

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



“Sistema Web de Información Georreferenciada del Estado Nutricional de Niños y Gestantes de la Dirección Regional de Salud de Ayacucho, 2016”

Tesis presentada por : Bach. SERRANO QUISPE, Pilar Gabriela

Para optar el título profesional de : Ingeniero de Sistemas

Tipo de Investigación : Aplicada

Área de Investigación : Ingeniería de software

Asesor : Ing. PERALTA SOTOMAYOR, Karel

AYACUCHO - PERÚ

2017

DEDICATORIA

A mis padres Epifanio y Maximiliana por su paciencia y comprensión infinita.

AGRADECIMIENTO

A mi Alma Mater, Universidad Nacional de San Cristóbal de Huamanga, por haberme formado con valores y conocimiento, por acogerme en sus aulas y brindarme la formación profesional.

A toda la plana docente de la Escuela de Formación Profesional de Ingeniería de Sistemas, con gratitud y reconocimiento imperecedero por sus enseñanzas y orientaciones durante mi permanencia en las aulas universitarias.

CONTENIDO

DEDICATORIA	ii
AGRADECIMIENTO	iii
RESUMEN.....	vii
INTRODUCCIÓN.....	viii
CAPÍTULO I.....	1
PLANTEAMIENTO DE LA INVESTIGACIÓN	1
1.1. DIAGNÓSTICO Y ENUNCIADO DEL PROBLEMA	1
1.2. DEFINICIÓN DEL PROBLEMA DE INVESTIGACIÓN.....	4
1.2.1. PROBLEMA PRINCIPAL.....	4
1.2.2. PROBLEMAS SECUNDARIOS.....	4
1.3. OBJETIVOS DE LA INVESTIGACIÓN	4
1.3.1. OBJETIVO GENERAL	4
1.3.2. OBJETIVOS ESPECÍFICOS	4
1.4. HIPÓTESIS DE LA INVESTIGACIÓN	4
1.5. JUSTIFICACIÓN Y DELIMITACIÓN DE LA INVESTIGACIÓN.....	5
1.5.1. IMPORTANCIA Y JUSTIFICACIÓN.....	5
1.5.2. DELIMITACIÓN.....	5
CAPÍTULO II.....	6
MARCO DE REFERENCIA DE LA INVESTIGACIÓN	6
2.1. ANTECEDENTES DE LA INVESTIGACIÓN	6
2.2. MARCO TEÓRICO	7
2.2.1. INFORMACIÓN GEORREFERENCIADA	7
2.2.2. ESTADO NUTRICIONAL.....	8
2.2.3. SOFTWARE	13
2.2.4. SISTEMA DE INFORMACIÓN	14
2.2.5. SISTEMA DE INFORMACIÓN GEOGRÁFICA (SIG).....	15
2.2.6. SISTEMA WEB.....	18
2.2.7. GOOGLE MAPS.....	18
2.2.8. METODOLOGÍA ICONIX	19

2.2.9.	Lenguaje de Programación C#	29
2.2.10.	Programación Orientada a Objetos con C#	31
2.2.11.	Plataforma .NET	33
2.2.12.	Programación con ASP.NET	35
2.2.13.	Lenguaje SQL	54
2.2.14.	Sistema Gestor de Base de Datos	56
2.2.15.	Enterprise Architect	58
2.2.16.	.NET Framework	58
2.2.17.	Ministerio de Salud	58
2.2.18.	Instituto Nacional de Salud	58
2.2.19.	Centro Nacional de Alimentación y Nutrición	59
2.2.20.	Dirección Regional de Salud	59
2.2.21.	Red de Salud	59
2.2.22.	Microrred de Salud	60
2.2.23.	Establecimientos de Salud	60
2.2.24.	Organización de Entidades por Niveles	61
CAPÍTULO III		63
METODOLOGÍA DE LA INVESTIGACIÓN		63
3.1.	Tipo de Investigación	63
3.2.	Nivel de la Investigación	63
3.3.	Método de la Investigación	63
3.4.	Diseño de la Investigación	63
3.5.	Población y Muestra	64
3.6.	Variables e Indicadores	64
3.7.	Técnicas e Instrumentos para la Recolección de Datos	64
3.8.	Técnicas de Procesamiento y Análisis de Datos	72
3.9.	Herramientas para el Tratamiento de Datos e Información	72
3.10.	Metodología Iconix para el Desarrollo del Sistema de Información Georreferenciada del Estado Nutricional de Niños y Gestantes	77
CAPÍTULO IV		86

