



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**  
**FACULTAD 7**

Trabajo de Diploma para Optar por el Título de  
Ingeniero en Ciencias Informáticas

**Título:** Componente web Enfermedades Transmitidas por Alimentos del Sistema  
Integral para la Atención Primaria de Salud

**Autor:** Katia Herrera Quintana

**Tutora:** Ing. Ariuska Sablón Hernández

**La Habana, junio del 2011**

**“Año 53 de la Revolución”**

### Datos de Contacto

**Nombre: Ing. Ariuska Hernández Sablón:** Graduada como Ingeniera en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI), en el curso (2007-2008). Actualmente, posee la categoría docente instructor recién graduado y se desempeña como profesora de la asignatura de Matemática. Se encuentra vinculada al desarrollo del proyecto productivo Atención Primaria de Salud (APS), solución informática que automatiza y gestiona los procesos esenciales de la Atención Primaria de la Salud en Cuba, en el mismo se destaca como Analista.

Correo electrónico: [asablon@uci.cu](mailto:asablon@uci.cu)

## Resumen

La presente investigación propone obtener un sistema informático que facilite la gestión de la información de las Enfermedades Transmitidas por Alimentos en la Atención Primaria de Salud, basado en tecnologías libres, multiplataforma y de código abierto.

Para el desarrollo de este componente se emplea el lenguaje de programación Java, para el trabajo con este lenguaje el Eclipse Galileo v3.5.2. Como Framework integrador se utilizó JBoss Seam v2.1.1. Como Sistema Gestor de Base de Datos se utilizó PostgreSQL v8.4, para la administración de la base de datos PgAdmin III v1.10.5. Se utilizó la herramienta Visual Paradigm v6.4 para el modelado de los artefactos haciendo uso de la notación UML v2.0.

Con el Componente Enfermedades Transmitidas por Alimentos del Subsistema Web del Sistema Integral para la Atención Primaria de Salud se espera mejorar dicho proceso, lo que posibilita que la información que este genera, sea actualizada y accesible en tiempo real. Además permite analizar e interpretar los datos para determinar en número la distribución y severidad de los casos, conocer los alimentos responsables de la transmisión de los agentes etiológicos, determinar los grupos de población más expuestos a riesgos y la identificación de los factores contribuyentes a la transmisión.

**Palabras clave:** Atención Primaria de Salud, Brote epidémico, Enfermedades Transmitidas por Alimentos, Intoxicación, Sistema informático.

## Tabla de Contenidos

<b>Introducción</b> .....	1
<b>Capítulo 1. Fundamentación Teórica</b> .....	6
1.1 Sistema Nacional de Salud .....	6
1.2 Marco conceptual.....	7
1.3 Análisis de soluciones existentes .....	8
1.4 Problema a Resolver.....	10
1.5 Tecnologías, herramientas y metodologías.....	11
<b>Capítulo 2. Características del Sistema</b> .....	20
2.1 Modelo de Negocio .....	20
2.2 Diagrama de proceso del negocio.....	20
2.3 Flujo actual del proceso involucrado. ....	21
2.4 Propuesta del sistema.....	25
<b>Capítulo 3. Diseño del Sistema</b> .....	31
3.1 Modelos de diseño.....	31
3.2 Patrones de diseño.....	31
3.3 Definición de Elementos de Diseño.....	33
<b>Capítulo 4. Implementación</b> .....	55
4.1 Propuesta de integración entre módulos .....	55
4.2 Propuesta de seguridad del módulo.....	55
4.3 Implementación.....	56
4.4 Diagrama de Despliegue.....	56
4.5 Estándares de codificación y tratamiento de excepciones .....	57
4.6 Tratamiento de excepciones. ....	58
<b>Conclusiones</b> .....	60
<b>Recomendaciones</b> .....	61
<b>Referencias Bibliográficas</b> .....	62
<b>Bibliografía</b> .....	66
<b>Anexos</b> .....	71

## Introducción

Las Tecnologías de la Información y la Comunicación (TIC) representan la revolución tecnológica más poderosa de estos tiempos. Sus repercusiones y la rapidez con que se producen están introduciendo cambios fundamentales que definirán los retos y las oportunidades en el futuro. Las TIC podrían convertirse en un instrumento eficaz para fomentar la igualdad en el terreno del crecimiento y el desarrollo económico, así como para facilitar el acceso a los conocimientos en todos los niveles de la sociedad. Permite además, compartir recursos de información con alta especialización, para que sean utilizados cada vez más, con elevada efectividad y eficiencia por aquellos que los necesitan. [1]

Con el objetivo de lograr un desarrollo económico y social desde una perspectiva de equidad y participación, donde la salud y la educación sean pilares esenciales, se crea en Cuba el Programa de Informatización de la Sociedad Cubana. Este programa persigue promover el uso masivo de las tecnologías a escala nacional. Este pretende impulsar de manera coherente a todos los sectores, con una identificación precisa de los actores de la Sociedad de la Información.

Por esta razón, desde 1992, el Ministerio de Salud Pública (por sus siglas MINSAP) ha trabajado en la introducción de las redes telemáticas como una estrategia para ponerlas al servicio de la preparación de los recursos humanos y brindar acceso a la información actualizada y pertinente dentro del Sistema Nacional de Salud (por sus siglas SNS). Este sistema tiene como propósito garantizar la calidad en la prevención, cuidado, rehabilitación de los pacientes y especializar los servicios de salud, para ello se organiza en tres niveles de atención médica: Atención Primaria, Atención Secundaria y Atención Terciaria.

La **Atención Primaria** abarca alrededor del 80% de los problemas de salud. Constituye, sin dudas, el eslabón fundamental del SNS del país. A partir de su organización se estructura todo el sistema de atención integral al hombre y a su medio. Cuenta con recursos y con un personal especializado que puede capacitar a médicos y enfermeras de la familia para lograr una participación activa de la comunidad en la identificación de los problemas de salud, en la toma de decisiones y en la ejecución de todas aquellas medidas que den solución a los problemas priorizados. Comprende actividades de promoción de la salud, educación sanitaria, prevención de la enfermedad, mantenimiento y recuperación de la salud, así como la rehabilitación física y el trabajo social. [2]

**Atención Secundaria:** Este nivel da cobertura acerca del 15 % de los problemas de salud, está orientada al diagnóstico y tratamiento de pacientes derivados de la Atención Primaria, abarca la atención de especialidades médica brindada en los diferentes tipos de instituciones hospitalarias entre ellas Hospitales Municipales, Generales y Especiales. [3]

**Atención Terciaria:** El nivel terciario abarca alrededor del 5 % de los problemas de salud, conocida habitualmente como Atención Hospitalaria, otorga prestaciones de mayor complejidad, tanto en atención cerrada como ambulatoria. A este nivel pertenecen los Institutos y Hospitales especializados. [4]

En Cuba el desarrollo actual de la APS, requiere de nuevas dimensiones de trabajo en la medicina familiar, que permitan con un enfoque más amplio, la integración de las visiones clínicas, epidemiológicas y sociales como herramientas fundamentales de trabajo, y constituyan estas el objetivo fundamental de la integración de la higiene y la epidemiología a la APS, apoyados en el proceso de descentralización del país.

La estrategia de utilización de la higiene y la epidemiología en la APS requiere de la estructuración de los sistemas de vigilancia que contribuyan al conocimiento de las incidencias y su análisis, que establezca las tendencias y el pronóstico del análisis de la situación de salud, en tanto, ofrezca la posibilidad de tomar decisiones prácticas de promoción de salud, prevención y control de las enfermedades.

El cumplimiento de las funciones de los higienistas y epidemiólogos desde Unidades Municipales de Higiene y Epidemiología (por sus siglas en español UMHE), con una absoluta verticalidad en sus acciones, contribuye a la obtención de resultados satisfactorios en los programas cubanos de salud pública.

Dentro de los programas que se desarrollan en las UMHE está el relacionado al estudio de las Enfermedades Transmitidas por Alimentos (por sus siglas ETA). El mecanismo de vigilancia de este, está integrado al sistema de vigilancia en salud pública e implica un trabajo de colaboración entre epidemiólogos, trabajadores sanitarios, médicos clínicos, responsables de Programas de Alimentos de los laboratorios y personal de salud en general.

Las ETA pueden generarse a partir de un alimento y/o agua contaminada. Son llamadas así porque el alimento actúa como vehículo de transmisión de organismos dañinos y sustancias tóxicas. Un brote de una ETA surge cuando dos o más personas sufren una enfermedad similar después de ingerir un mismo

alimento y los análisis epidemiológicos señalan al alimento como el origen de la enfermedad, que luego es confirmado por el laboratorio. [5]

La ocurrencia de brotes sugiere que otras enfermedades pueden ser debidas al consumo de alimentos contaminados pero no reconocidos como tales. Así el sistema de vigilancia epidemiológica y la investigación de brotes son herramientas poderosas para el entendimiento de la patogénesis de la enfermedad, detectar nuevos agentes y aprender cómo mejorar la conservación de los alimentos para evitar su contaminación desde su fuente. [6]

En todas las UMHE los casos de intoxicación alimentaria son registrados, lo que permite analizar las causas del brote y controlarlas para evitar la multiplicación de las enfermedades que surgen producto de la mala manipulación y elaboración de los alimentos.

En la actualidad el cúmulo de información sobre los brotes de ETA ha crecido considerablemente, aparecen nuevos agentes contaminantes y con ellos nuevas medidas a tomar con el objetivo de evitar la ocurrencia de brotes que pueda afectar a la población. Por tal motivo se realizan inspecciones y encuestas por parte de los técnicos de higiene y epidemiología encargados de este programa.

Todas estas actividades de control y prevención así como los análisis realizados a las muestras de alimentos son registrados en documentos o plantillas (Diligencia de Inspección Sanitaria, Encuesta ETA, Diligencia de Ocupación de la Muestra Resultado de Análisis) que con el paso del tiempo se deterioran, trayendo esto como consecuencia la pérdida de la información.

Debido al volumen de información que se gestiona de forma manual, por vía telefónica y en algunos casos a través del correo electrónico por parte de los especialistas, que registran la misma en documentos Word o tablas Excel, se dificulta el intercambio de datos, lo que obstaculiza la búsqueda de variables cuando se necesite consultar o confeccionar reportes e informes tales como: Anexo1, Anexo 3, Informe de muestra al INHA y Enfermedades de Declaración Obligatoria Notificación de Casos; los cuales son solicitados por instancias superiores o por el mismo director de Higiene y Epidemiología.

Después de analizar la situación antes expuesta, se plantea como **problema a resolver**: ¿Cómo facilitar el proceso de gestión de la información de las ETA en la Atención Primaria de Salud?

Se definió como **objeto de estudio** la gestión de la información de las ETA. Se concibió como **objetivo general**: desarrollar una aplicación web que gestione la información referente a las ETA en los

departamentos de Higiene y Epidemiología. Enmarcado en el **campo de acción**: El proceso de gestión de la información de las ETA en los departamentos de Higiene y Epidemiología pertenecientes a la Atención Primaria de Salud.

Por tal razón se plantea como **idea a defender**: “Desarrollando un componente web para el manejo de la información de las ETA en el Departamento de Higiene y Epidemiología de la Atención Primaria de Salud se obtendrá un mejor acceso y control de los datos.”

Para dar cumplimiento al objetivo general se plantean las siguientes **Tareas de investigación**:

- Analizar el estado del arte a nivel nacional e internacional relacionadas con los sistemas de Enfermedades Transmitidas por Alimentos.
- Realizar la investigación y estudio del proceso Enfermedades Transmitidas por Alimentos en los departamentos de Higiene y Epidemiología de la Atención Primaria de Salud.
- Elaborar los documentos y artefactos correspondientes mediante la metodología definida para las fases de trabajo: Modelado de Negocio, Gestión de Requerimientos, Análisis y Diseño e Implementación del proceso de Enfermedades Transmitidas por Alimentos.
- Implementar las funcionalidades del proceso Enfermedades Transmitidas por Alimentos.

Para la realización de esta investigación, se siguió una **estrategia descriptiva**, donde se le da menor importancia a las causas que originan el problema, en el cual, el principal objetivo es la profundización teórica del planteamiento investigativo, describir el fenómeno, reflejar lo esencial y más significativo del mismo para llegar a los resultados esperados. Para obtener una solución concreta de la investigación, se hace uso del **Método Empírico**. En la realización de la investigación se hizo una **observación** del proceso de gestión de la información de las ETA dirigida a conocer las características fenomenológicas del mismo. Mediante la entrevista se logró comprender la lógica del negocio así como las principales problemáticas que impiden el mejor desarrollo de los procesos involucrados, lo que condujo al planteamiento del objetivo general. Se realizó la investigación desde fuera del campo que se estudia, es decir, teniendo en cuenta que el observador no pertenece al grupo de personas involucrado directamente en el mismo, lográndose la detección más eficaz de sus irregularidades. Con el empleo de guías preconcebidas como resultado de la observación, se garantizó completitud en la información adquirida.

Atendiendo también a criterios del observador, se logró una valoración cualitativa más integral pues no fue atada únicamente a parámetros fijados con anterioridad.

Este documento cuenta con cuatro capítulos que incluye todo lo relacionado con el trabajo investigativo, así como el diseño e implementación del sistema.

**Capítulo I:** Fundamentación Teórica: fundamenta la necesidad, el valor de la investigación y se hace referencia a la informatización de la APS. Se realiza un estudio de sistemas que resuelven problemas similares al planteado, así como cada una de las tecnologías, metodologías y herramientas utilizadas para el desarrollo de la aplicación.

**Capítulo II:** Características del sistema: describe la propuesta del sistema, identificando el proceso del negocio, las actividades que lo componen y las reglas que lo rigen, así como las mejoras que propone el mismo; además se describen cada una de las actividades identificadas y quienes son los encargados de llevar a cabo cada actividad. Se detallan las funcionalidades que el sistema debe ser capaz de cumplir.

**Capítulo III:** Diseño del sistema: describe los aspectos relacionados al diseño de la solución propuesta. Se estructura el modelo de diseño, se modelan los diagramas y se describen los patrones usados.

**Capítulo IV:** Implementación: trata los aspectos relacionados con la construcción de la solución propuesta. Aborda aspectos donde se define la implementación del sistema, la integración con otros módulos o sistemas y se describen el tratamiento de errores en la solución del sistema.

## Capítulo 1. Fundamentación Teórica

En el presente capítulo se realiza un estudio de la situación a nivel mundial de los sistemas informáticos existentes relacionados con las Enfermedades de Transmisión Alimentaria en la APS. Además se definen las tecnologías, herramientas y metodologías actuales, utilizadas para el desarrollo de la solución propuesta.

### 1.1 Sistema Nacional de Salud

En 1960 se realiza la transformación del Ministerio de Salubridad en Ministerio de Salud Pública y se crea el Sistema Nacional de Salud basado en la colaboración internacional, la orientación profiláctica y la participación de la comunidad. Con el paso del tiempo se han trazado nuevas perspectivas para lograr incrementar el nivel de la salud en la población, haciendo énfasis principalmente en la Atención Primaria de Salud. [7]

#### 1.1.1 La Atención Primaria de Salud

La Atención Primaria es integrada por policlínicos, consultorios y cualquier otra institución de carácter médico que se encargue del control de las enfermedades endémicas locales, programas expandidos de inmunizaciones, atención materno-infantil, disponibilidad de medicamentos básicos, promoción de la nutrición, tratamiento de las enfermedades comunes y saneamiento ambiental. La atención que se provee a la comunidad, al enfermo aislado de acuerdo a las necesidades y condiciones actuales existentes, la compra de nuevos equipos para brindar una atención mejorada y diferente y el incremento de actividades de prevención han logrado que la APS cobre gran importancia en el SNS. [8]

##### 1.1.1.1 Higiene y Epidemiología

La epidemiología es una rama de la ciencia médica que permite investigar cualquier problema de salud que afecte a comunidades humanas. Esta rama es parte de la superestructura y como tal, responde a la base económica que determina una estructura social determinada. De la higiene que se tenga al manipular los alimentos, del cuidado que se tenga con el ambiente, con la basura que diariamente se vierte en las calles, en las bahías, o con los gases que las fábricas indiscriminadamente vierten en la atmósfera, depende en muchos casos la proliferación de las epidemias. Por esta causa la higiene y la epidemiología están relacionadas directamente y en la actualidad se han creado muchas instituciones que

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

responden a esta asociación con el objetivo de elevar la higiene de la población, erradicando en muchos casos brotes epidémicos. [9]

## 1.2 Marco conceptual.

Para facilitar la comprensión de dicha investigación se deben entender primeramente los términos que se emplean en la ocurrencia de una ETA, motivo por el cual se especifican algunos conceptos que no son comunes en el lenguaje cotidiano.

