

Diagrama de Clases de Diseño.

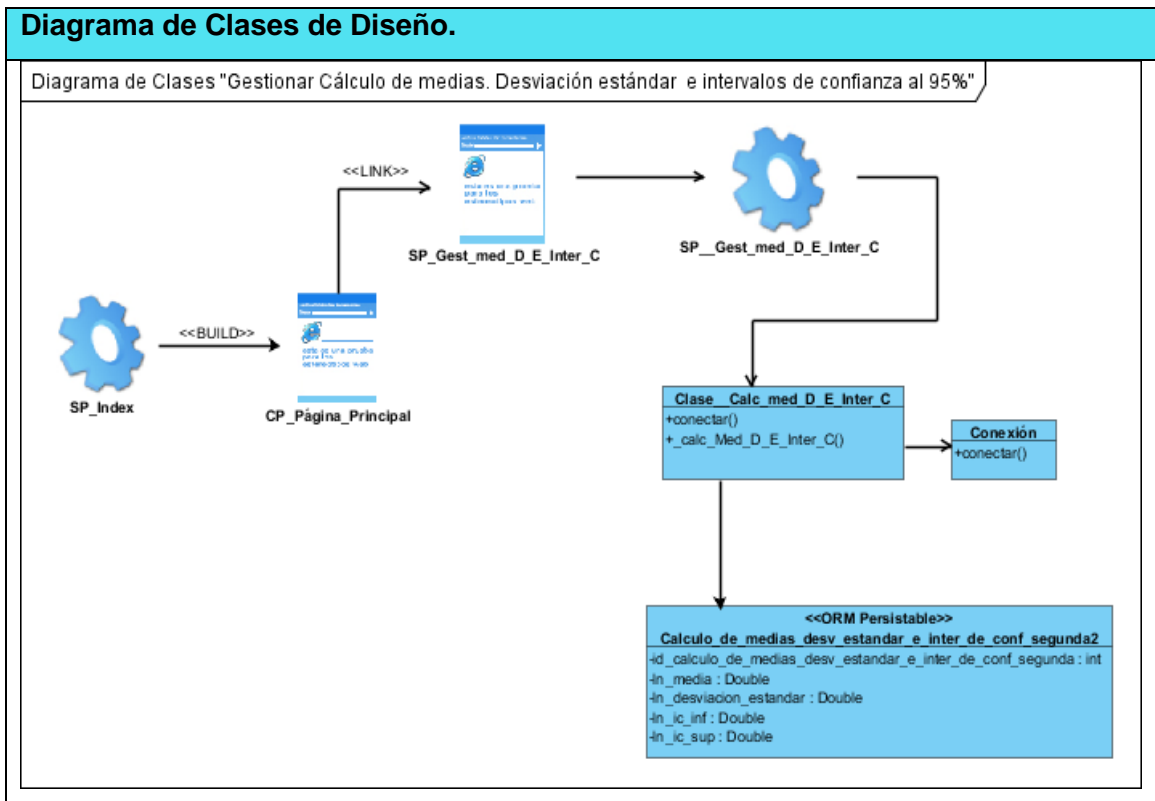
Diagrama de Clases "Agregar Nuevo caso por Enfermedad"1