ANÁLISIS Y RESULTADOS DE LA INVESTIGACIÓN	86
4.1. RESULTADOS Y ARTEFACTOS DEL SOFTWARE APLICANDO EL PROCESO DE ICONIX.....	86
4.1.1. ETAPA 1: ANÁLISIS DE REQUISITOS	86
4.1.2. HITO ETAPA 1: REVISIÓN DE REQUISITOS	97
4.1.3. ETAPA 2: ANÁLISIS Y DISEÑO PRELIMINAR	132
4.1.4. HITO ETAPA 2: REVISIÓN DEL DISEÑO PRELIMINAR (PDR).....	137
4.1.5. DISEÑO DETALLADO	137
4.1.6. DIAGRAMAS DE SECUENCIA	143
4.1.7. HITO ETAPA 3: REVISIÓN DEL DISEÑO DETALLADO	147
4.1.8. IMPLEMENTACIÓN	149
4.2. REVISIÓN DE CÓDIGO	152
4.2.1. PRUEBAS UNITARIAS.....	152
4.2.2. PRUEBAS FUNCIONALES	156
4.2.3. DIAGRAMA FÍSICO DE LA BASE DE DATOS	160
4.2.4. DIAGRAMA DE LA BASE DE DATOS.....	161
4.3. RESULTADOS DEL PROCESO DE DESARROLLO DEL SISTEMA.....	162
4.4. RESULTADOS DE LA EVALUACIÓN DE USABILIDAD DEL SISTEMA WEB GEORREFERENCIADO.....	169
4.5. DISCUSIÓN DE RESULTADOS	169
CAPÍTULO V.....	171
CONCLUSIONES Y RECOMENDACIONES	171
5.1. CONCLUSIONES	171
5.2. RECOMENDACIONES	172
BIBLIOGRAFÍA.....	173
ANEXO A: Planificación del proceso Iconix para el proyecto.....	178
ANEXO B: Calendario epidemiológico – 2016.....	179
ANEXO C: Reporte obtenido con el “Sistema Web de Información Georreferenciada del Estado Nutricional de Niños y Gestantes”	180
ANEXO D: Reporte obtenido con la Aplicación de escritorio Sistema Información del Estado Nutricional de Niños y Gestantes – propio de la DIRESA	181
ANEXO E: Tecnologías con la que cuenta cada Microrred de Salud.....	182

RESUMEN

Actualmente la Dirección Regional de Salud de Ayacucho (DIRESA) del Ministerio de Salud, está conformada por 65 establecimientos de salud, quienes hacen uso de una aplicación de escritorio para registrar datos de la atención de niños y gestantes, este aplicativo se encuentra aislado en cada establecimiento de salud y para obtener la información del estado nutricional primero se tiene que consolidar la información a nivel de microrred de salud, luego a red de salud y finalmente a nivel de dirección regional de salud; lo cual genera un proceso de información inadecuado y por ende lento, además que no se puede obtener información en tiempo real.

La presente investigación tiene como propósito implementar un sistema web georreferenciado que permita mostrar la información oportuna y exacta acerca del estado nutricional de niños y gestantes del Departamento de Ayacucho, de manera que las autoridades puedan focalizar y priorizar los grupos de población más vulnerables, por ende tomar decisiones orientadas a mejorar el estado nutricional de niños y gestantes. El tipo de investigación a estudiar es la investigación aplicada.

Para la implementación del sistema web se utilizará la metodología de desarrollo de software ICONIX, Google Maps para generar un Sistema de Información Geográfica y un sistema de administración de base de datos relacional.

PALABRAS CLAVE

Sistema de Información Geográfica, ICONIX, Estado nutricional, Niños, Gestantes.

INTRODUCCIÓN

En este mundo globalizado las Tecnologías de Información han tomado un papel muy importante, permiten ser cada vez más competitivos, ofrecer mejor calidad al consumidor, reducir costos, innovar con mayor rapidez y obtener resultados sorprendentes. Asimismo día a día las empresas van descubriendo el poder de la información y lo consideran como un activo muy valioso; teniendo presente esta tendencia se vio por conveniente implementar el sistema web de información georreferenciada del estado nutricional de niños y gestantes de la Dirección Regional de Salud de Ayacucho.

La desnutrición infantil es un problema que afecta a nuestra sociedad, que viene desde muchos años atrás; además de ser un problema permanente, la desnutrición limita el desarrollo económico del país al reducir la productividad del capital humano. El estado nutricional del niño se presenta así como un insumo esencial para el desarrollo social y económico para el país. Asimismo es importante evaluar el estado nutricional materno antes de la gestación y la ganancia de peso durante esta etapa, ya que son determinantes en el crecimiento fetal y peso del recién nacido.