***Intoxicación alimentaria:*** enfermedad que puede ser atribuida a un alimento específico, a una sustancia que se ha incorporado, a la contaminación del alimento a través de los recipientes o bien durante su preparación y distribución. [10]

***Brote de intoxicación alimentaria:*** cuando dos o más personas presentan un síndrome clínico similar después de consumir el mismo alimento con asociación de tiempo y lugar, si el análisis epidemiológico implica este alimento como origen de la enfermedad. Los brotes de intoxicación alimentaria pueden ser comunitarios o familiares: brotes comunitarios son aquellos que se han originado por ingestión en común, en un establecimiento público o institución, de alimentos contaminados. Brotes familiares son aquellos que se han originado por consumo de alimentos contaminados preparados en el hogar. [11]

***Vigilancia Epidemiológica:*** es la recogida, el análisis, la interpretación y la difusión sistemática y continua de datos sanitarios, con objeto de poder tomar las medidas de prevención y lucha pertinentes. En un sistema de vigilancia epidemiológica es esencial que la información recogida por el sistema sea exacta, comprensible y rápida, de forma que permita su diseminación precoz y la correspondiente acción pertinente. [12]

***Inocuidad de los alimentos:*** la garantía de que los alimentos no causarán daño al consumidor cuando se preparen y/o consuman de acuerdo con el uso a que se destinan. [13]

***Idoneidad de los alimentos:*** la garantía de que los alimentos son aceptables para el consumo humano, de acuerdo con el uso a que se destinan. [14]

***Manipulador:*** toda persona que manipule directamente alimentos envasados o no envasados, equipo y utensilios utilizados para los alimentos, o superficies que entren en contacto con los alimentos. [15]

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

**Prevención:** la prevención de la enfermedad es el conjunto de acciones realizadas para evitar que el daño o la enfermedad aparezcan, se prolonguen, ocasionen daños mayores o generen secuelas evitables. Existen tres niveles de prevención en los cuales deben participar los sistemas de salud donde quiera que operen. Estos niveles están inmersos en la definición e involucran para el nivel primario el evitar que la enfermedad haga aparición; en el segundo nivel se procura evitar que la enfermedad se prolongue u ocasione daños mayores, incluso secuelas; para el tercer nivel se pretende evitar que las secuelas agraven la situación de salud de las personas. [16]

**Agente etiológico:** entidad biológica, física o química capaz de causar enfermedad. [17]

**Transmisión:** traslado de un agente etiológico animado de una fuente primaria hacia un nuevo huésped. La transmisión puede ocurrir de manera directa o indirecta. [18]

## 1.3 Análisis de soluciones existentes

### 1.3.1 Ámbito Internacional

En la actualidad existen varios sistemas informáticos dirigidos a la salud, de ellos un grupo se centran en la automatización de la gestión de la información relacionada con las ETA. Esto se debe a la creciente necesidad de tener un control estricto sobre los brotes epidémicos de estas enfermedades. A continuación se especifican algunos de estos sistemas:

#### Epi-ETA

Epi-ETA es una red de epidemiólogos involucrados en la vigilancia de enfermedades transmitidas por alimentos. Su objetivo principal es promover la comunicación y la colaboración entre sus miembros. La red proporciona información y conocimiento científico, además ofrece un foro de comunicación electrónica y una plataforma para el adiestramiento en epidemiología de las ETA. Epi-ETA está abierta a los profesionales que trabajan en el campo de la epidemiología de las ETA. Aquellos que deseen ser miembros podrán suscribirse a la lista de servicios de la red, a través de Internet, para acceder en forma virtual a los foros virtuales, según los temas que resulten de interés, y también para participar de los cursos de entrenamiento de Epi-ETA. Este sistema solo brinda información, no es capaz de detectar ni dar seguimiento a un brote dado por una ETA. [19]

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

## **RILAA**

La Red Interamericana de Laboratorios de Análisis de Alimentos (por sus siglas en español RILAA) tiene como misión promover el desarrollo e interacción de los laboratorios analíticos para asegurar la inocuidad y calidad de los alimentos en el continente americano, a través de la armonización metodológica, la adopción de sistemas de gestión de calidad y la cooperación científico-técnica entre los países. RILAA está organizada en grupos técnicos de trabajo en calidad, análisis microbiológicos, y de residuos químicos y aditivos, que afianzan su labor a través de la implementación de grupos de discusión por correo electrónico. [20]

## **INOCUITO**

Inocuito es un sitio web que se implementó para transmitir los conceptos fundamentales de la inocuidad de los alimentos. Esta página solo brinda información de cómo conservar los alimentos para evitar un brote de las ETA a través de un personaje virtual que se convirtió en portavoz para los mensajes orientados a los niños y animó la página web con contenidos científicos. [21]

### **1.3.2 Ámbito Nacional**

#### **Sitio Web de Higiene y Epidemiología**

La Escuela Latinoamericana de Medicina cuenta con un sitio web de higiene y epidemiología, dirigido a sus estudiantes, profesionales y trabajadores. Su objetivo principal es contribuir al desarrollo del trabajo educativo y elevar la calidad de vida al brindar conocimientos esenciales, a través de la divulgación de informaciones sobre los diferentes aspectos que abarca la especialidad de higiene y epidemiología.

Constituye un espacio de comunicación y reflexión que permite divulgar y contribuir al conocimiento de aspectos tan importantes como las enfermedades transmisibles, las no transmisibles, el control de algunos vectores, el control sanitario internacional, la higiene de los alimentos, la higiene escolar, entre otros aspectos y con el que a su vez se contribuye a desarrollar la cultura computacional de los estudiantes y los trabajadores. Está diseñado en: Dreamweaver 4, Photoshop para editar las imágenes, Microsoft Word como Editor de texto de Microsoft Office y Acrobat Reader 5 para convertir de formato de texto de Word en formato PDF. A pesar del alto nivel promocional que el sitio ofrece, no cuenta con funcionalidades que permitan tabular informaciones o datos estadísticos que pudieran ser de gran ayuda para confeccionar tablas resúmenes de las enfermedades. [22]

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

## **SISTEMAAYUDA.**

El sistema de ayuda fue creado en el 2005 por el Centro de Desarrollo Informático para la Salud Pública (por sus siglas en español CEDISAP), pero nunca llevado a la práctica debido a que es un sistema que solo permite la identificación del agente causante del brote, dado los síntomas y el resultado del cálculo del período de incubación; su diseño no comprende el seguimiento de un brote ni la detección de los mismos.

## **DISEÑO DEL MÓDULO HIGIENE Y EPIDEMIOLOGÍA DEL SISTEMA INTEGRAL PARA LA ATENCIÓN PRIMARIA DE LA SALUD.**

La investigación se basa en la realización del análisis y diseño del Módulo de Higiene y Epidemiología del “Sistema Integral para la Atención Primaria de Salud” desarrollado en el Centro de Informática Médica, la misma expone el proceso relacionado con las ETA, en el que se ha profundizado durante el presente curso. Después de profundos estudios entorno al negocio se puede afirmar la existencia de nuevas actividades que son imprescindibles para lograr el desarrollo exitoso del mismo. A partir de esto se hace necesaria una nueva propuesta de negocio, requerimientos y análisis, por lo que la implementación de un sistema que integre todas las actividades identificadas se impone, permitiendo agilizar la gestión de las mismas.

### **1.4 Problema a Resolver**

El perfil epidemiológico y de salud se ha modificado con la integración de la Atención Primaria de Salud y el trabajo que vienen desarrollando los centros epidemiológicos. Para que se pueda desarrollar una vigilancia completa y adecuada se debe estar al tanto de brotes de enfermedades ocasionadas por factores ambientales. Además, se debe tener un control estricto de cada área de salud perteneciente a cada policlínico, realizar en las empresas o centros laborales inspecciones sanitarias que acrediten a estos libres de epidemias. [23]

Estos departamentos ejecutan todo el proceso de vigilancia creando diferentes programas para contrarrestar enfermedades, los cuales se sustentan en datos estadísticos que permiten comprobar si el número de epidemias en determinado momento ha aumentado o disminuido.

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

El estudio de las ETA es uno de los programas que se lleva cabo en los departamentos de higiene y epidemiología, este programa tiene como objetivo fundamental obtener, recolectar y analizar información necesaria y actualizada de las ETA; dicha información actualmente está recogida en planillas elaboradas para almacenar deficiencias, resultados o recomendaciones; documentos excel que almacenan tablas de informaciones y a la vez realizan pequeños cálculos. Estos documentos en general recogen las informaciones y datos estadísticos referentes a cada estudio, pero no satisfacen las necesidades que en la actualidad se demanda: disponibilidad real, tabulación de la información y acceso inmediato a la misma. Teniendo en cuenta que:

- La gestión de la información de los brotes se registra en documentos y planillas que con el paso del tiempo comienzan a deteriorarse. Esto puede provocar que si en algún momento se necesita consultar algún documento antiguo, no se pueda debido a que este se encuentre en mal estado y su contenido no se vea con claridad. Además, estos documentos pueden ser únicos y no estar almacenados en otro lugar, perdiéndose la información sin posibilidad de recuperarla nuevamente.
- La información que se maneja en estos departamentos se gestiona de forma manual, por vía telefónica y en algunos casos por correo electrónico en documentos word o tablas excel por los especialistas del departamento. Esto influye de cierta manera en el deficiente intercambio de los datos, puesto que las vías anteriores no son las más eficientes para realizar este tipo de proceso ya que demorarían las actividades que dependen de la información enviada. En muchos casos, no se podría asegurar que la información que llega por vía telefónica es auténtica y confiable por lo que se obstaculiza la búsqueda de variables cuando se necesita consultar o confeccionar reportes y sumarios solicitados.
- Al ser los canales de información poco fiables y eficientes, las informaciones que se envían de los estudios realizados en los Departamentos de Higiene y Epidemiología no se encuentran actualizadas y el acceso a ellas no ocurre inmediatamente, originando un atraso en la confección y envío de informes y reportes, semanales, mensuales y anuales a niveles superiores.

## **1.5 Tecnologías, herramientas y metodologías.**

Las tecnologías, herramientas y metodologías a usar para dar cumplimiento al objetivo general de la investigación serán las definidas por el Centro de Informática Médica para el desarrollo de sistemas informáticos. A continuación se exponen definiciones, características y ventajas de su utilización.

## 1.5.1 Estilos Arquitectónicos.

### **Patrón Modelo Vista Controlador (por sus siglas MVC)**

Con el fin de adquirir un diseño de clases correcto, así como una buena comunicación de sus componentes es utilizado el patrón Modelo Vista Controlador (por sus siglas en español MVC). El mismo se diseña con la finalidad de realizar una separación del Modelo, las Vistas y los Controladores, las cuales son entidades que se tratan por separado y cualquier cambio realizado en el modelo se genera automáticamente en la vista. Cuando es incorporado este modelo a cualquier diseño, las diferentes partes de un programa se pueden construir por separadas y luego se unirían en tiempo de ejecución.

#### **Modelo**

Todas las aplicaciones de software dejan a los usuarios manipular ciertos datos que proceden de una realidad sobre la que se pretende actuar. A estos datos en estado puro, que representan el estado de la realidad se les llama modelo: modelan la parte de la realidad sobre la que se desea actuar. El modelo no tiene conocimiento específico de los diferentes controladores y/o vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas, y notificar a las vistas cuándo deben reflejar un cambio en el modelo.

#### **Vista**

La vista es el objeto que maneja la presentación visual de los datos gestionados por el Modelo. Genera una representación visual del modelo y muestra los datos al usuario. Interacciona con el modelo a través de una referencia al propio modelo.

#### **Controlador**

Controla todo lo que puede realizar la aplicación. Responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Está compuesto por acciones que se representan con funciones en una clase. [24]

### **Desarrollo de Software Basado en Componentes (por sus siglas DSBC)**

El ser humano desde tiempos muy antiguos aprendió a reutilizar el conocimiento existente y en la actualidad esta estrategia está presente en el desarrollo de software, debido a la complejidad de los sistemas computacionales actuales.

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

El proceso de construcción de una pieza de software con componentes ya existentes, da origen al principio de reutilización del software, mediante el cual se promueve que los componentes sean implementados de una forma que permita su utilización funcional sobre diferentes sistemas en el futuro. [25]

Los beneficios más importantes de esta arquitectura son los siguientes:

**Reusabilidad de Servicios:** Reducción considerable de tiempos y costos de desarrollo de aplicaciones al utilizar servicios disponibles ya desarrollados, para resolver problemáticas comunes a otras aplicaciones. Aumentando por esta razón la robustez del nuevo sistema, al utilizarse software ya probado.

**Interoperabilidad de aplicaciones:** Disminución de la complejidad en el proceso de integración, pues se interactúa con elementos que se abstraen de la tecnología y ubicación de los servicios.

Se debe tener en cuenta que este tipo de arquitectura permite la integración de múltiples sistemas, lo que implica una dependencia en la información gestionada por cada uno de ellos. Los sistemas integrados deben convertir al software en una solución cualitativamente superior pero debe ser una relación de dependencia absoluta.

## 1.5.1 Tecnologías en la Capa de Presentación.

### JavaServer Faces (por sus siglas en inglés JSF) 1.2

La tecnología JSF constituye un marco de trabajo (framework) de interfaces de usuario del lado del servidor para aplicaciones web basadas en tecnología Java.

Toda aplicación JSF debe incluir una librería de etiquetas personalizadas que define las etiquetas que representan componentes UI, así como una librería de etiquetas para controlar otras acciones importantes, como validadores y manejadores de eventos. La librería de etiquetas de componentes elimina la necesidad de codificar componentes UI en HTML u otro lenguaje de marcas, lo que se traduce en el empleo de componentes completamente reutilizables. [26]

### RichFaces 3.3.1

RichFaces es una librería de componentes visuales para JSF. Además, RichFaces posee un framework avanzado para la integración de funcionalidades Ajax (por sus siglas en inglés Asynchronous Java Script and XML) en dichos componentes visuales, mediante el soporte de la librería Ajax4JSF. Se integra

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

perfectamente en el ciclo de vida de JSF y contiene un set de componentes visuales, los más comunes para el desarrollo de una aplicación web rica (Rich Internet Application), con un número bastante amplio que cubren casi todas las necesidades. [27]

## **Ajax4JSF.**

Ajax4jsf es una librería de código abierto que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax de forma limpia y sin añadir código Java Script. Ajax4jsf presenta mejoras sobre los propios beneficios del framework JSF incluyendo el ciclo de vida, validaciones, facilidades de conversión y el manejo de recursos estáticos y dinámicos. Permite definir un evento en una página que invoca una petición Ajax y luego las áreas de la página deberían sincronizarse con el árbol de componentes JSF después de que la petición Ajax cambie los datos en el servidor. [28]

## **Facelets 1.1**

Facelets es un framework simplificado de presentación, donde es posible diseñar de forma libre una página web y luego asociarle los componentes JSF específicos. Aporta mayor libertad al diseñador y mejora los informes de errores que tiene JSF. Permite crear plantillas para construir un árbol de componentes de forma que puedan definirse como composición de otros.

### **1.5.2 Tecnología en la Capa de Negocio**

#### **Servidor de Aplicaciones**

##### **Jboss Seam 2.1.1**

JBoss Seam es un framework que integra la capa de presentación (JSF) con la capa de negocios y persistencia (Enterprise Java Beans por sus siglas en inglés EJB). Con Seam basta agregar anotaciones propias de este a los objetos Entidad y Session de EJB, logrando escribir menos código Java y Xtensible Markup Language (por sus siglas en inglés XML).

Otra característica importante es que se pueden hacer validaciones en los Plain Old Object Java (por sus siglas en inglés POJOs) como además manejar directamente la lógica de la aplicación y de negocios desde los session beans.

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

## **Enterprise Java Beans 3.0**

Es una plataforma para construir aplicaciones de negocio portables, escalables, y reutilizables utilizando el lenguaje de programación java. Permite a los desarrolladores enfocarse en la lógica del negocio sin tener que emplear tiempo en la conformación del código de la infraestructura. Desde el punto de vista del desarrollador es un pedazo de código java que se ejecuta en un entorno especializado denominado contenedor EJB que proporciona un conjunto de servicios. El servicio de persistencia es proporcionado por un framework especializado denominado proveedor de persistencia.

## **Plain Old Java Object**

Usado por programadores java para enfatizar el uso de clases simples y que no dependen de un framework en especial.

### **1.5.3 Tecnologías en la Capa de Acceso a Datos.**

#### **Hibernate 3.3**

Hibernate es una herramienta de mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos XML que permiten establecer estas relaciones. Es software libre, distribuido bajo los términos de la licencia General Public License ( por sus siglas en inglés GNU) [31]

#### **Java Persistence API (por sus siglas en inglés JPA)**

Proporciona un modelo de persistencia basado en POJO's para mapear bases de datos relacionales en Java. El Java Persistence API fue desarrollado por el grupo de expertos de EJB 3.0, aunque su uso no se limita a los componentes software EJB. También puede utilizarse directamente en aplicaciones web y aplicaciones clientes; incluso fuera de la plataforma Java Enterprise Edition 5.0 (por sus siglas en inglés JEE).

En su definición, se han combinado ideas y conceptos de los principales frameworks de persistencia como Hibernate, Toplink y Java Data Objects, y de las versiones anteriores de EJB. Todos estos cuentan actualmente con una implementación JPA.

## 1.5.4 Lenguaje de programación.

Se utiliza Java como lenguaje de programación orientado a objetos, el cual fue desarrollado por Sun Microsystems a principio de los años noventa. Java hace uso de la sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir muchos errores, como la manipulación directa de punteros o memoria.

Tiene como ventaja que es un lenguaje multiplataforma, que se ha extendido cobrando cada día más importancia tanto en el ámbito de Internet como en la informática en general.

Java permite programar páginas web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema.

## 1.5.5 Herramientas a utilizar.

### Eclipse Galileo 3.5.2

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma. La definición que da el proyecto Eclipse acerca de su software es: "una especie de herramienta universal, un entorno de desarrollo integrado abierto y extensible para todo y nada en particular". Eclipse es, en el fondo, únicamente un almacén (workbench) sobre el que se pueden montar herramientas de desarrollo para cualquier lenguaje, mediante la implementación de los plugins adecuados.