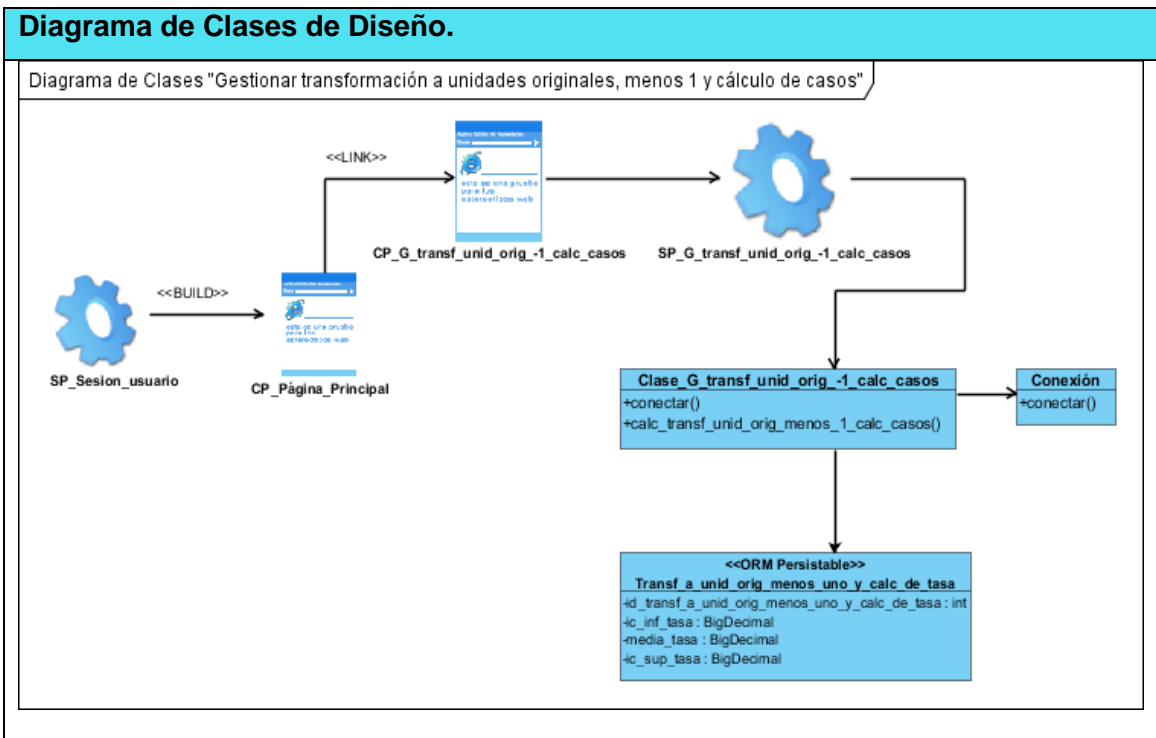
10- Diagrama de Clases. Gestionar Transformación Logarítmica de las tasas.



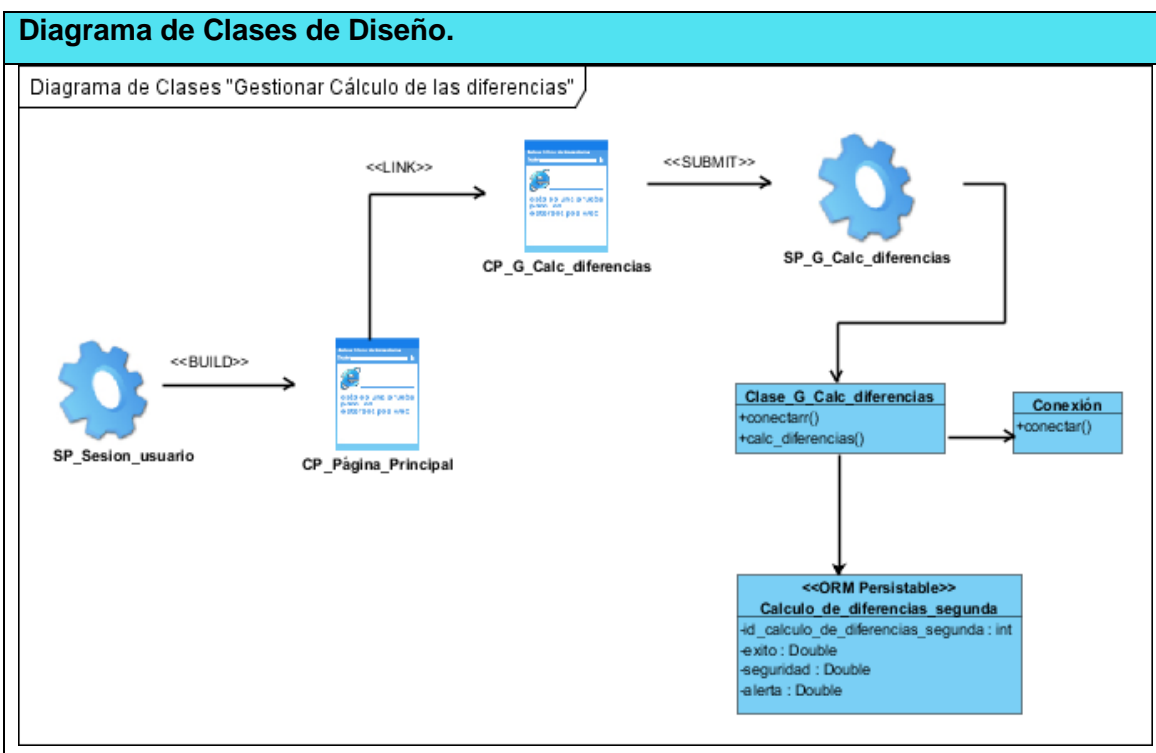
11- Diagrama de Clases. Gestionar cálculo de medias, Desviación estándar y cálculo de intervalos de confianza.