Actualmente se cuenta con una aplicación de escritorio con muchos errores en su funcionamiento, ello implica que los trabajadores deban prescindir de esta aplicación y los datos a procesar, consolidar y validar sean realizados manualmente en archivos planos como hojas de cálculo y programas estadísticos, ocupando un porcentaje significativo del tiempo laboral del personal a cargo; por lo que se debe implementar un sistema web georreferenciado que permita mostrar la información oportuna y exacta acerca del estado nutricional de niños y gestantes, para que las autoridades puedan focalizar y priorizar los grupos de población más vulnerables y por ende tomar decisiones orientadas a mejorar el estado nutricional de niños y gestantes del departamento de Ayacucho.

CAPÍTULO I

PLANTEAMIENTO DE LA INVESTIGACIÓN

1.1. DIAGNÓSTICO Y ENUNCIADO DEL PROBLEMA

En Ayacucho, uno de los problemas que requieren una atención prioritaria con abordaje multidisciplinario e intersectorial integral es la desnutrición infantil, cuyas consecuencias se presentan a lo largo de todo el ciclo de vida. Las consecuencias de una deficiente alimentación además de inhibir el desarrollo cognitivo y físico de los menores, afectan también su potencial productivo futuro; comprometiendo la situación económica y social de la persona, la familia, la comunidad y el país.

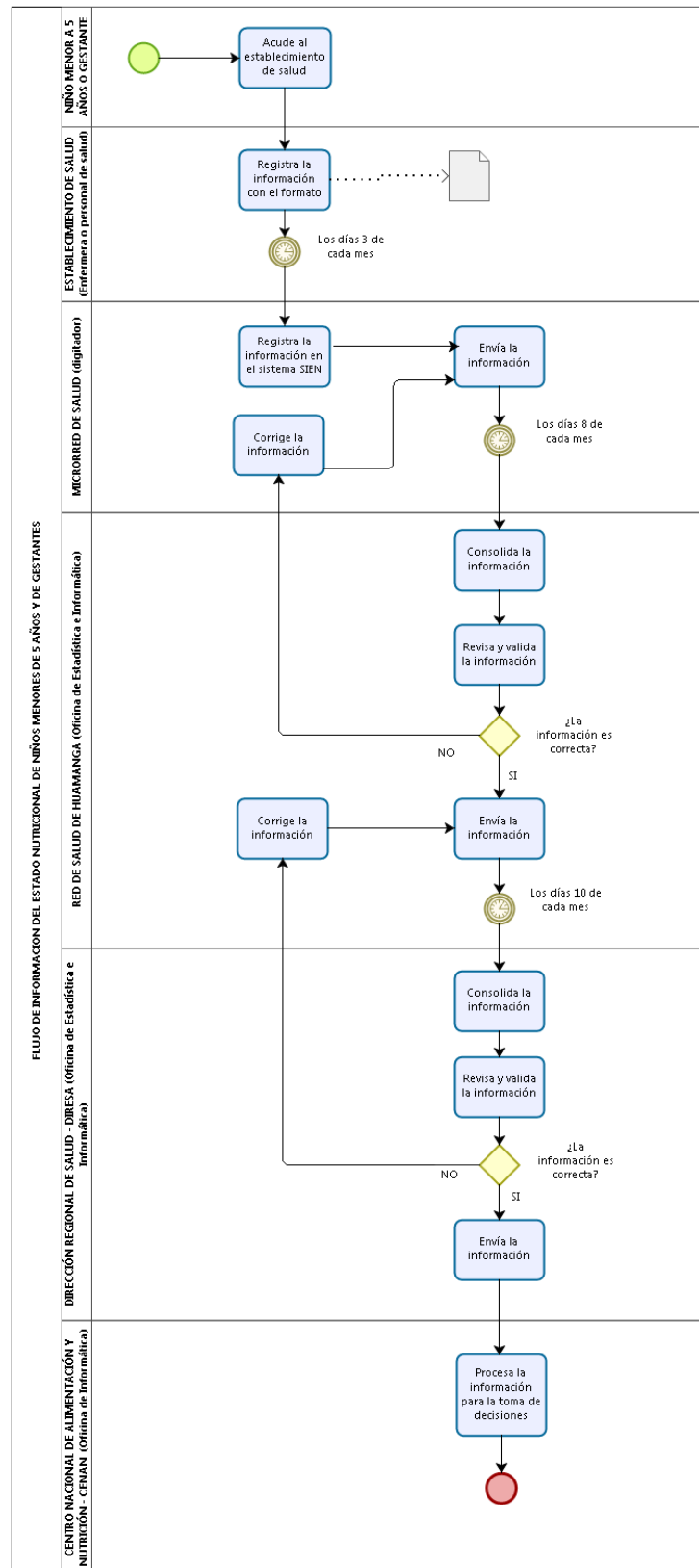
Actualmente los establecimientos de salud cuentan con un formato de registro diario del niño y de la gestante, en el cual el personal de salud registra manualmente la información de niños menores de 5 años y de madres gestantes, acabado este registro durante el mes de trabajo, la información plasmada en el formato es enviada a un punto de digitación; la digitación se realiza en un aplicativo de escritorio denominado Sistema de Información del Estado Nutricional de Niños y Gestantes (SIEN); una vez ingresada la información al aplicativo, la base de datos es enviada (cada 3 de cada mes) a la Microrred de Salud para su respectiva consolidación a nivel de establecimientos de salud, luego debe ser enviada (cada 8 de cada mes) a la Red de Salud (para su consolidación a nivel de microrredes de salud), después debe ser enviada (cada 10 de cada mes) a la Dirección Regional de Salud (DIRESA) donde se realiza el control de calidad y consolida mensualmente la información a nivel de redes de salud y se remite con periodicidad trimestral al Centro Nacional de Alimentación y Nutrición (CENAN), donde se consolida a nivel nacional. Cabe precisar que el flujo de recolección de la información es muy deficiente, ya que el aplicativo que se usa es una aplicación de escritorio, el mismo que para consolidar la información de una entidad a otra necesariamente se debe trasladar la base de datos a través de un dispositivo de almacenamiento, lo que ocasiona demora y pérdida de tiempo en la consolidación de la información; además a la hora de consolidar la información primero se tiene que realizar un control de calidad de los registros, asimismo verificar los datos considerados fuera de rango