### Visual Paradigm suite 6.4

Visual Paradigm para el Lenguaje de Modelamiento Unificado (por sus siglas en inglés UML) 2.0 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. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. VP-UML aporta a los desarrolladores de software una plataforma de desarrollo puntera para construir aplicaciones de calidad, mejores y más baratas con rapidez. Aporta una excelente interoperabilidad con otras herramientas CASE y muchos de los entornos de desarrollo integrado líderes del mercado. [32]

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

## **Toad Data Modeler 3.3.8.11**

Toad Data Modeler es una estupenda ayuda a la hora de diseñar y crear bases de datos. No solamente permite crear clase de esquemas, diagramas y diseños, sino que también genera el código SQL para construirlas. Quizá la mayor ventaja de Toad Data Modeler sea la versatilidad, ya que soporta cualquier sistema gestor de bases de datos: Access, Firebird, InterBase, MySQL, Oracle, Paradox, Postgre, Sybase, etc. Toad Data Modeler el cual facilita el proceso de creación de una base de datos desde los cimientos: construyendo el diagrama entidad-relación y definiendo las reglas de integridad referencial. [33]

## **1.5.6 Sistema gestor de Bases de Datos**

PostgreSQL 8.4, este es el gestor más avanzado de código abierto en la actualidad, ofrece servicios de control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo sub-consultas, transacciones, y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, perl, tcl y python). Es extensible, brinda gran estabilidad y confiabilidad y es multiplataforma. Tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/PGSQL. Este lenguaje es comparable al lenguaje procedural del sistema de gestión de base de datos relacional Oracle, PL/SQL. Una de sus ventajas es la excelente documentación que existe, además de contar con una comunidad de usuarios y desarrolladores a los que acudir en caso de tener problemas.

## **1.5.7 Guía de desarrollo**

Debido a los constantes cambios que están sucediendo en cuanto a las herramientas, además de destacar las necesidades existentes en la Universidad, la adaptabilidad en la misma y los recursos con los que se dispone en la actualidad entre otras características; se estableció de acuerdo con las políticas productivas de la universidad utilizar el proceso de mejora con el objetivo de alcanzar el nivel 2 de Capability Maturity Model Integration (por sus siglas en inglés CMMI).

### **Modelo para la mejora y evaluación de procesos**

Capability Maturity Model Integration es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. Las mejores prácticas de CMMI se publican en documentos llamados modelos, los cuales contienen el conjunto de prácticas relacionadas que son ejecutadas de forma conjunta para conseguir determinados objetivos.

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

Así es como el modelo CMMI establece una medida del progreso, conforme al avance en niveles de madurez. Cada nivel a su vez cuenta con un número de áreas de proceso que deben lograrse. El alcanzar estas áreas o estadios se detecta mediante la satisfacción o insatisfacción de varias metas claras y cuantificables.

CMMI consta de 22 áreas de procesos distribuidas dentro de 5 niveles de madurez:

- ➔ Nivel 1: Inicial.
- ➔ Nivel 2: Administrado.
- ➔ Nivel 3: Definido.
- ➔ Nivel 4: Cuantitativamente administrado.
- ➔ Nivel 5: Optimizado.

Las organizaciones son evaluadas y reciben una calificación de nivel 1-5 siguiendo los niveles de madurez. Este enfoque se denomina Representación Escalonada.

La Universidad de las Ciencias Informáticas se encuentra desarrollando un proceso de mejora con el objetivo de obtener el Nivel 2 de CMMI. Las áreas de procesos que lo forman son:

- ➔ Administración de Requisitos (REQM).
- ➔ Planeación del Proyecto (PP).
- ➔ Monitoreo y Control del Proyecto (PMC).
- ➔ Medición y Análisis (MA).
- ➔ Aseguramiento de la Calidad de Procesos y productos (PPQA).
- ➔ Administración de la Configuración (CM).
- ➔ Administración de Acuerdos con Proveedores (SAM).

Dentro de las áreas de procesos mencionadas anteriormente la Universidad se encuentra enfocada en la Administración de Requisitos.

El objetivo de la Administración de Requisitos es gestionar los requisitos de los elementos del proyecto y sus componentes e identificar inconsistencias entre estos requisitos, el plan de proyecto y los elementos

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

de trabajo. En este proceso se deben gestionar todos los requisitos del proyecto, tanto técnicos como no técnicos. Estos requisitos han de ser revisados conjuntamente con la fuente de los mismos así como con las personas que se encargarán del desarrollo posterior. Para llevar a cabo estas actividades es utilizado el documento IPP- 3510:2009 Libro de Proceso para la Administración de Requisitos realizado por la universidad y cuyo objetivo es definir el proceso de administración de requisitos.

Este documento establece el ciclo de vida a seguir asociado a los proyectos involucrados en el proceso de mejora, el cual consta de 9 fases (Estudio Preliminar, Modelación del Negocio, Requisitos, Análisis y Diseño, Pruebas Internas, Pruebas de Liberación, Despliegue y Soporte). En este se establece por cada fase la relación con los subprocesos descritos en el libro de procesos específico para el área de Administración de Requisitos.

La descripción del IPP-3510:2009 Libro de Proceso para la Administración de Requisitos incluye la definición de roles, sus responsabilidades y las habilidades en la ejecución de las actividades de los distintos procesos y los productos típicos de trabajos que se obtienen como resultado de la ejecución de dichas actividades. [34]

## **1.5.8 Lenguaje Unificado de Modelado.**

El Lenguaje Unificado de Modelado es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

## **Notación Utilizada para Modelar los Procesos del Negocio (por sus siglas en inglés BPMN)**

La Notación para el Modelado de Procesos de Negocio es un nuevo estándar que permite el modelado de procesos de negocio, en un formato de flujo de trabajo. BPMN proporciona a los negocios la capacidad de entender sus procedimientos internos en una notación gráfica, facilitando a las organizaciones la habilidad para comunicar esos procedimientos de una manera estándar.

En este capítulo se profundizó en el conocimiento de algunos conceptos relacionados con las ETA necesarios para la comprensión de las actividades identificadas. Se realizó un estudio de los principales sistemas informáticos dirigidos a las ETA, a nivel nacional e internacional, llegando a la conclusión que los mismos no cumplen con el problema planteado. Además se abordó sobre las tecnologías, metodologías, lenguajes y herramientas que serán utilizadas a lo largo del desarrollo del sistema propuesto.

### Capítulo 2. Características del Sistema.

En este capítulo se detallará el negocio del proceso involucrado, haciendo un profundo y extenso análisis de las actividades que serán objeto de automatización. De este análisis se obtendrán los documentos que contienen la información gestionada para el correcto funcionamiento de esta proceso, algunos de ellos son los modelos del negocio y los diagramas correspondientes, describiéndose los actores y trabajadores que intervienen en los mismos. Además, se especifican los requerimientos funcionales y no funcionales; mostrando claramente los aspectos fundamentales que darán paso a la descripción de la solución que se propone.

#### 2.1 Modelo de Negocio

Un proceso de negocio consiste en una colección de actividades que son realizadas coordinadamente en un ambiente técnico y organizacional. La conjunción de estas actividades logra un objetivo del negocio. Los procesos de negocio toman una o más tipos de entradas (precondiciones) y crea una salida (poscondición) que posee un determinado valor para el cliente.

El modelado del negocio es la técnica usada para lograr un mayor entendimiento de los procesos del negocio de la organización. Los objetivos que se persiguen con la realización de este modelo son: comprender la estructura de la organización, identificar mejoras potenciales y entender los problemas actuales, asegurarse de que los clientes, usuarios finales y desarrolladores tengan una idea común de la organización y obtener los requerimientos del sistema a partir del modelo de negocio.

Para dicho modelado se utiliza la Modelo de Procesos de Negocios. La misma aporta un buen entendimiento del negocio, facilita la comunicación entre el diseño y la implementación de los procesos del negocio.

#### 2.2 Diagrama de proceso del negocio

##### 2.2.1 Enfermedades Transmitidas por Alimentos

# CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

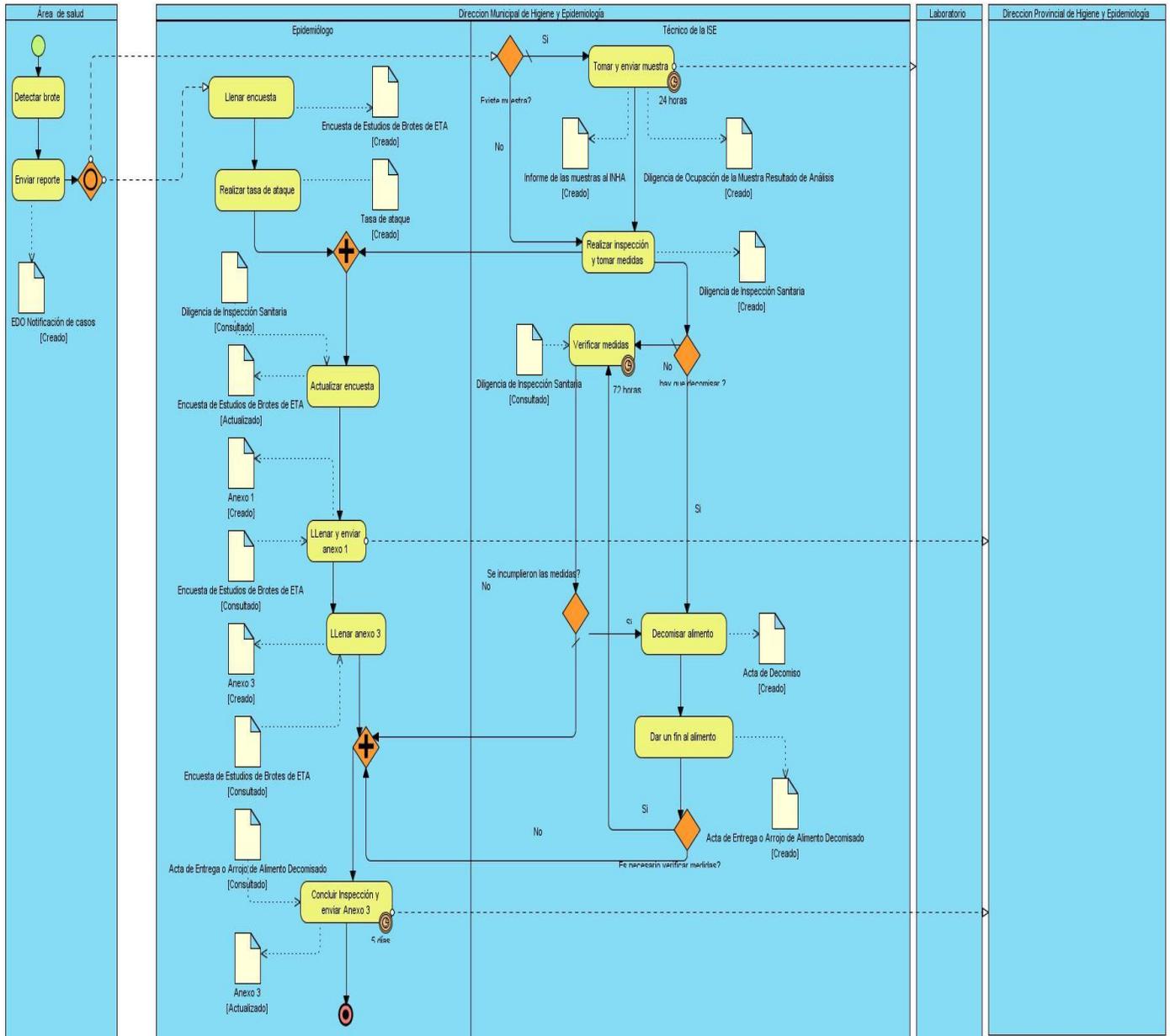


Figura 1. Diagrama de Procesos del Negocio Enfermedades Transmitidas por Alimentos.

## 2.3 Flujo actual del proceso involucrado.

### 2.3.1 Proceso de Enfermedades Transmitidas por Alimentos.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

Nombre:	P5_Proceso Enfermedades Transmitidas por Alimentos.
Objetivos:	Ver qué tipo de alimento fue el que provoco el daño a las personas y evitar la proliferación de la enfermedad.
Evento(s) que lo generan:	Que exista un brote de Intoxicación Alimentaria.
Precondiciones:	El Equipo Básico de Salud y el Grupo Básico de Salud notifiquen a la dirección municipal de Higiene y Epidemiología una Enfermedad de Declaración Obligatoria.
Poscondiciones:	Creación de Anexo 1. Creación de Anexo 3. Creación de Encuesta de Estudios de Brotes de ETA. Creación de Diligencia de Inspección Sanitaria.
Reglas de Negocio:	Regla de negocio (Regla Textual # 5, 19 y 20) Ver documento AAPS_SIAPS_0116_RNeg_HigEpid_W.
Responsables:	Un miembro de HE y el técnico de la ISE.
Clientes internos:	CESIM.
Clientes externos:	Paciente.
Entradas:	Enfermedades de Declaración Obligatoria Notificación de Casos.
Salidas:	Anexo3. Anexo1. Acta de Decomiso. Encuesta de Estudios de Brotes de ETA. Diligencia de Inspección Sanitaria. Información del laboratorio del INHA. Acta de Entrega o Arrojo de Alimentos Decomisados. Diligencia de ocupación de Muestra Resultado de Análisis.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

Actividades:	<ol style="list-style-type: none"><li>1- Detectar brote.</li><li>2- Enviar reporte.</li><li>3- Llenar encuesta.</li><li>4- Realizar tasa de ataque.</li><li>5- Tomar y enviar muestra.</li><li>6- Realizar inspección y tomar medidas.</li><li>7- Actualizar encuesta.</li><li>8- Llenar Anexo 1.</li><li>9- Verificar medidas.</li><li>10- Llenar Anexo 3.</li><li>11- Decomisar alimento.</li><li>12- Dar un fin al alimento.</li><li>13- Concluir inspección y enviar anexo 3.</li></ol>
--------------	---

### 2.3.1.1 Estudio de Enfermedades Transmitidas por Alimentos. Descripción Textual.

El Proceso de Enfermedades Transmitidas por Alimentos tiene como primera actividad **Detectar el brote**. Estos pueden ser detectados de varias formas:

- 1- Cuando una persona se da cuenta que otras personas han enfermado en forma similar después de consumir un mismo alimento o haber asistido a un mismo evento y lo comunica al policlínico correspondiente.
- 2- Cuando un médico o enfermera se da cuenta que ha visto más casos que el número usual de pacientes con la misma enfermedad y lo reporta tomando las medidas correspondientes.

Una vez detectado el brote se procede a **enviar el reporte** de Enfermedades de Declaración Obligatoria y Notificación de Casos este se hace llegar de inmediato a la UMHE correspondiente donde se forma un equipo para el estudio y seguimiento del mismo. Estos reportes no siempre proceden de la misma fuente; su procedencia depende de la forma en la que se detectan los brotes, por lo que el reporte puede ser enviado desde un Consultorio Médico para la Familia (CMF) o desde un policlínico. El equipo formado por

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

---

técnicos y especialistas en Higiene y Epidemiología se dirige al lugar donde ocurre la intoxicación alimentaria para realizar las investigaciones necesarias.

El epidemiólogo como parte de este equipo **encuesta** a las personas afectadas y a las no afectadas; dentro de este grupo de personas se encuentran los manipuladores siendo estos los de mayor riesgo ya que son los que están en contacto directo con el alimento después de su último tratamiento térmico. También es de vital importancia para el epidemiólogo conocer los alimentos ingeridos en las últimas 24 horas así como el comienzo de los primeros síntomas. Una vez recogido estos datos se calcula el período de incubación, la **tasa de ataque** y se desglosa la cantidad de enfermos por grupos de edades y sexo. (Ver adjunto 1y 1.a).

Con la culminación de la encuesta el epidemiólogo puede dar un diagnóstico presuntivo del agente causante de la intoxicación y confeccionar la curva epidémica. Paralelo al llenado de la encuesta el técnico de la ISE realiza la **toma y envío de muestras** de los alimentos ingeridos en caso de que exista. Esta muestra es enviada al laboratorio de Microbiología de Higiene y Epidemiología en conjunto con la planilla de Ocupación de la Muestra Resultados de Análisis (ver adjunto 4) y el Informe de las Muestras al INHA, en estos documentos se especifica el nombre del producto, la cantidad de muestras tomadas, el tipo de análisis que se le debe realizar, tipo de envase y si este va abierto o cerrado, así como otros datos ya recogidos en la encuesta.

Luego el técnico de la ISE procede a **realizar la inspección** al lugar donde se elaboraron los alimentos ingeridos. De acuerdo a las deficiencias sanitarias observadas procede a dar los plazos correspondientes para la solución de las mismas advirtiendo al interesado que en caso de incumplimiento se procederá con las leyes vigentes; es válido aclarar, que si no existe muestra de alimentos la inspección es lo primero que realizaría el técnico de la ISE. Las medidas tomadas durante la inspección además de ser recogidas en el documento Diligencia de Inspección Sanitaria son consultadas por el epidemiólogo y plasmadas en el acápite Medidas Tomadas dentro del documento Encuesta ETA Modelo 84-22.

La información recogida en la encuesta y las medidas aplicadas en la inspección son utilizadas por el epidemiólogo para la **confección del Anexo 1** (ver adjunto 2), este es enviado a la dirección Provincial de Higiene y Epidemiología; a partir de este momento se cuenta con 5 días para enviar el Anexo 3. Durante este período el técnico de la ISE **verifica las medidas** tomadas durante la inspección, actividad que debe realizarse en no más de 72 horas. De esta forma comprueba si las deficiencias encontradas fueron

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

---

resueltas, de no ser así, se procede a la aplicación de los decretos ley correspondiente. Si en uno de ellos se establece el **decomiso del alimento**, se llena por cada alimento la planilla Acta de Decomiso, posteriormente se decide el **destino del alimento** decomisado registrándose por cada 5 alimentos una planilla de Acta de Entrega o Arrojo Decomisado. Paralelo a estas acciones se llena el Anexo 3 (ver adjunto 3), este es un documento que resume el estudio del brote. Al finalizar el estudio el epidemiólogo actualiza el Anexo 3 con las conclusiones de la inspección y es enviado a la dirección provincial de Higiene y Epidemiología.

### 2.4 Propuesta del sistema

#### 2.4.1 Requerimientos

El Libro de Proceso para la Administración de Requisitos propone como una de las fases para el ciclo del desarrollo de un software el de Requisitos, el cual tiene su mayor peso en la fase de inicio, teniéndose como artefactos fundamentales de esta fase la Especificación de Requisitos, Descripción de los Requisitos Funcionales y las Salidas del Sistema.