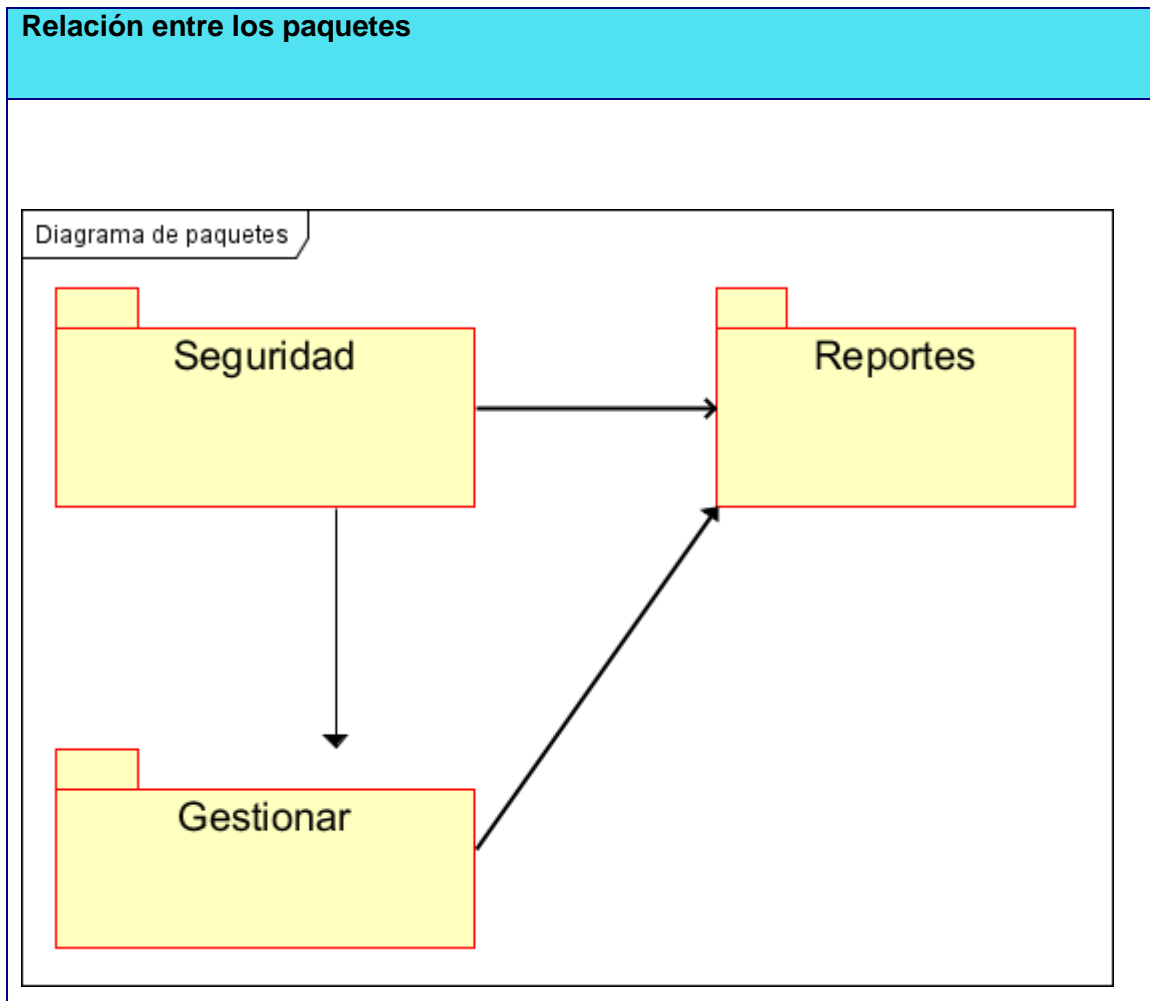
12- Diagrama de Clases. Transformación a unidades originales.