del niño y de la gestante, toda esta evaluación debería ser realizarlo por el mismo sistema en la primera fase de registro, ya que todo sistema debería de validar el ingreso de información de cada campo de registro, además de que los reportes que emite el sistema no son adecuados para la toma de decisiones, ya que no focalizan los sectores más vulnerables a la desnutrición. Por ende las autoridades no pueden tomar decisiones adecuadas y oportunas.

A la fecha, el Instituto Nacional de Salud a través de su página web tiene publicada la información del estado nutricional de niños menores de 5 años y gestantes del año 2014, lo cual implica que las autoridades no cuentan con información actualizada para priorizar los grupos de población más vulnerables, asimismo, la focalización de las intervenciones en materia de salud y de desarrollo económico, social del sector público y privado son inadecuadas. En efecto, existe una creciente demanda de información estadística confiable a esos niveles administrativos para hacer más eficaz y eficiente la ejecución de los recursos financieros que se asignan a los gobiernos regionales y locales, principalmente en el área social.

Figura N° 1.1: Flujo de información del estado nutricional de niños y gestantes. Fuente:

Elaboración propia



1.2. DEFINICIÓN DEL PROBLEMA DE INVESTIGACIÓN

1.2.1. PROBLEMA PRINCIPAL

¿Cómo se relaciona la información georreferenciada con el estado nutricional de niños y gestantes de la Dirección Regional de Salud de Ayacucho, 2016?

1.2.2. PROBLEMAS SECUNDARIOS

- a. ¿De qué manera la información georreferenciada ayudará a realizar la gestión de información del estado nutricional de niños y gestantes?
- b. ¿De qué manera mejorará el estado nutricional de niños y gestantes con el uso de la información georreferenciada?
- c. ¿Cuál es la probabilidad de que a través de un mapa cuantitativo mejore la representación de la información del estado nutricional de niños y gestantes?

1.3. OBJETIVOS DE LA INVESTIGACIÓN

1.3.1. OBJETIVO GENERAL

Implementar un sistema web georreferenciado que permita mostrar la información oportuna y exacta acerca del estado nutricional de niños y gestantes de la Dirección Regional de Salud de Ayacucho.

1.3.2. OBJETIVOS ESPECÍFICOS

- a. Garantizar la capacitación acerca del manejo de la información georreferenciada para que sea monitoreada y que se tome una adecuada decisión en los planes de control asistencial en el estado nutricional de niños y gestantes, en la provincia de Huamanga.
- b. Analizar los resultados de la información georreferenciada para implementar planes que conlleve a mejorar el estado nutricional de los niños y gestantes, en la provincia de Huamanga.
- c. Desarrollar la representación de un mapa cuantitativo que permita mostrar la representación de la información que queremos utilizar según la provincia que se está monitoreando.