#### 2.4.2 Especificación de los Requerimientos de Software

Los requerimientos de software son condiciones o capacidades que tiene que alcanzar o poseer un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente. Estos posibilitan determinar la solución tecnológica a algún problema existente, logrando como resultado, un informe detallado de los requerimientos necesarios para la solución a dicho problema, que permita alcanzar los objetivos y solucionar los problemas existentes. [35]

Los requisitos se pueden clasificar en: funcionales y no funcionales, donde la primera clasificación corresponde a las especificaciones del sistema, mientras que los últimos expresan las cualidades que el producto debe tener.

#### 2.4.3 Requerimientos funcionales

Los requerimientos funcionales especifican capacidades o condiciones que el sistema debe cumplir, sin tomar en consideración ningún tipo de restricción física, de manera que especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. [36]

Después de analizados los procesos del negocio, se han definido los siguientes requisitos funcionales:

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

---

### **Gestionar encuesta de persona enferma.**

Registrar encuesta.

Mostrar encuesta.

Modificar encuesta.

Mostrar curva epidémica.

Mostrar tasa de ataque.

### **Gestionar datos aspecto medidas.**

Eliminar medidas tomadas.

Registrar medidas tomadas.

Buscar medidas tomadas.

Modificar medidas tomadas.

### **Gestionar datos etiología.**

Eliminar etiología

Registrar etiología

Buscar etiología

Modificar etiología

### **Gestionar datos grupo de edades.**

Eliminar grupo de edades.

Registrar grupo de edades.

Buscar grupo de edades.

Modificar grupo de edades.

### **Gestionar datos alimento.**

Eliminar alimento.

Registrar alimento.

Buscar alimento.

Modificar alimento.

### **Gestionar datos síntoma intoxicación.**

Eliminar síntoma intoxicación.

### **Gestionar datos factor de contaminación.**

Eliminar factor de contaminación.

Registrar factor de contaminación.

Buscar factor de contaminación.

Modificar factor de contaminación.

### **Gestionar datos factor de sobrevivencia.**

Eliminar factor de sobrevivencia.

Registrar factor de sobrevivencia.

Buscar factor de sobrevivencia.

Modificar factor de sobrevivencia.

### **Gestionar datos factor de multiplicación.**

Eliminar factor de multiplicación.

Registrar factor de multiplicación.

Buscar factor de multiplicación.

Modificar factor de multiplicación.

### **Gestionar datos del lugar de pérdida de**

Eliminar del lugar de pérdida de inocuidad.

Registrar del lugar de pérdida de inocuidad.

Buscar del lugar de pérdida de inocuidad.

Modificar del lugar de pérdida de inocuidad.

### **Gestionar datos grupo de alimento.**

Eliminar grupo de alimento.

Registrar grupo de alimento.

Buscar grupo de alimento.

Modificar grupo de alimento.

### **Gestionar datos tipo de envase.**

Buscar tipo de envase.

Agregar tipo de envase.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

---

Registrar síntoma intoxicación.

Buscar síntoma intoxicación.

Modificar síntoma intoxicación.

### **Gestionar datos grupo por especie.**

Eliminar ingrediente datos grupo por especie.

Registrar datos grupo por especie.

Buscar datos grupo por especie.

Modificar datos grupo por especie.

### **Gestionar datos tipo de muestra.**

Eliminar tipo de muestra.

Registrar tipo de muestra.

Buscar tipo de muestra.

Modificar tipo de muestra.

### **Gestionar datos tipo de análisis.**

Eliminar tipo de análisis.

Registrar tipo de análisis.

Buscar tipo de análisis.

Modificar tipo de análisis.

### **Gestionar datos lugar de confección.**

Buscar lugar de confección del alimento.

Agregar lugar de confección del alimento.

Modificar lugar de confección del alimento.

Eliminar lugar de confección del alimento.

Modificar tipo de envase.

Eliminar tipo de envase.

### **Gestionar datos forma de venta.**

Buscar forma de venta.

Agregar forma de venta.

Modificar forma de venta.

Eliminar forma de venta.

### **Insertar diligencia de inspección sanitaria.**

Ver detalles de la diligencia de inspección sanitaria.

Modificar diligencia de ocupación de la muestra.

### **Insertar informe del INHA.**

Ver detalles de informe del INHA.

Mostar de informe del INHA.

### **Insertar acta de decomiso o arroj.**

Ver detalles acta de decomiso o arroj.

Mostar acta de decomiso o arroj.

### **Llenar diligencia de ocupación de la muestra.**

Ver detalles diligencia de ocupación de la muestra.

Mostar diligencia de ocupación de la muestra.

### **Buscar documentos asociados al brote.**

### **Generar anexo 1**

### **Generar anexo 3**

#### **2.4.4 Requerimientos no funcionales.**

Los requerimientos no funcionales tienen que ver con las propiedades o cualidades que de una forma u otra limitan al sistema y que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

---

requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir, una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

### **Fiabilidad**

- Las informaciones relacionadas con los brotes que vayan a ser intercambiadas con otros centros de higiene y epidemiología por la red pública, viajarán cifradas para evitar accesos o modificaciones no autorizadas.
- Se mantendrá seguridad y control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas podrán cambiarse solo por el propio usuario o por el administrador del sistema.
- Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantizando sólo la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión.
- Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.
- Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD, independientemente de que para el sistema, este elemento ya no exista.
- El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.

### **Eficiencia**

- El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria. Para ello se potenciará como regla guardar en la memoria caché datos y recursos de alta demanda.
- El sistema respetará buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos. Se deberá usar siempre que sea posible el patrón Singleton, destruir referencias que ya no estén siendo usadas, optimizar el trabajo con cadenas, entre otras buenas prácticas que ayudan a mejorar el rendimiento.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

---

### **Soporte**

- Se permitirá administración remota, monitoreo del funcionamiento del sistema en los centros relacionados con la Atención Primaria para la Salud y detección de fallas de comunicación. Se permitirá realizar copias de seguridad de la base de datos hacia otro dispositivo de almacenamiento externo, además de recuperar la base de datos a partir de los respaldos realizados. Se permitirá el chequeo de las operaciones y acceso de los usuarios al sistema. Se permitirá establecer nomencladores de configuración del sistema y actualización de éstos.

### **Restricciones de diseño**

- La capa de presentación contendrá todas las vistas y la lógica de la presentación. El flujo web se manejará de forma declarativa y basándose en definiciones de procesos del negocio. La capa del negocio mantendrá el estado de las conversaciones y procesos del negocio que concurrentemente pueden estar siendo ejecutados por cada usuario. La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas. El acceso a datos está basado en el estándar JPA y particularmente en la implementación del motor de persistencia Hibernate.

### **Interfaz**

#### **Interfaces de usuario**

- Las ventanas del sistema contendrán los datos claros y bien estructurados, además de permitir la interpretación correcta de la información. La interfaz contará con teclas de acceso rápido y menús desplegables que faciliten y aceleren su utilización. La entrada de datos incorrecta será detectada claramente e informada al usuario. Todos los textos y mensajes en pantalla aparecerán en idioma configurado.

#### **Interfaces de comunicación**

- Para el intercambio electrónico de datos entre aplicaciones se usará el protocolo Health Level Seven (por sus siglas en inglés HL7). El sistema usará el formato estándar Web Services Description Language para la descripción de los servicios web. También implementará mecanismos de encriptación de datos para el intercambio de información con sistemas externos. Además

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.

---

utilizará mecanismos de compactación de los datos que se intercambiarán con sistemas externos con el objetivo de minimizar el tráfico en la red y economizar el ancho de banda.

### Requerimientos de hardware

#### Estaciones de trabajo

- En la solución se incluyen estaciones de trabajo para las consultas del Sistema para la Atención Primaria alas SIAPS, las que necesitan capacidad de hardware que soporte un sistema operativo que cuente con un navegador actualizado y que siga los estándares web, se recomienda IE 7, Firefox 2 o versiones superiores. Por lo que se escogieron estaciones de trabajo e 256 Mb de memoria RAM y un microprocesador de 2.0 Hz con sistema operativo GNU/Linux.

#### Servidores

- La solución estará conformada, fundamentalmente, por servidores de alta capacidad de procesamiento y redundancia, que permitan garantizar movilidad y residencia de la información y las aplicaciones bajo esquemas seguros y confiables. Servidores de Base de datos: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual- Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux. Servidores de Aplicaciones: 2 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux. Servidores de Intercambio: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 2 GB de memoria y 2x72GB de disco y sistema operativo Linux.

#### Requerimientos de software

- El sistema debe correr en sistemas operativos Windows, Unix y Linux, utilizando la plataforma JAVA (Java Virtual Machine, JBoss AS y PostgreSQL). El sistema deberá disponer de un navegador web, estos pueden ser IE 7, Opera 9, Google chrome 1 y Firefox 2 o versiones superiores de estos.

El desarrollo de este capítulo permitió comprender el funcionamiento del negocio, identificar los procesos a automatizar y a partir de ahí se detectaron los requerimientos funcionales y no funcionales, brindando un mejor entendimiento del sistema a construir. El resultado de estos dos flujos de trabajo y en específico el de requerimientos permiten comenzar con las actividades del flujo de trabajo de Diseño.

### Capítulo 3. Diseño del Sistema

El diseño de un sistema de software tiene el propósito de formular los modelos que se centran en los requisitos no funcionales y en el dominio de la solución y que prepara para la implementación y prueba del sistema; contribuye a la definición de una arquitectura estable, sólida, lo que permite básicamente una especificación que describe cómo implementar el sistema. El Diseño es la única manera de materializar con precisión los requerimientos del cliente.

El objetivo de este capítulo está orientado a lograr un mejor entendimiento de los elementos relacionados con el diseño del sistema se muestran los diagramas de clases del diseño, se especifica la estructura y definición de los elementos que este posee y además se justifica el uso de alguno de los patrones de diseño utilizados.

#### 3.1 Modelos de diseño.

El modelo de diseño es un modelo de objetos centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con los lenguajes de programación, tienen impacto en el sistema a considerar. Sirve de abstracción para actividades de implementación, mediante la captura de los requisitos o subsistemas individuales, interfaces y clases. En la elaboración del diseño se utilizan un grupo de patrones o modelos para tener una comprensión más simple de la arquitectura del sistema. [38]

#### 3.2 Patrones de diseño.

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño es un modelo que se puede seguir para la realización de una actividad determinada, es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características.

El modelo de datos se encarga del almacenamiento de los datos, su estructura y las relaciones entre los mismos. Para la administración de los datos se utiliza el framework Hibernate. El cual tiene como ventajas el control de la accesibilidad a la información almacenada, así como la realización de rápidas consultas.

## CAPÍTULO 3. DISEÑO DEL SISTEMA.

---

Hibernate implementa una serie de patrones para la administración del modelo de datos, entre ellos están los siguientes:

- **Active record:** Permite la asociación de filas únicas de la base de datos con objetos del lenguaje de programación usado.
- **Identity map:** Mantiene referencias a los objetos cargados y devuelve las mismas cada vez que se solicita un objeto ya cargado. De esta forma, actúa como un caché dentro de la transacción, lo cual también contribuye a reducir el número de accesos a la base de datos.
- **Identity field:** Permite mapear las relaciones entre referencias entre objetos y claves entre tablas.
- **Foreign key mapping:** Permite mapear las relaciones de asociación de clave foránea.
- **Association Table mapping:** Permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, lo cual crea una base de datos orientada a objetos virtuales, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).
- **Lazy load:** Carga los distintos componentes de una clase a medida que son solicitados.
- **Query object:** Permite manipular las consultas en tiempo de ejecución. Dado que las consultas son los objetos, se pueden cambiar para agregar o quitar condiciones. Esto proporciona la flexibilidad necesaria para crearlas o modificarlas dinámicamente.

Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Entre estos tipos de patrones se pueden destacar los siguientes:

- **Experto:** La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada.

- **Creador:** Este patrón como su nombre lo indica es el que crea, el que guía la asignación de responsabilidades relacionadas con la creación de objetos, asigna la responsabilidad de que una clase B cree un Objeto de la clase A.
- **Alta cohesión:** La cohesión es una medida de la fuerza con la que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento. Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable, una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo.
- **Bajo acoplamiento:** El acoplamiento de una clase es el conjunto de dependencias que tiene con otras clases. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la menor repercusión posible en el resto de las clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.
- **Controlador:** Asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

### 3.3 Definición de Elementos de Diseño.

La representación de las clases del diseño está asociada al uso de UML para el modelado de Aplicaciones WEB, siendo identificadas las siguientes clases: Página Servidor (Server Page), Página Cliente (Client Page) Formulario (Form), para la representación de las clases contenedoras del código de las páginas servidoras, clientes y los formularios respectivamente.

Para modelar los diagramas de paquetes y los de clases del diseño se utilizará la siguiente nomenclatura:

Diagrama de paquetes: Diagrama de paquetes\_<Nombre del módulo>, Diagramas de clases del diseño: DCD\_<Nombre de la funcionalidad>.

La clase contenedora del framework JSF se encarga de generar el código de las páginas clientes (<<construye>>), los formularios componen las páginas clientes (permite que entre las páginas clientes y sus formularios exista la relación de composición), a través de los cuales se muestran e introducen los

## CAPÍTULO 3. DISEÑO DEL SISTEMA.

---

datos necesarios, los cuales son enviados hacia el contenedor del framework JSF que construyó la página cliente asociada (<<envía>>). Las páginas clientes incluyen clases con ficheros JavaScript y librerías (<<incluyen>>).

Para representar las clases previamente mencionadas se procedió usando la siguiente nomenclatura, frmsiaps, <Nombre de la vista>.siaps, CR<Nombre de la Opción>.java (Exportar), CC<Nombre de la Opción>.java (Registrar), CL<Nombre de la Opción>.java (Listar), CM<Nombre de la Opción>.java (Modificar), CD<Nombre de la Opción>.java (Detalles) y <Tr o Tn o Tb><Nombre de la Opción>.java, para los formularios, páginas clientes, clases servidoras y entidades respectivamente ubicadas en los diferentes niveles de la aplicación.

En los diagramas se encuentran otros elementos como son los frameworks JSF, EJB 3.0, JasperReport, JPA, Hibernate y Seam. Además, existen un conjunto de clases entre las que se identifican librerías, JavaScript, JSF (Servlets), Seam (Servlets), Bitácora, entre otras.

### **3.3.1 Diagrama de Clases del Diseño.**

#### **Gestionar Encuesta ETA.**

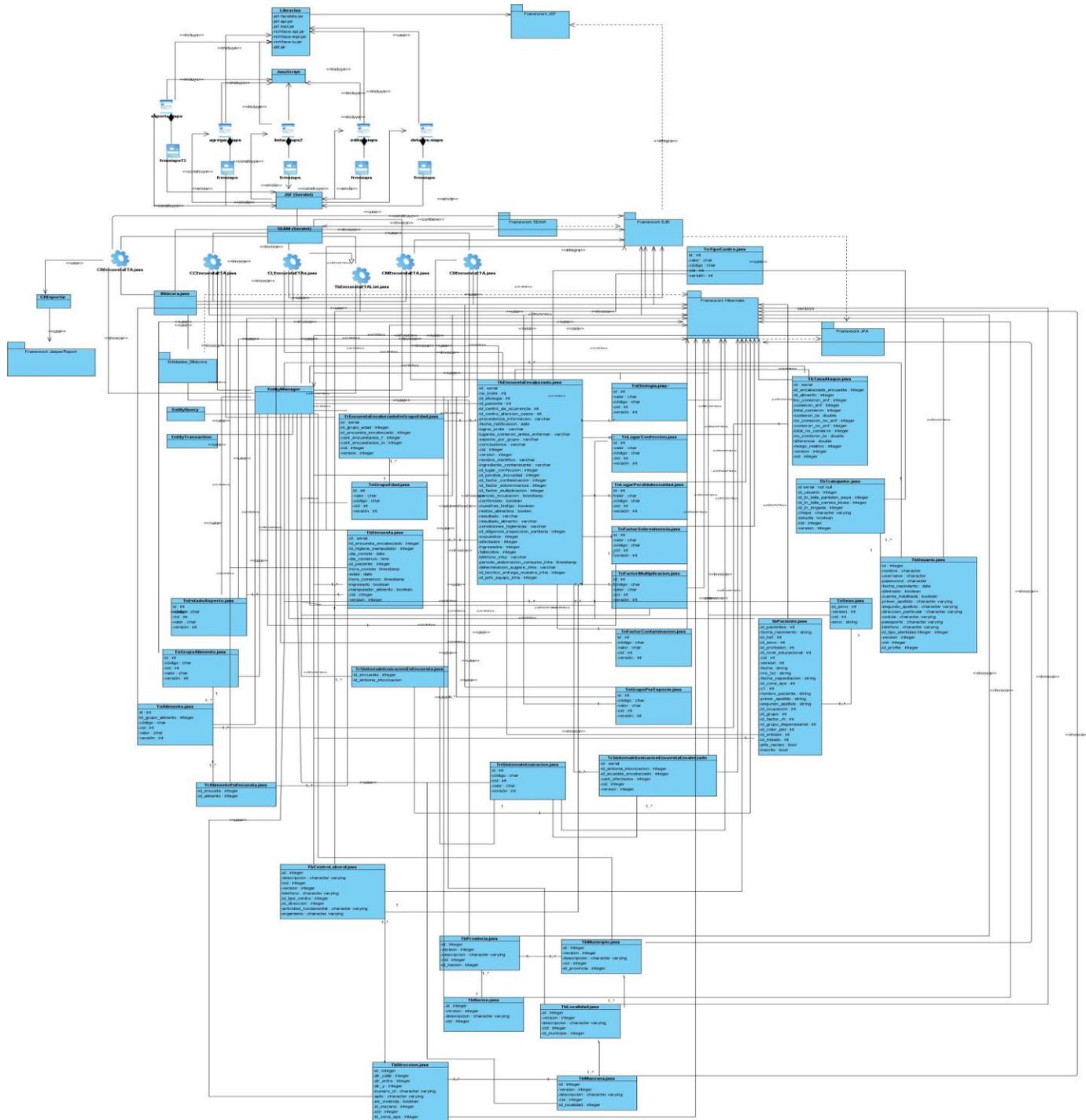


Figura 2. Diagrama de clases del diseño. Gestionar Encuesta ETA.