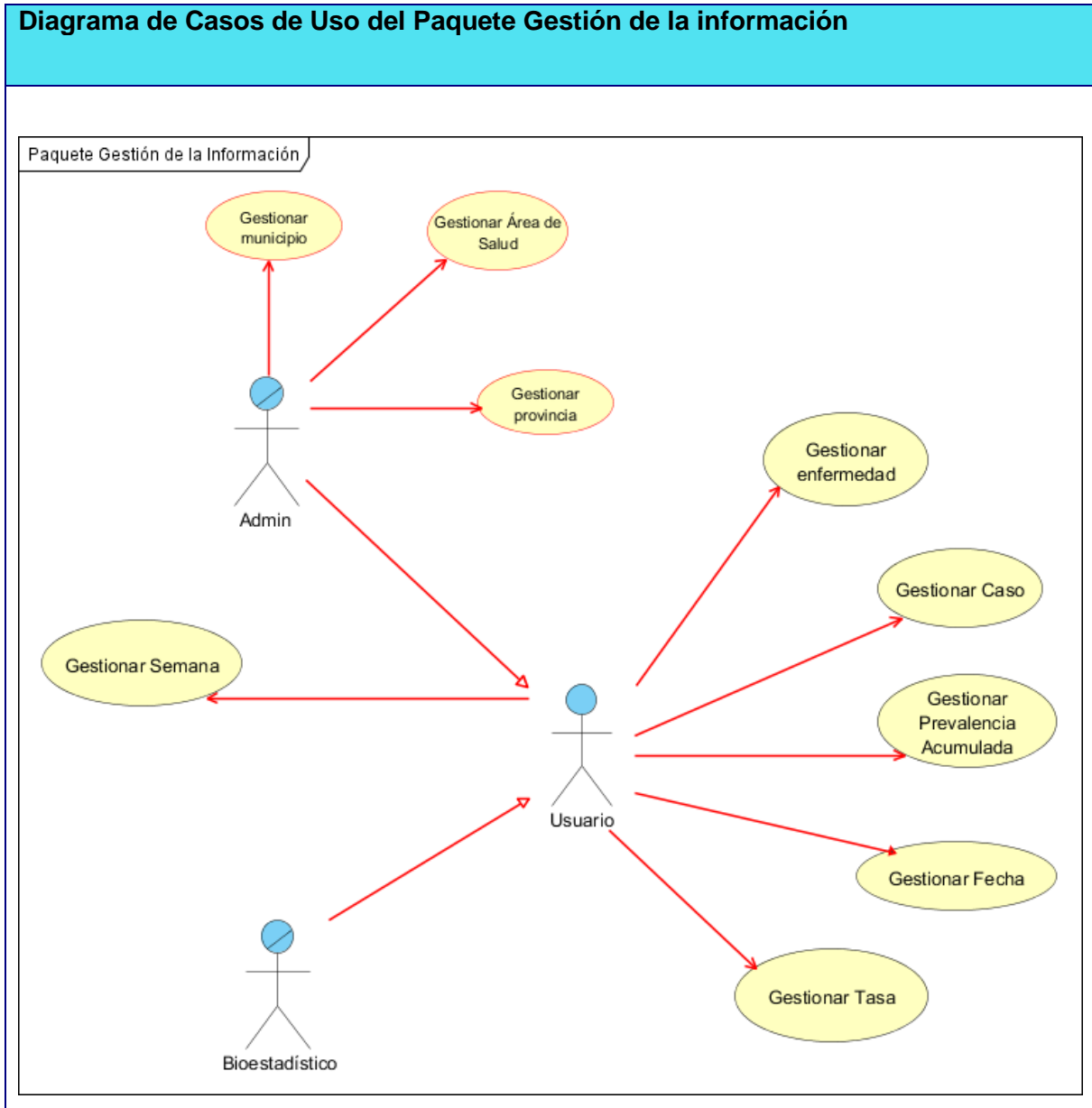
13- Diagrama de Clases. Cálculo de las diferencias.