### 3.3.2 Descripción textual.

#### 3.3.1.1 Clases comunes. Descripción.

<b>Capa de Presentación</b>	
<p><b>Nombre:</b></p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p style="text-align: center; background-color: #e0f0ff;"><b>JavaScript</b></p> </div> <p>Figura 3. Clase JavaScript</p>	<p><b>Propósito:</b></p> <p>Proveer el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas.</p> <p><b>Descripción:</b></p> <p>Permite realizar las validaciones del lado del cliente y construir páginas más dinámicas integradas a un navegador web.</p>
<p><b>Nombre:</b></p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p style="text-align: center; background-color: #e0f0ff;"><b>Librerías</b></p> <ul style="list-style-type: none"> <li>-jsf-facelets.jar</li> <li>-jsf-api.jar</li> <li>-jsf-impl.jar</li> <li>-richface-api.jar</li> <li>-richface-impl.jar</li> <li>-richface-iu.jar</li> <li>-jstl.jar</li> </ul> </div> <p>Figura 4. Clase Librerías</p>	<p><b>Propósito:</b></p> <p>Permite la creación de interfaces.</p> <p><b>Descripción:</b></p> <p>Contiene un conjunto de librerías que permiten construir una interfaz de usuario con componentes reutilizables y extensibles. Reducen significativamente la carga de construir y mantener aplicaciones web con componentes de interfaz del lado del servidor.</p>
<p><b>Nombre:</b></p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p style="text-align: center; background-color: #e0f0ff;"><b>JSF (Servlet)</b></p> </div> <p>Figura 5. Clase JSF Servlet</p>	<p><b>Propósito:</b></p> <p>Interceptar las peticiones a las páginas JSF.</p> <p><b>Descripción:</b></p> <p>Es el controlador de JSF que intercepta las peticiones de las páginas clientes, asociándoles a estas páginas, clases java que recogen la información introducida y que disponen de métodos que responden a las acciones del usuario. Además, prepara el contexto JSF antes de enrutar a las páginas correspondientes e interviene en la construcción de la respuesta para generar la vista, luego de ser invocada una petición.</p>
<p><b>Nombre:</b></p>	<p><b>Propósito:</b></p>

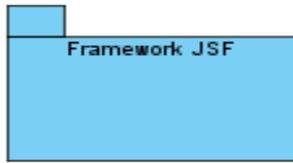


Figura 6. Paquete Framework JSF

Simplificar el desarrollo de las interfaces de usuario en aplicaciones java basadas en el patrón Modelo-Vista-Controlador.

**Descripción:**

La tecnología Java Server Faces es un framework de los componentes de la interfaz de usuario y es válido para todas aquellas aplicaciones web basadas en la tecnología JAVA, está basado en el patrón MVC. Forma parte de la especificación JEE 5.

### Capa de Negocio

**Nombre:**

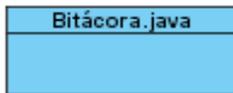


Figura 7. Clase Bitácora.java

**Propósito:**

Proveer las funcionalidades para realizar las auditorias del sistema.

**Descripción:**

Permite realizar las funciones para almacenar datos como la fecha, hora, usuario, contraseña, entre otros, del usuario que inicia y finaliza la sesión. Además, datos como el módulo y funcionalidad accedidos así como las que acciones ejecutadas.

**Nombre:**



Figura 8. Paquete Framework JasperReports

**Propósito:**

Añadir características de generación de reportes a aplicaciones java.

**Descripción:**

Es una librería de clases de Java de código abierto desarrollada para facilitar el agregar capacidades de reporte a las aplicaciones Java. Permite realizar reportes de código abierto que tiene como función el llevar documentos ricos en contenido a la pantalla, a la impresora, o a archivos PDF, HTML, XLS, CSV y XML.

**Nombre:**



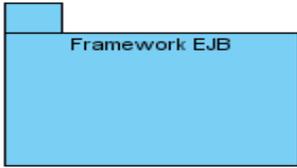
Figura 9. Clase CRExportar

**Propósito:**

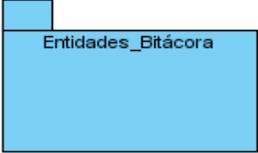
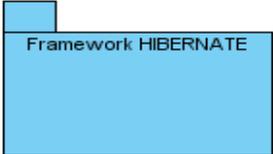
Proveer las funcionalidades para exportar a diferentes formatos.

**Descripción:**

Permite realizar las funciones para exportar a diferentes tipos de archivos mediante el uso de las librerías del Framework JasperReports.

<p><b>Nombre:</b></p>  <p>Figura 10. Paquete Framework EJB</p>	<p><b>Propósito:</b></p> <p>Encapsular la lógica de negocio que cumplimenta el propósito de la aplicación.</p> <p><b>Descripción:</b></p> <p>El Framework EJB está incluido en las capas de Negocio y Persistencia. Es una plataforma para construir aplicaciones de negocio portables, escalables, y reutilizables utilizando el lenguaje de programación Java. El objetivo de Enterprise JavaBeans (EJB) 3.0 es simplificar el desarrollo de aplicaciones Java y estandarizar el API de persistencia para la plataforma Java. Forma parte de la especificación JEE 5.</p>
<b>Capa de Datos</b>	
<p><b>Nombre:</b></p>  <p>Figura 11. Clase EntityManager</p>	<p><b>Propósito:</b></p> <p>Gestionar las entidades proveyendo servicios de persistencia.</p> <p><b>Descripción:</b></p> <p>Permite realizar las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) que impliquen entidades.</p>
<p><b>Nombre:</b></p>  <p>Figura 12. Clase EntityQuery</p>	<p><b>Propósito:</b></p> <p>Agregar consultas que pueden aplicarse a las entidades del modelo.</p> <p><b>Descripción:</b></p> <p>Permite encontrar objetos persistentes manejando cierto criterio de búsqueda. Permite realizar peticiones a la base de datos y controla cómo se ejecuta dicha petición. Se utiliza para enlazar los parámetros de la petición, limitar el número de resultados devueltos por la petición y para ejecutar dicha petición.</p>
<p><b>Nombre:</b></p> 	<p><b>Propósito:</b></p> <p>Agrupar las operaciones sobre datos persistentes en una unidad transaccional.</p> <p><b>Descripción:</b></p>

## CAPÍTULO 3. DISEÑO DEL SISTEMA.

<p>Figura 13. Clase EntityTransaction</p>	<p>Permite realizar operaciones sobre datos persistentes de manera que agrupados formen una unidad de trabajo transaccional, en el que todo el grupo sincroniza su estado de persistencia en la base de datos o todos fallan en el intento, en caso de fallo, la base de datos quedará con su estado original. Maneja el concepto de todos o ninguno para mantener la integridad de los datos.</p>
<p><b>Nombre:</b></p>	<p><b>Propósito:</b></p>
<div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px; margin-bottom: 10px;">  <p>Entidades_Bitácora</p> </div> <p>Figura 14. Clase Entidades Bitácora</p>	<p>Agrupar las entidades que contienen la información de las auditorías del sistema.</p> <p><b>Descripción:</b></p> <p>Contiene el conjunto de entidades que poseen la información de los usuarios y sus trazas en cuanto a sesión utilizada, módulos accedidos, funcionalidades permitidas, acciones realizadas y atributos modificados. Son utilizadas por la clase Bitácora.java para realizar las auditorías del sistema.</p>
<p><b>Nombre:</b></p>	<p><b>Propósito:</b></p>
<div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px; margin-bottom: 10px;">  <p>Framework HIBERNATE</p> </div> <p>Figura 15. Paquete Framework HIBERNATE</p>	<p>Proveer el mapeo objeto/relacional con la base de datos.</p> <p><b>Descripción:</b></p> <p>Conjunto de clases agrupadas en componentes que constituyen una herramienta de Mapeo objeto/relacional ó ORM de código abierto (Object Relational Mapping) y un generador de sentencias SQL. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada permite generar Bases de Datos en cualquiera de los entornos soportados: Oracle, PostgreSQL, DB2, MySQL, entre otras.</p>
<p><b>Nombre:</b></p>	<p><b>Propósito:</b></p>

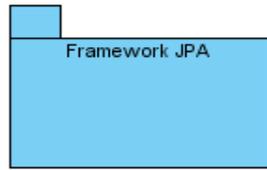


Figura 16. Paquete Framework JPA

Unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional.

**Descripción:**

Conjunto de clase agrupadas en componentes que constituyen la API de persistencia desarrollada para la plataforma Java EE e incluida en el estándar EJB 3.0 como parte de JSR 220, aunque su uso no se limita a los componentes software EJB. Permite unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos.

**Nombre:**

**Propósito:**

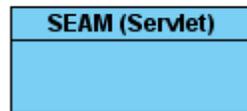


Figura 17. Clase SEAM Servlet

Proveer la interacción de la capa de presentación con la de negocio.

**Descripción:**

No se encuentra en ninguna capa en específico, sino que se encuentra ubicado entre la capa de Presentación y la de Negocio. Es el controlador de SEAM que capta las peticiones derivadas de la interacción del usuario después de interactuar con el Servlet de JSF. Enrutar las peticiones hacia los Beans que posibilitarán darle respuesta a la petición solicitada. Interviene en la integración de las capas de presentación y negocio.

### 3.3.1.1 Realizar Encuesta. Descripción.

#### Capa de Presentación

**Nombre:**

**Propósito:**



Figura 18. Clase agregar.siaps

Proveer la interacción con el usuario.

**Descripción:**

La clase agregar.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permitirá insertar todos los datos

## CAPÍTULO 3. DISEÑO DEL SISTEMA.

	<p>necesarios para proceder con la encuesta al paciente y que todo quede correctamente registrado. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.</p>
<p><b>Nombre:</b></p>	<p><b>Propósito:</b></p>
<p></p> <p>Figura 19. frmsiaps</p>	<p>Enviar los datos a las páginas servidoras.</p> <p><b>Descripción:</b></p> <p>La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para insertar todos los datos necesarios para proceder con la encuesta y que todo quede correctamente registrado. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.</p>
<p><b>Nombre:</b></p>	<p><b>Propósito:</b></p>
<p></p> <p>Figura 20. Clase listado.siaps</p>	<p>Proveer la interacción con el usuario.</p> <p><b>Descripción:</b></p> <p>La clase Listado.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permitirá buscar cualquier encuesta realizada previamente según la fecha de notificación, número del brote o algún centro en específico. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.</p>
<p><b>Nombre:</b></p>	<p><b>Propósito:</b></p>
<p></p> <p>Figura 21. frmsiaps</p>	<p>Enviar los datos a las páginas servidoras.</p> <p><b>Descripción:</b></p> <p>La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para listar todos los datos</p>

## CAPÍTULO 3. DISEÑO DEL SISTEMA.

	<p>necesarios de la Encuesta. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.</p>
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 22. Clase editar.siaaps</p>	<p>Proveer la interacción con el usuario.</p> <p><b>Descripción:</b></p> <p>La clase editar.siaaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permitirá acceder a modificar algunos campos de la encuesta llenados con anterioridad. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.</p>
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 23. frmsiaaps</p>	<p>Enviar los datos a las páginas servidoras.</p> <p><b>Descripción:</b></p> <p>La clase frmsiaaps contiene una colección de elementos de entrada que están contenidos en la página cliente para modificar los datos necesarios de la encuesta realizada con anterioridad. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.</p>
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 24. Clase detalles.siaaps</p>	<p>Proveer la interacción con el usuario.</p> <p><b>Descripción:</b></p> <p>La clase detalles.siaaps es una página web que se ejecuta del lado del cliente sobre un navegador. La cual permitirá ver los detalles de la encuesta previamente realizada, se mostrará una serie de datos que no se muestran en la encuesta. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones</p>

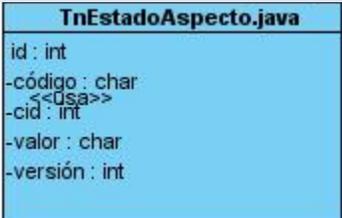
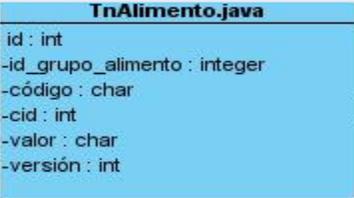
## CAPÍTULO 3. DISEÑO DEL SISTEMA.

	innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 25. frmsiaps</p>	<p>Enviar los datos a las páginas servidoras.</p> <p><b>Descripción:</b></p> <p>La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para mostrar todos los datos necesarios de la encuesta. Sus atributos son los elementos de entrada del formulario (inputboxes, textareas, checkboxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.</p>
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 26. Clase exportar.siaps</p>	<p>Proveer la interacción con el usuario.</p> <p><b>Descripción:</b></p> <p>La clase exportar.siaps es una página web que se ejecuta del lado del cliente sobre un navegador. Permitirá exportar los datos de la encuesta, da la opción de escoger la cantidad de páginas que deseas exportar y además te brinda la opción de exportar tanto en formato WORD, como en PDF. Posee un conjunto de validaciones en JavaScript que permite no realizar peticiones innecesarias y por lo tanto se incrementa su usabilidad. Utiliza diferentes librerías basadas en el Framework JSF.</p>
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 27. frmsiaps</p>	<p>La clase frmsiaps contiene una colección de elementos de entrada que están contenidos en la página cliente para exportar todos los datos de la encuesta. Sus atributos son los elementos de entrada del formulario (input boxes, text areas, check boxes, entre otros). No tienen operaciones, el método para el paso de los parámetros es \$_POST.</p>
<b>Capa de Negocio</b>	

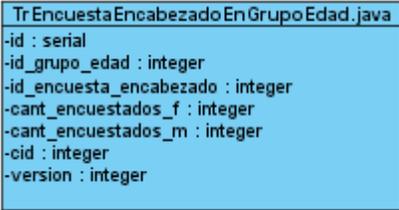
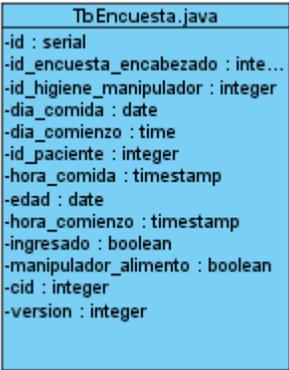
## CAPÍTULO 3. DISEÑO DEL SISTEMA.

<b>Nombre:</b>	<b>Propósito:</b>
 CCEncuestaETA.java Figura 28. Clase CCEncuestaETA.java	Proveer una respuesta a las peticiones realizadas en la vista. <b>Descripción:</b> La clase CCEncuestaETA.java es una clase que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de insertar los datos sobre la página cliente correspondiente. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.
<b>Nombre:</b>	<b>Propósito:</b>
 TbEncuestaETAList.java Figura 29. Clase TbEncuestaETAList.java	Proveer una respuesta a las peticiones realizadas en la vista. <b>Descripción:</b> La clase TbEncuestaETAList.java es una clase que se genera de forma automática el realizar la ingeniería inversa y que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de listar los datos solicitados sobre la página cliente correspondiente. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.
<b>Nombre:</b>	<b>Propósito:</b>
 CLEncuestaETAs.java Figura 30. CLEncuestaETA.java	Proveer una respuesta a las peticiones realizadas en la vista. <b>Descripción:</b> La clase CLEncuestaETA.java es una clase que se ejecuta del lado del servidor. Hereda todas las funcionalidades de la clase autogenerada TbEncuestaETAList.java y permite guardar los cambios originales de la misma evitando que los mismos se pierdan al realizar la ingeniería inversa. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través

	del Framework SEAM.
<b>Nombre:</b>	<b>Propósito:</b>
 CMEncuestaETA.java Figura 31. Clase CMEncuestaETA.java	Proveer una respuesta a las peticiones realizadas en la vista. <b>Descripción:</b> La clase CMEncuestaETA.java es una clase que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de modificar los datos de cualquier encuesta realizada con anterioridad que se solicite sobre la página cliente correspondiente. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.
<b>Nombre:</b>	<b>Propósito:</b>
 CDEncuestaETA.java Figura 32. Clase CDEncuestaETA.java	Proveer una respuesta a las peticiones realizadas en la vista. <b>Descripción:</b> La clase CDEncuestaETA.java es una clase que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de mostrar los datos de la encuesta seleccionada con anterioridad que es solicitado desde la página cliente correspondiente. Hace uno del Framework EJB que encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.
<b>Nombre:</b>	<b>Propósito:</b>
 CREncuestaETA.java Figura 33. Clase CREncuestaETA.java	<b>Descripción:</b> La clase CREncuestaETA.java es una clase que se ejecuta del lado del servidor. Permite darle respuesta a las peticiones que se desencadenan en la vista a través de los métodos que contienen. Se encarga de gestionar la acción de exportar solicitada sobre la página cliente correspondiente. Hace uno del Framework EJB que