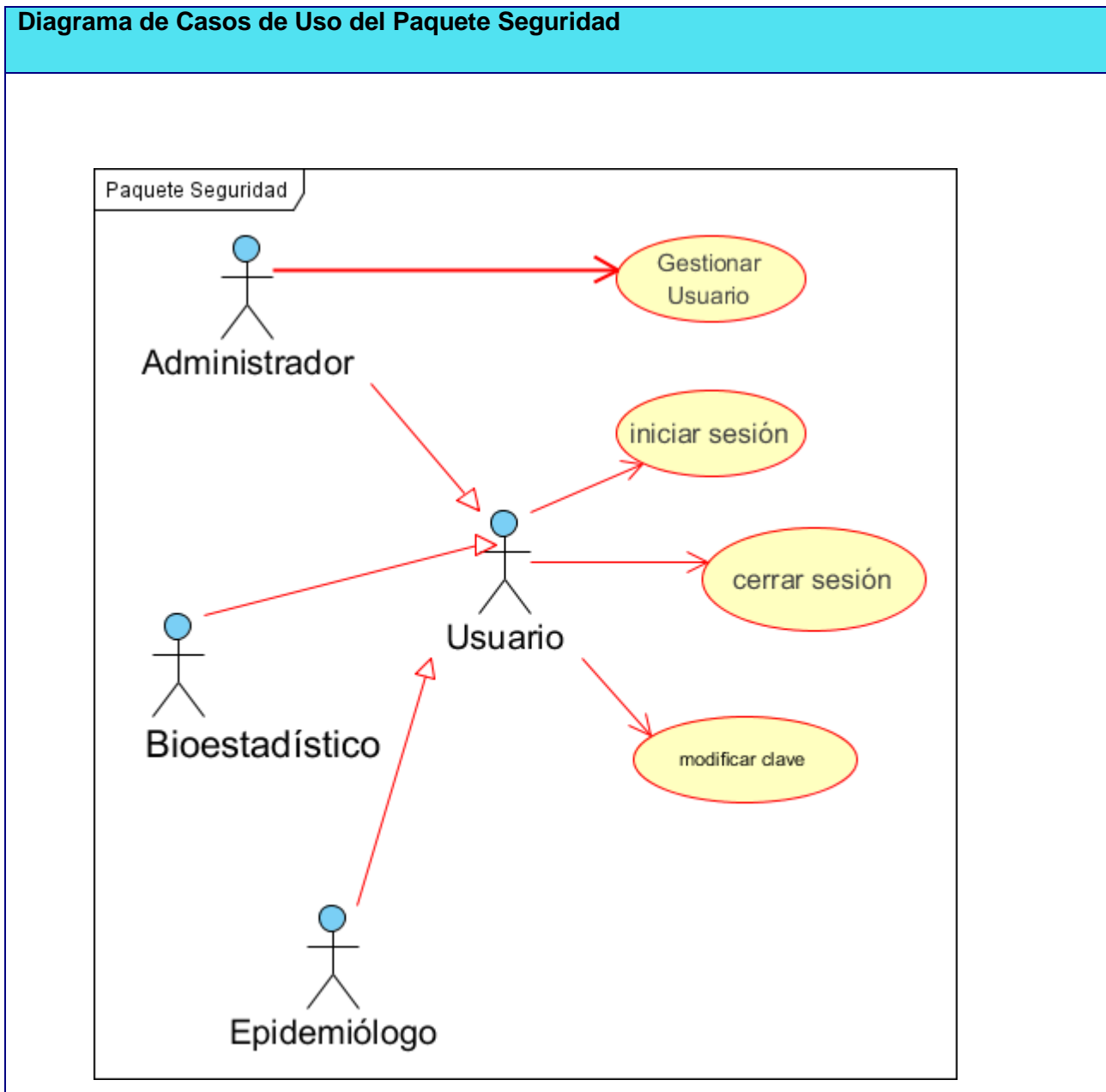
14- Figura 4 Relación entre los paquetes del sistema.