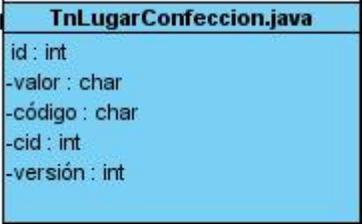
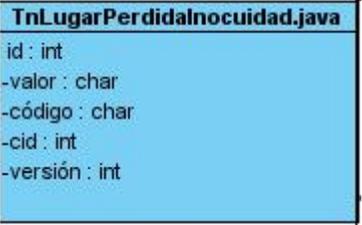
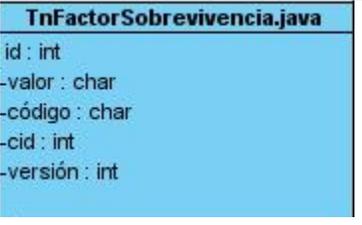
	encapsula la lógica de negocio, integrándose con la vista a través del Framework SEAM.
<b>Capa de Datos</b>	
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 34. Clase TnEstadoAspecto.java</p>	<p>Proveer el mapeo con la base de datos.</p> <p><b>Descripción:</b></p> <p>La clase TnEstadoAspecto.java es una clase que se ejecuta del lado del servidor. En ella se almacenan el aspecto del local donde se produjo el brote. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p>
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 35. Clase TnGrupoAlimento.java</p>	<p>Proveer el mapeo con la base de datos.</p> <p><b>Descripción:</b></p> <p>La clase TnGrupoAlimento.java es una clase que se ejecuta del lado del servidor. En ella se almacenan el nombre de los grupos de alimentos. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p>
<b>Nombre:</b>	<b>Propósito:</b>
 <p>Figura 36. Clase TnAlimento.java</p>	<p><b>Descripción:</b></p> <p>La clase TnAlimento.java es una clase que se ejecuta del lado del servidor. En ella se almacena el nombre de los alimento por grupo. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las</p>

## CAPÍTULO 3. DISEÑO DEL SISTEMA.

	páginas clientes. Hace uso del Framework Hibernate y JPA.
<b>Nombre:</b>	<b>Propósito:</b>
 <pre> TrEncuestaEncabezadoEnGrupoEdad.java -id : serial -id_grupo_edad : integer -id_encuesta_encabezado : integer -cant_encuestados_f : integer -cant_encuestados_m : integer -cid : integer -version : integer         </pre>	<b>Descripción:</b> La clase TrEncuestaEncabezadoEnGrupoEdad.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de los grupos de edades y la cantidad de encuestados femeninos y masculino. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.
Figura 37. Clase TrEncuestaEncabezadoEnGrupoEdad.java	
<b>Nombre:</b>	<b>Propósito:</b>
 <pre> TnGrupoEdad.java id : int -valor : char -código : char -cid : int -versión : int         </pre>	<b>Descripción:</b> La clase TnGrupoEdad.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los grupos de edades existentes. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.
Figura 38. Clase TnGrupoEdad.java	
<b>Nombre:</b>	<b>Propósito:</b>
 <pre> TbEncuesta.java -id : serial -id_encuesta_encabezado : inte... -id_higiene_manipulador : integer -dia_comida : date -dia_comienzo : time -id_paciente : integer -hora_comida : timestamp -edad : date -hora_comienzo : timestamp -ingresado : boolean -manipulador_alimento : boolean -cid : integer -version : integer         </pre>	Proveer el mapeo con la base de datos.  <b>Descripción:</b> La clase TbEncuesta.java es una clase que se ejecuta del lado del servidor. En ella se almacenan parte de los datos que se recogen en la encuesta: higiene del manipulador, hora de comienzo de los síntomas, hora de comida entre otros. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.
Figura 39. Clase TbEncuesta.java	

<p><b>Nombre:</b></p> <div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px; margin-bottom: 5px;"> <p style="text-align: center; margin: 0;"><b>TnSintomaIntoxicacion.java</b></p> <pre style="margin: 0;">id : int -código : char -cid : int -valor : char -versión : int</pre> </div> <p>Figura 40. Clase TnSintomasIntoxicacion.java</p>	<p><b>Propósito:</b></p> <p>Proveer el mapeo con la base de datos.</p> <p><b>Descripción:</b></p> <p>La clase TnSintomasIntoxicacion.java es una clase que se ejecuta del lado del servidor. En ella se las descripciones o nombres referentes a los síntomas de intoxicación. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p>
<p><b>Nombre:</b></p> <div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px; margin-bottom: 5px;"> <p style="text-align: center; margin: 0;"><b>TnEtiologia.java</b></p> <pre style="margin: 0;">id : int -valor : char -código : char -cid : int -versión : int</pre> </div> <p>Figura 41. Clase TnEtiologia.java</p>	<p><b>Propósito:</b></p> <p>Proveer el mapeo con la base de datos.</p> <p><b>Descripción:</b></p> <p>La clase TnEtiologia.java es una clase que se ejecuta del lado del servidor. En ella se almacenan las descripciones o nombres referentes a los factores etiológicos. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p>
<p><b>Nombre:</b></p> <div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px; margin-bottom: 5px;"> <p style="text-align: center; margin: 0;"><b>TrSintomaIntoxicacionEnEncuesta.java</b></p> <pre style="margin: 0;">-id_encuesta : integer -id_sintoma_intoxicacion</pre> </div> <p>Figura 42. Clase TrSintomaIntoxicacionEnEncuesta. java</p>	<p><b>Propósito:</b></p> <p>Proveer el mapeo con la base de datos.</p> <p><b>Descripción:</b></p> <p>La clase TrSintomaIntoxicacionEnEncuesta.java es una clase que se ejecuta del lado del servidor. En ella se almacenan la relación entre los síntomas de intoxicación existentes y los que presentan los pacientes encuestados. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del</p>

## CAPÍTULO 3. DISEÑO DEL SISTEMA.

	Framework Hibernate y JPA.
<b>Nombre:</b>	<b>Propósito:</b>
 <pre>TnLugarConfeccion.java id : int -valor : char -código : char -cid : int -versión : int</pre>	Proveer el mapeo con la base de datos. <b>Descripción:</b> La clase TnLugarConfeccion.java es una clase que se ejecuta del lado del servidor. En ella se almacenan las descripciones o nombres referentes a los lugares de confección del alimento. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.
Figura 43. Clase TnLugarConfeccion.java	
<b>Nombre:</b>	<b>Propósito:</b>
 <pre>TnLugarPerdidadInocuidad.java id : int -valor : char -código : char -cid : int -versión : int</pre>	Proveer el mapeo con la base de datos. <b>Descripción:</b> La clase TnLugarPerdidadInocuidad.java es una clase que se ejecuta del lado del servidor. En ella se almacenan las descripciones o nombres referentes a los lugares donde el alimento perdió la inocuidad. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.
Figura 44. Clase TnLugarPerdidadInocuidad.java	
<b>Nombre:</b>	<b>Propósito:</b>
 <pre>TnFactorSobrevivencia.java id : int -valor : char -código : char -cid : int -versión : int</pre>	Proveer el mapeo con la base de datos. <b>Descripción:</b> La clase TnFactorSobrevivencia.java es una clase que se ejecuta del lado del servidor. En ella se almacenan las descripciones o nombres referentes a los factores que contribuyeron a la sobrevivencia de las bacterias que dieron origen a la enfermedad.
Figura 45. Clase	

## CAPÍTULO 3. DISEÑO DEL SISTEMA.

<p>TnFactorSobrevivencia.java</p>	<p>Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p>
<p><b>Nombre:</b></p>	<p><b>Propósito:</b></p>
<div data-bbox="175 554 464 730" style="border: 1px solid black; background-color: #e0f0ff; padding: 5px;"> <p><b>TnFactorMultiplicacion.java</b>  id : int  -código : char  -valor : char  -cid : int  -versión : int</p> </div> <p>Figura 46. Clase TnFactorMultiplicacion.java</p>	<p>Proveer el mapeo con la base de datos.</p> <p><b>Descripción:</b></p> <p>La clase TnFactorMultiplicacion.java es una clase que se ejecuta del lado del servidor En ella se almacenan las descripciones o nombres referentes a los factores que contribuyeron a la multiplicación de las bacterias que dieron origen a la enfermedad. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p>
<p><b>Nombre:</b></p>	<p><b>Propósito:</b></p>
<div data-bbox="162 1098 519 1318" style="border: 1px solid black; background-color: #e0f0ff; padding: 5px;"> <p><b>TnFactorContaminacion.java</b>  id : int  -código : char  -valor : char  -cid : int  -versión : int</p> </div> <p>Figura 47. Clase TnFactorContaminacion.java</p>	<p>Proveer el mapeo con la base de datos.</p> <p><b>Descripción:</b></p> <p>La clase TnLugarConfeccion.java es una clase que se ejecuta del lado del servidor. En ella se almacenan las descripciones o nombres referentes a los factores que contribuyeron a la contaminación de las de los alimentos. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p>
<p><b>Nombre:</b></p>	<p><b>Propósito:</b></p>

<div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px; margin-bottom: 5px;"> <b>TnGrupoPorEspecie.java</b>                  id : int                  -código : char                  -valor : char                  -cid : int                  -versión : int             </div> <p>Figura 48. Clase TnGrupoPorEspecie.java</p>	<p>Proveer el mapeo con la base de datos.</p> <p><b>Descripción:</b></p> <p>La clase TnGrupoPorEspecie.java es una clase que se ejecuta del lado del servidor. En ella se almacenan las descripciones o nombres específicos de las especies que forman parte del grupo 10 de los alimentos. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p>
<p><b>Nombre:</b></p>	<p><b>Propósito:</b></p>
<div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px; margin-bottom: 5px;"> <b>TnSintomaIntoxicacion.java</b>                  id : int                  -código : char                  -cid : int                  -valor : char                  -versión : int             </div> <p>Figura 49. Clase TnSintomaIntoxicacion.java</p>	<p>Proveer el mapeo con la base de datos</p> <p><b>Descripción:</b></p> <p>La clase TnSintomaIntoxicacion.java es una clase que se ejecuta del lado del servidor. En ella se almacenan las descripciones o nombres referentes a los síntomas que puede presentar un paciente cuando sufre de una ETA. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.</p>
<p><b>Nombre:</b></p>	<p><b>Propósito:</b></p>
<div style="border: 1px solid black; background-color: #e0f0ff; padding: 5px; margin-bottom: 5px;"> <b>TnSexo.java</b>                  -id_sexo : int                  -version : int                  -cid : int                  -sexo : string             </div> <p>Figura 50. Clase TnSexo.java</p>	<p>Proveer el mapeo con la base de datos.</p> <p><b>Descripción:</b></p> <p>La clase TnSexo.java es una clase que se ejecuta del lado del servidor. En ella se almacena el tipo de sexo: femenino o masculino. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework</p>

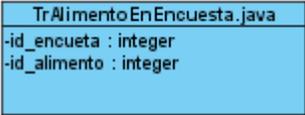
	Hibernate y JPA.
<b>Nombre:</b>	<b>Propósito:</b>
 <p>TrAlimentoEnEncuesta.java -id_encuesta : integer -id_alimento : integer</p>	Proveer el mapeo con la base de datos. <b>Descripción:</b> La clase TrAlimentoEnEncuesta.java es una clase que se ejecuta del lado del servidor. En ella se almacena la relación que existe entre los alimentos existentes y los consumidos por los pacientes encuestados. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.
<b>Nombre:</b>	<b>Propósito:</b>

Figura 51. Clase

TrAlimentoEnEncuesta.java

TbEncuestaEncabezado.java
<pre> -id : serial -no_brote : int -id_etiologia : int -id_paciente : int -id_centro_de_ocurrencia : int -id_centro_atencion_casos : int -procedencia_informacion : varchar -fecha_notificacion : date -lugar_brote : varchar -lugares_comieron_antes_enfermar : varchar -especie_por_grupo : varchar -conclusiones : varchar -cid : integer -version : integer -nombre_cientifico : varchar -ingrediente_contaminante : varchar -id_lugar_confeccion : integer -id_perdida_inocuidad : integer -id_factor_contaminacion : integer -id_factor_sobrevivencia : integer -id_factor_multiplicacion : integer -periodo_incubacion : timestamp -confirmado : boolean -muestras_testigo : boolean -restos_alimentos : boolean -resultado : varchar -resultado_alimento : varchar -condiciones_higienicas : varchar -id_diligencia_inspeccion_sanitaria : integer -expuestos : integer -afectados : integer -ingresados : integer -fallecidos : integer -telefono_inha : varchar -periodo_elaboracion_consumo_inha : timest... -determinacion_sugiere_inha : varchar -id_tecnico_entrega_muestra_inha : integer -id_jefe_equipo_inha : integer                     </pre>

Figura 52. Clase  
TbEncuestaEncabezado.java

Proveer el mapeo con la base de datos.

**Descripción:**

La clase TbEncuestaEncabezado.java es una clase que se ejecuta del lado del servidor. En ella se almacenan los datos de la encuesta que se les realiza a los pacientes. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.

**Nombre:**

**Propósito:**

TbTasaAtaque.java
<pre> -id : serial -id_encabezado_encuesta : integer -id_alimento : integer -no_comieron_enf : integer -comieron_enf : integer -total_comieron : integer -comieron_ta : double -no_comieron_no_enf : integer -comieron_no_enf : integer -total_no_comieron : integer -no_comieron_ta : double -diferencia : double -riesgo_relativo : integer -version : integer -cid : integer                     </pre>

Proveer el mapeo con la base de datos.

**Descripción:**

La clase TbTasaAtaque.java es una clase que se ejecuta del lado del servidor. En ella se almacena la cantidad de pacientes que comieron él alimento y enfermaron así como lo que no comieron y enfermaron entre otros para luego realizar cálculos y obtener el alimento sospechoso. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para

## CAPÍTULO 3. DISEÑO DEL SISTEMA.

Figura 53. Clase TbTasaAtaque.java	darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.
<b>Nombre:</b>	<b>Propósito:</b>
	Proveer el mapeo con la base de datos. <b>Descripción:</b> La clase TbDireccion.java es una clase que se ejecuta del lado del servidor. En ella se almacena la dirección del centro laboral. Representa una tabla en el modelo de datos relacional y cada instancia de esta entidad corresponde a un registro en esa tabla. Es persistida por las clases servidoras para darle una respuesta a las páginas clientes. Hace uso del Framework Hibernate y JPA.
Figura 54. Clase TbDireccion.java	

Para obtener más información de las clases del diseño del proceso de ETA, remitirse al Expediente de Proyecto. [39]

En el capítulo se presentó la propuesta de solución al problema planteado. Se definieron los diagramas de clases del diseño, la descripción de los patrones utilizados así como la descripción de cada una de las clases del diseño con el objetivo de entender mejor el sistema, empleando para ello el Lenguaje unificado de Modelado y la herramienta case Visual Paradigm. Además, los artefactos obtenidos en el desarrollo del diseño propuesto constituyen la entrada fundamental para el desarrollo del Flujo de Trabajo de Implementación. A partir de este punto se puede comenzar a construir la aplicación, teniendo en cuenta los requerimientos y funcionalidades derivados de esta etapa.

### Capítulo 4. Implementación

Este capítulo constituye la continuidad del modelo de diseño. Fundamenta la necesidad de integración del Componente Web Enfermedades Transmitidas por Alimentos con otros sistemas. Se presenta el Diagrama de Despliegue de la solución propuesta y se definen también elementos clave para la seguridad, el tratamiento de errores y los estilos de código a utilizar en la aplicación.

#### 4.1 Propuesta de integración entre módulos

Actualmente casi ninguna aplicación de la salud se crea como un sistema aislado por lo que se contempla la integración del Componente Web Enfermedades Transmitidas por Alimentos con el módulo de Higiene y Epidemiología y este a su vez con el Sistema Integral para la Atención Primaria de Salud alas SIAPS como plataforma única para la gestión, procesamiento y transmisión de la información en el SNS. Este intercambio de información se efectuará entre las bases de datos.

##### 4.1.1 Módulo Medicina Familiar del Subsistema Web.

El Componente Web Enfermedades Transmitidas por Alimentos del SIAPS se relaciona con el Módulo de Medicina Familiar del cual toma los datos de los pacientes registrados a la hora de realizar la encuesta.

##### 4.1.2 Módulo Configuración del Subsistema Web.

El Módulo Configuración del Subsistema Web le permite al Componente Web Enfermedades Transmitidas por Alimentos obtener la información relacionada con los nomencladores, Clasificador Internacional de Enfermedades (por sus siglas CIE), Centros Laborales, para la ubicación del lugar donde se produjo el brote, así como el centro que se está haciendo cargo del estudio, además de proporcionar los datos del Personal de Salud y las entidades a las que pertenece.

#### 4.2 Propuesta de seguridad del módulo.

Por la importancia de las actividades del proceso de ETA, surge la necesidad de que estas sean realizadas con la calidad requerida y que el acceso a la información registrada solo sea posible para las personas que se encuentren acreditadas, todo esto trae consigo que se tengan en cuenta una serie de requisitos de seguridad de modo que no se pueda afectar el funcionamiento del sistema.

Para garantizar lo antes planteado se llevará a cabo un control entre los usuarios y sus contraseñas, permitiendo el acceso por tipo de usuario logrando así la visibilidad sólo a las áreas establecidas de

acorde a la función que realizan. Las contraseñas solo podrán ser cambiadas por el usuario o por el administrador del sistema.

Para lograr la fidelidad de los datos, todo el intercambio entre el sistema y otros sistemas que soliciten información desde cualquier entidad, se realizará de forma cifrada eliminando posibilidades de acceso o modificación de la misma.

### **4.3 Implementación.**

El desarrollo del flujo de trabajo de Implementación comienza con el resultado del diseño, definiéndose los nodos que se utilizarán, la ubicación en ellos de los componentes y la estructura de capas de la aplicación. Además de implementar las clases encontrados durante el diseño, estas clases se implementan como componentes de ficheros que contienen código fuente. Uno de los principales propósitos de este flujo de trabajo consiste en desarrollar la arquitectura y el sistema en su conjunto.

### **4.4 Diagrama de Despliegue.**

Un diagrama de Despliegue describe la distribución física del sistema, muestra cómo están distribuidos los componentes de software entre los distintos nodos. Los cuales se representan como máquinas físicas y procesadores, y la construcción interna puede ser representada por nodos o artefactos embebidos. Los nodos representan recursos de cómputos entre los cuales existen relaciones que representan el medio de comunicación. Permite comprender la correspondencia entre la arquitectura de software y la arquitectura de hardware.

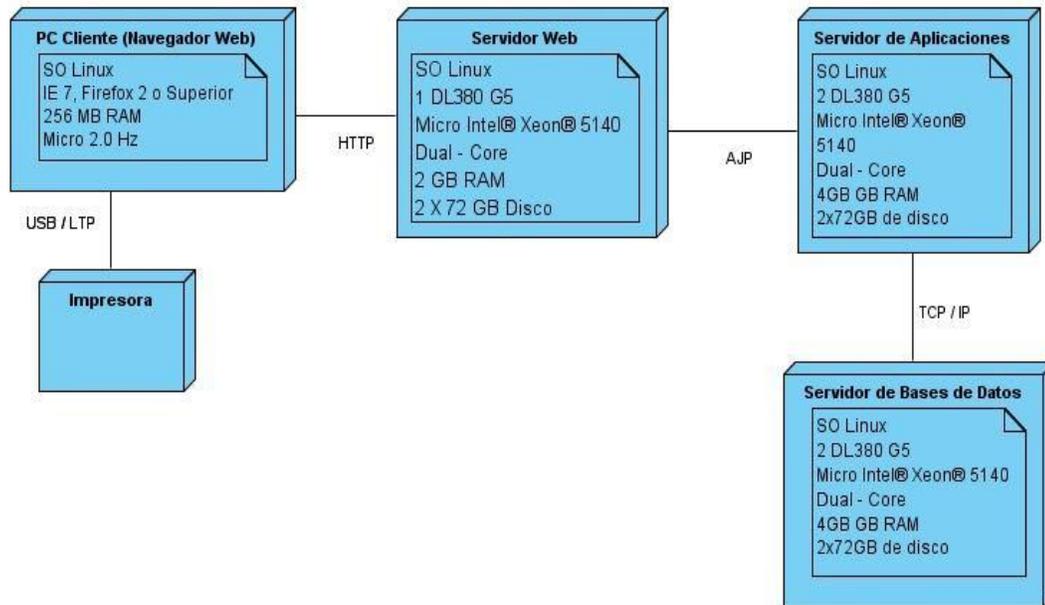


Figura 55. Diagrama de Despliegue.

## 4.5 Estándares de codificación y tratamiento de excepciones

### 4.5.1 Estándares de codificación

En la implementación de un sistema informático, mantener un estándar en la codificación es muy importante ya que por lo general ningún software mantiene toda su vida el autor original y el uso de estándares de codificación mejoran la lectura y comprensión del software. Por esta razón es necesario seguir la convención que estandariza la implementación del mismo. Teniendo en cuenta todo lo expuesto a continuación se plantean una serie de estrategias de codificación a utilizar para la implementación del Componente Web Enfermedades Transmitidas por Alimentos del Sistema Integral para la Atención Primaria de Salud alas SIAPS.

**Idioma:** se debe utilizar como idioma el español, las palabras no se acentuarán.

**Comentarios:** los ficheros fuente comienzan con un comentario en el que se lista el nombre de la clase, información de la versión, fecha, y copyright.

**Líneas:** las líneas en blanco mejoran la facilidad de lectura separando secciones de código que están lógicamente relacionadas. Se deben usar siempre dos líneas en blanco entre las secciones de un fichero fuente y entre las definiciones de clases e interfaces. También se debe usar siempre una línea en blanco

entre métodos, entre las variables locales de un método y su primera sentencia, antes de un comentario de bloque o de un comentario de una línea y entre las distintas secciones lógicas de un método para facilitar la lectura.

**Espacios en blanco:** debe usarse un espacio en blanco entre una palabra clave del lenguaje que va seguida por un paréntesis.

**Indentación y longitud de la línea:** Se deben emplear cuatro espacios como unidad de indentación. La construcción exacta de la indentación (espacios en blanco contra tabuladores) no se especifica. Los tabuladores deben ser exactamente cada 8 espacios. Se debe evitar las líneas de más de 80 caracteres, ya que no son manejadas bien por muchas terminales y herramientas.

**Variables, constantes, clases y métodos:** Los nombres de variables no deben empezar con los caracteres subguión "\_" o signo del dólar "\$", aunque ambos están permitidos por el lenguaje. Todas las variables e instancias de clase o método empezarán con minúscula. Las palabras internas que lo forman (si son compuestas) empiezan con su primera letra en mayúsculas.

Los nombres de las variables deben ser cortos pero con significado. La elección del nombre de una variable debe ser un mnemónico, designado para indicar a un observador casual su función. Los nombres de variables de un solo carácter se deben evitar, excepto para variables índices temporales.

Los nombres de las variables declaradas como constantes deben ir totalmente en mayúsculas separando las palabras con un subguión ("\_"). (Las constantes ANSI se deben evitar, para facilitar su depuración).

Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas. Mantener los nombres de las clases, simples y descriptivos. Usar palabras completas, evitar acrónimos y abreviaturas.

Los métodos deben ser verbos, cuando son compuestos tendrán la primera letra en minúscula, y la primera letra de las siguientes palabras que lo forma en mayúscula.

### 4.6 Tratamiento de excepciones.

En el componente web propuesto se manejan todas las habilidades para el tratamiento de excepciones que brinda la plataforma. Para cada fragmento de código donde se espere una situación anómala, se definen las excepciones correspondientes para luego ser tratadas evitando la interrupción del sistema. En las páginas clientes se cuenta con un conjunto de componentes denominados validadores, que permiten

## CAPÍTULO 4. IMPLEMENTACIÓN

---

establecer tipos de datos y formatos controlando que el envío de los activos al servidor sean los esperados.

También se emplean un conjunto de tipos de excepciones predefinidas por los marcos de trabajos que se utilizan en el sistema. Por ejemplo, el marco de trabajo Seam tiene un conjunto de excepciones predefinidas, que conjuntamente con la clase FacesMessages, permite tratar estas situaciones desde las clases controladoras correspondientes y mostrar mediante la clase antes mencionada, los resultados del tratamiento. Seam permite además mediante el fichero de configuración page.xml, todo un flujo de navegación basado en excepciones.

En este capítulo se presentaron los estándares de codificación y de diseño usados para lograr una aplicación con uniformidad, también se mostró el diagrama de despliegue donde se ilustran los nodos que serán usados para la implantación de la aplicación. Además, se trataron las posibles excepciones que el sistema pudiera presentar con el objetivo de obtener un sistema seguro y confiable.

### Conclusiones

- Las actividades del proceso de ETA llevadas a cabo en las UMHE demostraron la necesidad de desarrollar un sistema informático para la gestión de la información. A través de estas actividades se identificaron las funcionalidades previas a tener en cuenta para el desarrollo del sistema propuesto.
- Se implementaron funcionalidades que permiten procesar la encuesta relacionada a una ETA y generar el Anexo 1 en forma de reporte, mostrándose en este la curva epidémica asociada al caso consultado logrando disminuir el tiempo de gestión de la información.
- La aplicación del patrón de diseño MVC resultó conveniente pues permitió la organización de las funcionalidades en los tres componentes de diseño según sus responsabilidades, disminuyendo la complejidad de implementación.
- La adopción de las pautas de diseño de interfaz de usuario permite que las nuevas interfaces sean visualmente homogéneas a las existentes, obteniendo una aplicación visualmente uniforme.

### Recomendaciones

Se recomienda:

- Incluir las funcionalidades Mostrar Anexo 3, Llenar informe del INHA, Insertar acta de decomiso o arrojo, Llenar diligencia de ocupación de la muestra resultado de análisis y Llenar diligencia de inspección sanitaria para garantizar el total cumplimiento de las actividades que se realizan en el proceso de las Enfermedades Transmitidas por Alimento.
- Definir e implementar reportes estadísticos que contribuyan a obtener información más precisa y detallada.
- Incluir un sistema en las Áreas de Salud que automatice la gestión de la información de las enfermedades de declaración obligatoria, que llegan diariamente a los centros de Higiene y Epidemiología.

### Referencias Bibliográficas

1. Llanusa Ruiz, Susana Beatriz. Revista Cubana de Salud Pública. [En línea] [Citado el: 8 de Diciembre de 2010.] [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S0864-34662005000300008&lng=en&nrm=iso&ignore=.html](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-34662005000300008&lng=en&nrm=iso&ignore=.html)
2. **Santana, Manuel A.** Conceptos y definiciones a ser utilizados por los organismos comprometidos en la atención de emergencias. [En línea]  
[http://www.google.com.cu/url?sa=t&source=web&cd=2&ved=0CB0QFjAB&url=http%3A%2F%2Fwww.crid.or.cr%2Fdigitalizacion%2Fpdf%2Fspa%2Fdoc6742%2Fdoc6742-contenido.pdf&rct=i&q=definicion%20de%20atencion%20secundaria&ei=qTuKTZLBDKGS0QG0z5mODg&usq=AFQjCNHf\\_tljCJxDaz](http://www.google.com.cu/url?sa=t&source=web&cd=2&ved=0CB0QFjAB&url=http%3A%2F%2Fwww.crid.or.cr%2Fdigitalizacion%2Fpdf%2Fspa%2Fdoc6742%2Fdoc6742-contenido.pdf&rct=i&q=definicion%20de%20atencion%20secundaria&ei=qTuKTZLBDKGS0QG0z5mODg&usq=AFQjCNHf_tljCJxDaz).
3. Ídem a Referencia 2.
4. **Rojas Ochoa, Francisco.** La atención primaria de salud en Cuba. [En línea] 1984.  
[http://www.infodoctor.org/salek/Que\\_es\\_la-Atencion\\_Primaria.pdf](http://www.infodoctor.org/salek/Que_es_la-Atencion_Primaria.pdf).
5. **Puñales sosa, Osvaldo.** *La Inspección Sanitaria Estatal, herramienta fundamental en la Gestión de Inocuidad por un Turismo de Excelencia*. La Habana : s.n., 2005.
6. **Aguilera, Ximena.** Normas técnicas de vigilancia de enfermedades transmisibles. [En línea] [Citado el: 19 de octubre de 2010.] <http://epi.minsal.cl/epi/html/public/enftransmisibles.pdf>.
7. **Trincado Agudo, María Teresa.** Sistema Nacional de Salud. [En línea] 2006.  
<http://www.google.com.cu/url?sa=t&source=web&cd=7&ved=0CEUQFjAG&url=http%3A%2F%2Fwww.amenac.com%2Fpdf%2Ftrincado-agudo-02.pdf&rct=i&q=definicion%20del%20Sistema%20nacional%20de%20salud%20de%20cuba&ei=W D6KTcXhJ4Hf0gH7q53LAq&usq=AFQjCNGh3jDI2deQP-GVHpg4PDK>.
8. Ídem a Referencia 4.
9. [En línea] [Citado el: 25 de 10 de 2010.] <http://www.scielosp.org/pdf/rcsp/v35n1/spu05109.pdf>.
10. **Bonvini, Natalia.** Sistemas de Información y Gestión del Conocimiento en Inocuidad de los Alimentos. [En línea] [Citado el: 10 de 11 de 2010.] <http://www.mpps.gob.ve/Boletines/SVETA.pdf>.

## REFERENCIAS BIBLIOGRÁFICAS

---

11. Agentes bacterianos asociados a brotes de Enfermedades Transmitidas por Alimentos (ETA) aislados. [En línea] [Citado el: 26 de septiembre de 2010.]  
<http://www.veterinaria.org/redvet/n030310/031016.pdf>.
12. [En línea] [Citado el: 02 de 11 de 2010.]  
<http://www.msc.es/organización/sns/planCalidadSNS/syp01.htm>.
13. **Puñales Sosa, Osvaldo.** *Inocuidad e Higiene de los Alimentos.* La Habana : s.n., 2009.
14. Ídem a Referencia 13.
15. Ídem a Referencia 13.
16. [En línea] [Citado el: 19 de octubre de 2010.]  
<http://www.msc.es/organización/sns/planCalidadSNS/syp01.htm>.
17. **Lezcano, Elizabeth.** Guía de aplicación de buenas prácticas de manufactura. [En línea]  
<http://www.sagpya.gov.ar/alimento>.
18. Ídem a Referencia 17.
19. SVETA Enfermedades Transmitidas por Alimento. Sistema de Vigilancia Epidemiológica. [En línea] [Citado el: 25 de septiembre de 2010.] <http://www.mpps.gob.ve/Boletines/SVETA.pdf>.
20. Ídem a Referencia 19.
21. Ídem a Referencia 19.
22. Ídem a Referencia 19.
23. **Puñales Sosa, Osvaldo.** *La Inspección Sanitaria Estatal, herramienta fundamental en la Gestión de Inocuidad por un Turismo de Excelencia.* Ciudad de La Habana : s.n., 2005.
24. Patrón "Modelo-Vista-Controlador". [En línea] [Citado el: 6 de diciembre de 2010.]  
<http://www.proactiva-calidad.com/java/patrones/mvc.html>
25. Arquitectura Basada en Componentes. [En línea] [Citado el: 6 de diciembre de 2010.]  
<http://www.scribd.com/doc/14704374/Arquitectura-Basada-en-Componentes>.

## REFERENCIAS BIBLIOGRÁFICAS

---

26. Beneficios de la Tecnología JavaServer Faces. [En línea] [Citado el: 6 de diciembre de 2010.] <http://www.jcp.org/en/jsr/detail?id=152>.
27. **Sánchez Suárez, José Manuel.** *Introducción a RichFaces*. [En línea] [Citado el: 6 de diciembre de 2010.]. <http://frankseguel.blogspot.com/2008/05/introduccion-richfaces.html>
28. **Alonso Ramos, Juan.** *Introducción a Ajax4jsf*. [En línea] [Citado el: 6 de diciembre de 2010.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf>.
29. **Crombet Pérez, Yilianne Arias y López Hernández, Yurien.** *Desarrollo del Módulo Enfermería del Subsistema Web del Sistema Integral para la Atención Primaria de Salud alas SIAPS*. La Habana : s.n., 2010.
30. **José Alberto.** [En línea] [Citado el: 6 de diciembre de 2010.] [http://www.google.com/cu/#hl=es-419&source=hp&q=visual+paradigm+caracteristicas&oq=visual+paradigm+caracteristicas&aq=f&aqi=&aql=&gs\\_sm=e&gs\\_upl=664316l664961l0l4l4l0l0l0l0l106l243l2.1&fp=6efb24ce67212e1f&biw=1024&bih=639](http://www.google.com/cu/#hl=es-419&source=hp&q=visual+paradigm+caracteristicas&oq=visual+paradigm+caracteristicas&aq=f&aqi=&aql=&gs_sm=e&gs_upl=664316l664961l0l4l4l0l0l0l0l106l243l2.1&fp=6efb24ce67212e1f&biw=1024&bih=639)
31. Programas.com. [En línea] [Citado el: 28 de enero de 2011.] <http://toad-data-modeler.programas-gratis.net>.
32. Ídem a Referencia 29.
33. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. Addison Wesley. s.l. : Madrid, Pearson Educación, 2000. 458.
34. **Sánchez Fornaris, Maite y Alcantara Rabí, Elvia Dayanis y Hernández Luque, Eyllin.** Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica. [En línea] 15 de Febrero de 2011. [http://vinculando.org/articulos/sociedad\\_america\\_latina/propuesta\\_guia\\_de\\_medidas\\_para\\_evaluacion\\_sistemas\\_informacion.html](http://vinculando.org/articulos/sociedad_america_latina/propuesta_guia_de_medidas_para_evaluacion_sistemas_informacion.html).
35. **Hernández Sablón, Ariuska.** Repositorio de documentos para el Departamento de Atención Primaria de Salud del Centro de Infomática Médica. [En línea] 2011.

## REFERENCIAS BIBLIOGRÁFICAS

---

[https://repositorio.cesim.prod.uci.cu/svn/aps/higieneE/WEB/EXPEDIENTE\\_DE\\_PROYECTO/INGENIERIA/ARQUITECTURA\\_Y\\_DISENO](https://repositorio.cesim.prod.uci.cu/svn/aps/higieneE/WEB/EXPEDIENTE_DE_PROYECTO/INGENIERIA/ARQUITECTURA_Y_DISENO)



12. **Bonvini, Natalia.** *Sistemas de Información y Gestión del Conocimiento en Inocuidad de los Alimentos.* [En línea] [Citado el: 10 de 11 de 2010.] <http://www.mpps.gob.ve/Boletines/SVETA.pdf>.
13. **Castro Domínguez, Arnaldo.** *Enfermedades transmitidas por alimentos y su prevención.* La Habana : s.n.
14. **Castro Domínguez, Arnaldo.** *Guía para el estudio de las enfermedades transmitidas por alimentos.* La Habana : s.n.
15. **Crombet Pérez, Yilianne Arias y López Hernández, Yurien.** *Desarrollo del Módulo Enfermería del Subsistema Web del Sistema Integral para la Atención Primaria de Salud alas SIAPS.* La Habana : s.n., 2010.
16. *Curso Universitario Sistema de Información en los Sistemas de Salud (10 x 10 de la AMIA).* Argentina : s.n., 2009.
17. *Desarrollo de Software basado en Componente.* [En línea] [Citado el: 6 de diciembre de 2010.] <http://www.lambdasi.com.ar/textocomp.asp?id=9>.
18. **Ford, Aníbal y Chicco , Ivanna.** *Una navegación incierta. Mrcosur en internet.* [En línea] 1998. <http://www.unesco.org/most>.
19. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* Addison Wesley. s.l. : Madrid, Pearson Educación, 2000. 458.
20. **José Alberto.** [En línea] [Citado el: 6 de diciembre de 2010.] [http://www.google.com/cu/#hl=es-419&source=hp&q=visual+paradigm+caracteristicas&oq=visual+paradigm+caracteristicas&aq=f&aqi=&aql=&gs\\_sm=e&gs\\_upl=664316l664961l0l4l4l0l0l0l0l106l243l2.1&fp=6efb24ce67212e1f&biw=1024&bih=639](http://www.google.com/cu/#hl=es-419&source=hp&q=visual+paradigm+caracteristicas&oq=visual+paradigm+caracteristicas&aq=f&aqi=&aql=&gs_sm=e&gs_upl=664316l664961l0l4l4l0l0l0l0l106l243l2.1&fp=6efb24ce67212e1f&biw=1024&bih=639)
21. Junta de Andalucía. *RichFaces.* [En línea] 2008. [Citado el: 14 de diciembre de 2010.] <http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/RichFaces>
22. **Lezcano, Elizabeth.** *Guía de aplicación de buenas prácticas de manufactura.* [En línea] <http://www.sagpya.gov.ar/alimento>.