15- Figura 5 Diagrama de Casos de Uso: Paquete Gestión de la información.

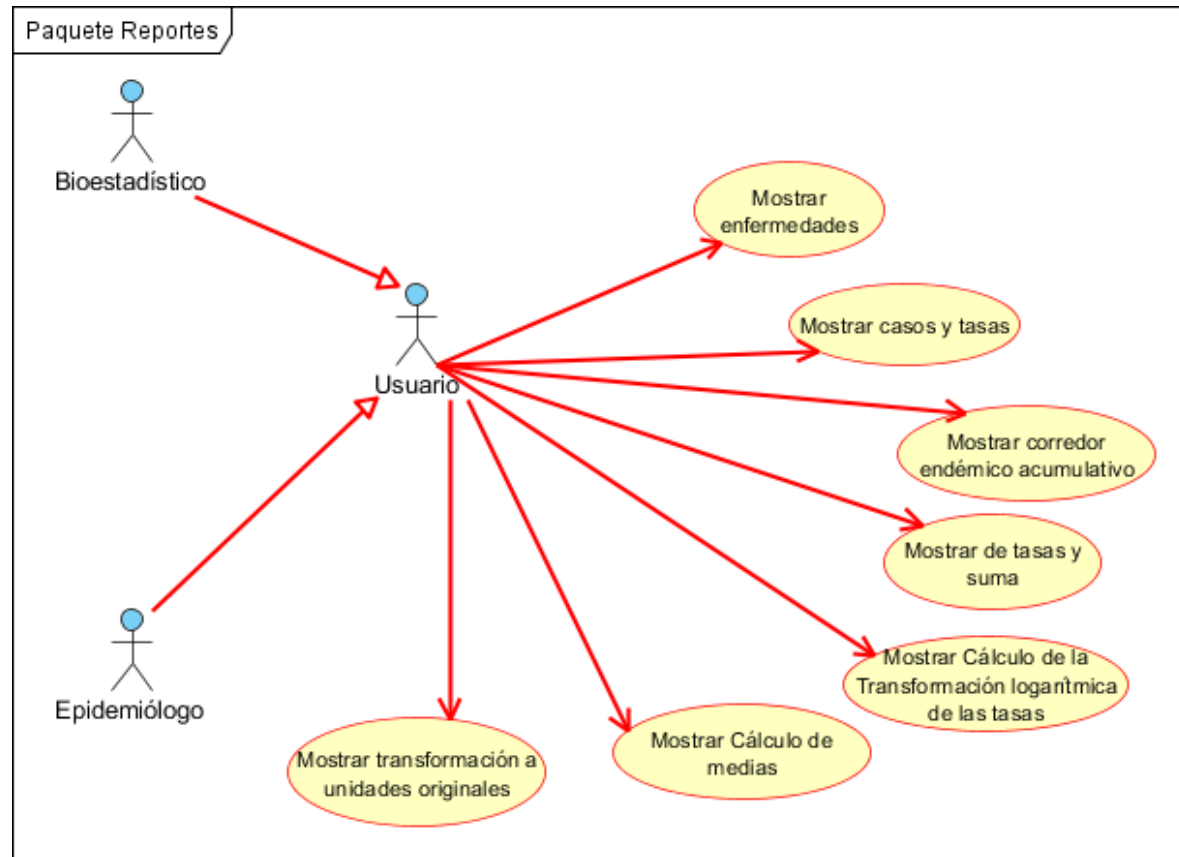


16- Figura 6 Diagrama de Casos de Uso: Paquete Seguridad.