23. **Machin Navas, Mirialys.** *El uso de las TIC para el aprendizaje de la Programación.* [En línea] [Citado el: 8 de Diciembre de 2010.]  
<http://www.monografias.com/trabajos55/tic-en-programacion/tic-en-programacion2.shtml>.
24. **Marín Díaz, Miguel E.** *Capacitación en el área de la Informática en Salud. Revista cubana de Informática Médica.* [Disponible en:  
[http://www.cecam.sld.cu/pages/rcim/revista\\_10/articulos\\_hm/capacitacion.htm](http://www.cecam.sld.cu/pages/rcim/revista_10/articulos_hm/capacitacion.htm)
25. **Martínez Rodríguez, Jorge.** *Las TIC para el desarrollo. Una aplicación en la salud cubana.* [En línea]  
<http://www.eclac.cl/socinfo/noticias/paginas/6/35876/Cuba.pdf>.
26. **Programas.com.** [En línea] [Citado el: 28 de enero de 2011.] <http://toad-data-modeler.programas-gratis.net>.
27. **Puñales Sosa, Osvaldo.** *Inocuidad e Higiene de los Alimentos.* La Habana : s.n., 2009.
28. **Puñales Sosa, Osvaldo.** *La Inspección Sanitaria Estatal, herramienta fundamental en la Gestión de Inocuidad por un Turismo de Excelencia.* Ciudad de La Habana : s.n., 2005.
29. **Rojas Ochoa, Francisco.** *La atención primaria de salud en Cuba.* [En línea] 1984.  
[http://www.infodoctor.org/salek/Que\\_es\\_la-Atencion Primaria.pdf](http://www.infodoctor.org/salek/Que_es_la-Atencion Primaria.pdf).
30. **Sánchez Fornaris, Maite y Alcantara Rabí, Elvia Dayanis y Hernández Luque, Eyllin.** *Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica.* [En línea] 15 de Febrero de 2011.  
[http://vinculando.org/articulos/sociedad\\_america\\_latina/propuesta\\_guia\\_de\\_medidas\\_para\\_evaluacion\\_sistemas\\_informacion.html](http://vinculando.org/articulos/sociedad_america_latina/propuesta_guia_de_medidas_para_evaluacion_sistemas_informacion.html).
31. **Sánchez Suárez, José Manuel.** *Introducción a RichFaces.*  
<http://www.versionero.com/noticia/210/visual-paradigm-for-uml>.
32. **Santana, Manuel A.** *Conceptos y definiciones a ser utilizados por los organismos comprometidos en la atención de emergencias.* [En línea]  
[http://www.google.com/cu/url?sa=t&source=web&cd=2&ved=0CB0QFjAB&url=http%3A%2F%2Fwww.crid.or.cr%2Fdigitalizacion%2Fpdf%2Fspa%2Fdoc6742%2Fdoc6742-contenido.pdf&rct=j&q=definicion%20de%20atencion%20secundaria&ei=qTuKTZLBDKGS0QG0z5mODg&usq=AFQjCNHf\\_tljCJxDaz](http://www.google.com/cu/url?sa=t&source=web&cd=2&ved=0CB0QFjAB&url=http%3A%2F%2Fwww.crid.or.cr%2Fdigitalizacion%2Fpdf%2Fspa%2Fdoc6742%2Fdoc6742-contenido.pdf&rct=j&q=definicion%20de%20atencion%20secundaria&ei=qTuKTZLBDKGS0QG0z5mODg&usq=AFQjCNHf_tljCJxDaz).

33. *SVETA Enfermedades Transmitidas por Alimento. Sistema de Vigilancia Epidemiologica.* [En línea] [Citado el: 25 de septiembre de 2010.] <http://www.mpps.gob.ve/Boletines/SVETA.pdf>.
34. **Trejo Delbarne, Raúl.** *Desafíos y divergencias de la sociedad de la información en América Latina.* [En línea] <http://www.elbardelauni.com>.
35. **Trincado Agudo, María Teresa.** *Sistema Nacional de Salud.* [En línea] 2006.  
<http://www.google.com.cu/url?sa=t&source=web&cd=7&ved=0CEUQFjAG&url=http%3A%2F%2Fwww.amenac.com%2Fpdf%2Ftrincado-agudo-2.pdf&rct=j&q=definicion%20del%20Sistema%20nacional%20de%20salud%20de%20cuba&ei=WD6KTcXhJ4Hf0gH7g53LAq&usq=AFQjCNGh3jDI2deQP-GVHpg4PDK>.
36. *Tutorial de JavaServer Faces.* [En línea] [Citado el: 6 de diciembre de 2010.]  
<http://www.jcp.org/en/jsr/detail?id=127>.

### Glosario de términos.

**Agente etiológico:** entidad biológica, física o química capaz de causar enfermedad.

**Brote:** aparición de dos o más casos de una enfermedad transmisible, con nexos epidemiológicos o vinculados a una fuente común.

**Caso confirmado:** persona en quién se aisló o identificó un agente etiológico a través de evidencias clínicas, epidemiológicas o de laboratorio, siguiendo los criterios definidos para cada enfermedad específica.

**Caso-control-estudios:** tipo de estudio epidemiológico en el que se busca inferir una asociación entre un determinado factor de riesgo y la ocurrencia de una enfermedad, en grupos seleccionados, a partir de la comparación de la presencia del factor de riesgo en grupos de enfermos y no enfermos.

**Componente:** Parte física y reemplazable de un sistema que se ajusta a, y proporciona la realización de, un conjunto de interfaces.

**Periodo de incubación:** intervalo entre la exposición eficaz del huésped susceptible o sus productos tóxicos al agente biológico y el inicio de los signos y síntomas clínicos de la enfermedad en ese huésped

**Subsistema:** Agrupación de elementos, de los que algunos constituyen una especificación del comportamiento ofrecido por los elementos contenidos.

**Tasa de ataque:** es un tipo especial de tasa de incidencia corresponde al número de personas que presentan una enfermedad relacionado con el número de personas expuestas al riesgo de sufrirla, en un período limitado de tiempo y en condiciones especiales, como una epidemia

**Unidad de Salud:** Centro de trabajo que pertenece al Ministerio de Salud Pública (MINSAP).





Anexo 2. Anexo No. 1



Unidad Municipal de Higiene y Epidemiología  
Órgano de Poder Popular  
Municipio Playa



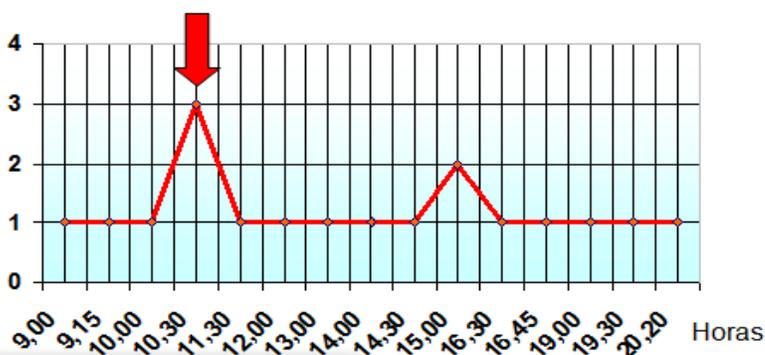
**ANEXO No. 1**

Brote: \_\_\_\_\_

1. Provincia : Ciudad de la Habana
2. Municipio: Playa
3. Fecha del Inicio de Brote:
4. Fecha de Comienzo del estudio:
5. Lugar del Brote:
6. Nombre del lugar:
7. Organismo:
8. Comensales:
9. Afectados: Extranjeros: Ingresados: Fallecidos: **(poner números)**
10. Síntomas:
11. Período de incubación:
12. Menú: **(TODO)**. Alimento sospechosos: **(Por tasa de ataque)**
13. Ingrediente contaminante:
14. Lugar de confección:
15. Existían Muestras testigo: **(enviadas al INHEN, Siempre previa consulta con el coordinador de Alimentos )**
16. Existían restos de alimentos:
17. Medidas Tomadas: **EJEMPLOS**, Educación sanitaria ,Decreto ley (,), Paralización Sanitaria de las actividades de elaboración y manipulación de alimentos.
18. Encuesta confeccionada por: **(.....) y en consulta con el Coordinador de Alimento**

Curva Epidémica

Casos



Anexo 3. Anexo No. 3



Unidad Municipal de Higiene y Epidemiología  
Órgano de Poder Popular  
Municipio Playa



Anexo No. 3

1. Provincia: Ciudad de la Habana
2. Brote: ETA
3. No: \_\_\_\_\_
4. Municipio: Playa
5. Fecha del Inicio de Brote: \_\_\_\_\_
6. Comienzo del estudio: \_\_\_\_\_
7. Local donde se produjo el Brote: \_\_\_\_\_
8. Nombre del lugar: \_\_\_\_\_
9. Organismo: \_\_\_\_\_
10. Expuestos:..... 11. Afectados:..... 12. Ingresados:..... 13. Fallecidos:.....
14. Edad (F: 0 a 4 años):.....(F: 5 a 11 años):.....(F: 12 a 18 años):.....(F: 19 a 44 años):.....  
(F: 45 a 64 años):..... (F: 65 y más):..... (F: Total):.....
15. Edad (M: 0 a 4 años):..... (M: 5 a 11 años):.....(M: 12 a 18 años):.....(M: 19 a 44 años):.....  
(M: 45 a 64 años):.....(M: 65 y más):..... (M: Total):.....
16. Extranjeros afectados:.....
17. Síntomas: (Diarreas):.....(Nauseas):.....(Vómitos):.....(Cólicos):.....(Fiebre):.....(Cefalalgia):.....(Artralgias).....  
(Mialgias):.....(Decaimiento):.....(Mareos):..... (Otro 1): Dolor abdominal:..... (Otro 2): Escalofríos:.....
18. Período de incubación:.....
19. Grupo de alimento sospechoso:.....
20. Específico: \_\_\_\_\_ (Para el grupo 10) B: bovina, P: porcina A: aviar O: otras
21. Alimento sospechoso: \_\_\_\_\_
22. Nombre científico: \_\_\_\_\_ (peces y plantas)
23. Ingrediente contaminante: \_\_\_\_\_
24. Confirmado: (S/N)
25. Lugar de confección: \_\_\_\_ (1) En el mismo lugar (2) En la cocina centralizada (3) Otros específico: \_\_\_\_\_
26. Lugar donde el alimento perdió su inocuidad: \_\_\_\_\_
27. Factores contribuyentes: Contaminación:..... Sobre vivencia:..... Multiplicación:.....
28. Existían Muestras testigo: S/N:
29. Existían restos de alimentos: S/N:
30. Resultados: \_\_\_\_\_
31. Resultados de los especímenes: \_\_\_\_\_
32. **Enfermos:** Diarreas: No. \_\_\_ Agente: \_\_\_\_\_ Pos: \_\_\_  
Vómitos: No. \_\_\_ Agente: \_\_\_\_\_ Pos: \_\_\_  
Sangre: No. \_\_\_ Agente: \_\_\_\_\_ Pos: \_\_\_  
Otros: No. \_\_\_ Agente: \_\_\_\_\_ Pos: \_\_\_
33. sanos: (copros-sanos): No. \_\_\_ Agente: \_\_\_\_\_ Pos: \_\_\_  
(Sangre-sanos): No. \_\_\_ Agente: \_\_\_\_\_ Pos: \_\_\_  
(Otros-sanos): No. \_\_\_ Agente: \_\_\_\_\_ Pos: \_\_\_
34. Ambiente: No. \_\_\_ (Agente-amb): \_\_\_\_\_ Pos: \_\_\_
35. Hisopaje: No. \_\_\_ (Agente-his): \_\_\_\_\_ Pos: \_\_\_
36. Agua: No. \_\_\_ (Agente-agua): \_\_\_\_\_ Pos: \_\_\_
37. Otro: No. \_\_\_ (Agente-otros): \_\_\_\_\_ Pos: \_\_\_
38. Etiología: Biológica: (B) Química: (Q) Desconocida (D)
39. Agente: Grupo: \_\_\_\_\_
40. Confirmado por laboratorio: S/N: \_\_\_\_\_
41. Confirmado por el Epidemiólogo: S/N: \_\_\_\_\_
42. Encuesta confeccionada por: (El coordinador de alimentos)

Informe final de Brote

(EJEMPLO)

La información del brote llega a través de las áreas de salud de los Policlínicos 1ro de Enero y Docente Playa, cuando se dan cuenta que están viendo un número de casos no usual con los mismos síntomas y que habían ingerido alimentos en el mismo lugar, el evento es comunicado de inmediato a la UMHE Playa y se activa la comisión de estudio de las ETA

En visita a la Cocina Centralizada donde se confeccionaron los alimentos, se detectan gran cantidad de deficiencias higiénicas de gran envergadura, por destacar algunas el incumplimiento de las normas elementales de conservación de los alimentos en frío, gran infestación de vectores y deterioro total de las condiciones higiénicas.

Se procedió a la aplicación de los decretos ley correspondientes, se impartió educación sanitaria y se paralizaron las actividades a de elaboración y manipulación de alimentos.

Se comprobó mediante la ISE que los alimentos implicados permanecieron expuestos a temperatura ambiente por tiempos prolongados entre su elaboración y consumo, tiempo suficiente para la multiplicación del microorganismo y la producción de la toxina, o bien sea durante su almacenamiento que fue totalmente inadecuado.

Tasa de Ataque

Alimentos servidos	Comieron				No comieron				Diferencia %
	E	NE	T	TA	E	NE	T	TA	
Amoz amarillo	33	9	42	79	4	1	5	80,0	- 1,4
Bopa de fideos	34	3	37	92	3	7	10	30	61,8
Pilete de pollo conformado	31	6	37	84	5	4	9	56	28,2
Harina de maíz	8	4	12	67	28	7	35	80	- 13,3
Galletas de dulce	8	2	10	80	28	8	37	78	1,8

Anexo 4. Diligencia de ocupación de muestra. Resultado de Análisis.

1126-1127

MOD. 85.08  
**MINISTERIO DE SALUD PUBLICA**  
**HIGIENE DE LOS ALIMENTOS**  
 Unidad Ejecutora: U.M.H.E. Playa

**DILIGENCIA DE OCUPACION DE MUESTRA RESULTADO DE ANALISIS**

Muest. o. RECOGIDA CON FECHA  

21	3	01	
D	M	A	HORA

 a.m. p.m.

EL QUE SUSCRIBE, CONSTITUIDO EN CUMPLIMIENTO DE ORDENES SUPERIORES EN:

ESTABLECIMIENTO - Nombre o razón social: <u>Yogurt</u>		GIRO: <u>Fabrica</u>	LICENCIA SANITARIA No.:	
DIRECCION - calle o carretera: <u>17727</u>	No. o KM.	LOCALIDAD:	MUNICIPIO: <u>Playa</u>	PROVINCIA: <u>C-11</u>
ADMINISTRADOR - 1er. Apellido: <u>Huay</u>	2do. Apellido: <u>Rodriguez</u>	Nombre: <u>Hugo</u>		
DIRECCION - calle o carretera:	No. o KM.	LOCALIDAD:	MUNICIPIO:	PROVINCIA:

PROCEDE A OCUPAR:

PRODUCTO: NOMBRE <u>Yogurt</u>	FORMA DE VENTA: Puro <input type="checkbox"/> Ampliado <input type="checkbox"/> Imitado <input type="checkbox"/>
CLASE DE MUESTRA: <u>Líquido</u>	ANÁLISIS: Químico <input type="checkbox"/> Bacteriológico <input checked="" type="checkbox"/>
Número de Muestras: <u>dos</u>	LECHE O PRODUC. CONG.: Temperatura: _____ Leche: _____ Grado: _____ Hora: _____ e.m. p.m.

ENVASE:  
Frasco Cristal ABIERTO  CERRADO SIN SEÑALES DE VIOLACION

Y LE HACE SABER AL INTERESADO:

1er. Apellido: <u>Huay</u>	2do. Apellido: <u>Rodriguez</u>	Nombre: <u>Hugo</u>
-------------------------------	------------------------------------	------------------------

En su carácter de aducir del derecho que tiene de solicitar una verificación del análisis en su oportunidad.  
 Las muestras ocupadas se remiten al Laboratorio EPHE 31,76 Huay para su análisis.  
 El interesado declara que el producto del que se ha tomado la muestra, es igual al resto que se encuentra en existencia en el establecimiento.  
 Y para constancia se extiende la presente diligencia en el lugar y fecha señalados.

NOTA: Si se toma la muestra de un envase mayor, ocupense muestras de etiquetas, o cópiese a continuación el texto de la misma.

<u>Federa Prod</u>
1- <u>19/3/01</u>
2- <u>19/3/01</u>

NOTA: Si se desea un análisis especial, hágase a continuación la aclaración:

Control de lactosa  
Estafilococo

INTERESADO: [Firma] Firma      TRABAJADOR SANITARIO: Lic. P. P. P. Firma

Fecha: \_\_\_\_\_ de \_\_\_\_\_ de 198\_\_  
 Este envase de \_\_\_\_\_ ha sido sellado y precintado para su análisis en el laboratorio: \_\_\_\_\_  
 Trabajador sanitario: \_\_\_\_\_  
 MUESTRIA No. \_\_\_\_\_