17- Figura 7 Diagrama de Caso de Uso: Paquete Reportes.

Diagrama de Casos de Uso del Paquete Reportes



18-Vistas de la aplicación

INICIO | **IDENTIDAD** | **MISIÓN Y VISIÓN** | **¿QUÉ HACEMOS?** | **CONTÁCTENOS**

ADMINISTRACIÓN | **REGISTROS** | **CÁLCULOS** | **REPORTES** | **GRÁFICOS** | **GESTIÓN** | **DIRECCIÓN**

GESTION DE INSTITUCIONES

Bienvenido, omi [Cambiar contraseña](#) | [Perfil](#) | [Salir](#)

FUNCIONAMIENTO

[Estrategia de gestión](#)
[Sesiones científicas](#)
[Productos informáticos](#)
[Bibliografías](#)

NOTICIAS

ni idea

NÚMERO DE VISITANTES

Hoy	1
Esta semana	27
Este mes	62
Hasta hoy	139

Inicio

Buscar | Agregar | Editar | Borrar

EVENTOS

de andad

ENLACES DE INTERÉS

X	Nombre	Descripción
	dsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf
	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf
	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf
	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf
	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf
	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf
	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf
	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf
	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf
	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf
	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf
	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf	dsfsdfsdfsafsdfs dsfsdf sdf dsfsdfsdfsafsdfs dsfsdf sdf

eee.cu
cuba.cu

19- Vistas de la aplicación

INICIO | **IDENTIDAD** | **MISIÓN Y VISIÓN** | **¿QUÉ HACEMOS?** | **CONTÁCTENOS**

ADMINISTRACIÓN | **REGISTROS** | **CÁLCULOS** | **REPORTES** | **GRÁFICOS** | **GESTIÓN** | **DIRECCIÓN**

PROVINCIAS
MUNICIPIOS
ENTIDADES
USUARIOS
GESTIONAR ENFERMEDADES
GESTIONAR FECHA
GESTIONAR SEMANAS
GESTIONAR CASOS

Bienvenido, omi [Cambiar contraseña](#) | [Perfil](#) | [Salir](#)

Enfermedades

Buscar | Agregar | Editar | Borrar

Agregar Enfermedad

Nombre:
Entidad:

Aceptar | Cancelar

NÚMERO DE VISITANTES

Hoy	1
Esta semana	27
Este mes	62
Hasta hoy	139

EVENTOS

de andad

ENLACES DE INTERÉS

eee.cu
cuba.cu

localhost/higiene_sw/?ver=enfermedades

20- Vistas de la aplicación

The screenshot displays a web application interface with a dark navigation bar at the top containing links: INICIO, IDENTIDAD, MISIÓN Y VISIÓN, ¿QUÉ HACEMOS?, and CONTÁCTENOS. Below this is a secondary navigation bar with: ADMINISTRACIÓN, REGISTROS, CÁLCULOS, REPORTES, GRÁFICOS, GESTIÓN, and DIRECCIÓN. A left sidebar lists menu items: PROVINCIAS, MUNICIPIOS, ENTIDADES, USUARIOS, GESTIONAR ENFERMEDADES, GESTIONAR FECHA, GESTIONAR SEMANAS, and GESTIONAR CASOS. A 'NÚMERO DE VISITANTES' widget shows: Hoy (1), Esta semana (27), Este mes (62), and Hasta hoy (139). The main content area features a modal window titled 'Casos de Incidencias por Enfermedades' with buttons for 'Buscar', 'Agregar', 'Editar', and 'Borrar'. It contains a table with the following data:

X	Cantidad de Casos	Semana	Año Ocurrencia	Enfermedad
<input type="checkbox"/>	9	42	2010	catarro
<input type="checkbox"/>	70	1	2013	catarro
<input type="checkbox"/>	34	1	1990	catarro
<input type="checkbox"/>	657	1	2013	catarro
<input type="checkbox"/>	3	1	1990	catarro
<input type="checkbox"/>	20	2	2013	fiebre
<input type="checkbox"/>	30	1	2013	fiebre

Additional interface elements include a 'Bienvenido, omi' message with links for 'Cambiar contraseña', 'Perfil', and 'Salir', and a right sidebar with 'EVENTOS' and 'ENLACES DE INTERÉS' sections.

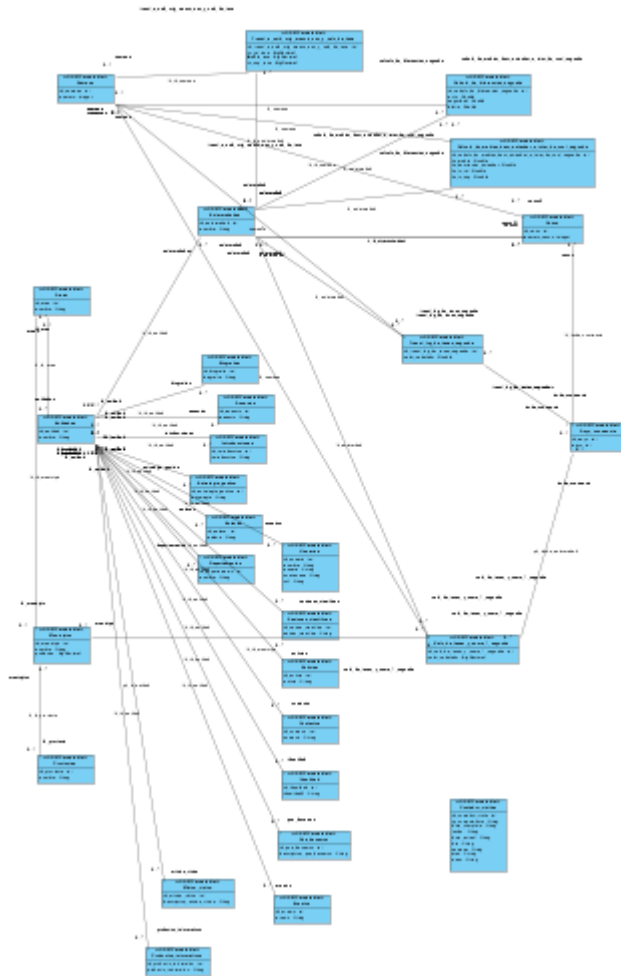
21- Vistas de la aplicación

The screenshot shows a web application interface with a dark navigation bar containing: INICIO, IDENTIDAD, MISIÓN Y VISIÓN, ¿QUÉ HACEMOS?, and CONTÁCTENOS. Below it is a navigation bar with: ADMINISTRACIÓN, REGISTROS, CÁLCULOS, REPORTES, GRÁFICOS, GESTIÓN, and DIRECCIÓN. A left sidebar includes 'FUNCIONAMIENTO' with links for 'Estrategia de gestión', 'Sesiones científicas', 'Productos informáticos', and 'Bibliografías', and a 'NÚMERO DE VISITANTES' widget showing: Hoy (2), Esta semana (28), Este mes (63), and Hasta hoy (140). The main content area features a modal window titled 'Agregar Nuevos Casos' with the following form fields:

- Cantidad de Casos: 4
- Semana: 1
- Fecha Ocurrencia: 2014
- Enfermedad: L.presuntiva

Buttons for 'Aceptar' and 'Cancelar' are located at the bottom of the form. The right sidebar contains 'EVENTOS' and 'ENLACES DE INTERÉS' sections.

22- Modelo de Clases Persistentes.



23- Diagrama de Modelo de Datos.