1.4. HIPÓTESIS DE LA INVESTIGACIÓN

Con la implementación de un sistema web de información georreferenciada se

logrará tener la información oportuna y exacta, la cual permitirá tomar mejores decisiones.

1.5. JUSTIFICACIÓN Y DELIMITACIÓN DE LA INVESTIGACIÓN

1.5.1. IMPORTANCIA Y JUSTIFICACIÓN

La estimación de la desnutrición crónica para áreas menores, como son provincias o distritos, constituye una herramienta útil para los decisores de política. En efecto, existe una creciente demanda de información estadística confiable a esos niveles administrativos para hacer más eficaz y eficiente la ejecución de los recursos financieros que se asignan a los gobiernos regionales y locales, principalmente en el área social. (INEI, 2009, p. 01).

Contar con información de la prevalencia de desnutrición crónica en niños menores de 5 años a nivel distrital, permitirá determinar y priorizar los grupos de población más vulnerables, asimismo, focalizar las intervenciones en materia de salud y de desarrollo económico y social del sector público y privado. (INEI, 2009, p. 07).

Se procederá a implementar un sistema web para registrar la información del estado nutricional de niños y gestantes, con participación del digitador de cada establecimiento de salud perteneciente a un distrito de la provincia de Huamanga, la cual nos facilitará acceder a la información georreferenciada en tiempo real. Con la información georreferenciada las autoridades podrán tomar decisiones adecuadas en la distribución de recursos, de esta manera se beneficiará la población de niños con más vulnerabilidad a la desnutrición.

1.5.2. DELIMITACIÓN

La investigación se realizará en la región de Ayacucho, la información de los indicadores de investigación se obtendrán de la Red de Salud de Huamanga (el cual comprende a los 16 distritos de la provincia de Huamanga) del año 2016. La presente investigación tiene como variable de investigación al Estado Nutricional de Niños y Gestantes.

CAPÍTULO II

MARCO DE REFERENCIA DE LA INVESTIGACIÓN

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

Según la Oficina Nacional de Gobierno Electrónico e Informática (ONGEI, 2014), en la región existen experiencias de sistemas de información geográfica para la gestión del desarrollo territorial, que tiene como fin contribuir a una eficiente y oportuna toma de decisiones; para ello, el sistema se enfoca en apoyar a los actores involucrados con información político - administrativa, socio-económica y ambiental georreferenciada para facilitar la elaboración de diagnósticos, seguimiento, evaluación y control de sus respectivos planes de desarrollo; contribuir en la definición de estrategias y focalización de metas en los planes de ordenamiento territorial y generar cultura en el uso de la información geográfica.

Longhi (2012) indica que en un proyecto académico sobre sistemas de información geográfica y desnutrición en el norte grande argentino; los Sistemas de Información Geográfica (SIG) se revelan como una herramienta adecuada para el análisis de la situación de salud, la investigación operacional y la vigilancia con vistas a la prevención y al control de problemas relacionados con la desnutrición; se constituyen, pues, en un instrumento útil para la aplicación de políticas públicas.

Burstein (2002) en su investigación concluye que en salud pública específicamente, los SIG son herramientas poderosas, ya que permiten combinar datos demográficos (edad, sexo, educación, etc.) con datos de salud (tipos de enfermedades, incidencias, prevalencias, características clínicas o patológicas, etc.), características del medio natural (clima, altitud, precipitación, etc.) y con cualquier otra información que el especialista considere necesaria. Estas virtudes han convertido a los SIG en los últimos años en herramientas para el análisis geográfico de gran difusión debido a la multitud de actividades en los que pueden ser útiles. Estos sistemas procuran, en definitiva, “espacializar” los procesos humanos, es decir, ubicar los hechos sociales en su territorio, considerando la mutua interacción y el

cambio permanente entre la actividad humana y el medio. Tal propósito se logra combinando información cartográfica con estadística poblacional.

La estimación de la desnutrición crónica para áreas menores, como son provincias o distritos, constituye una herramienta útil para los decisores de política. En efecto, existe una creciente demanda de información estadística confiable a esos niveles administrativos para hacer más eficaz y eficiente la ejecución de los recursos financieros que se asignan a los gobiernos regionales y locales, principalmente en el área social. (INEI, 2009, p. 03).

2.2. MARCO TEÓRICO

2.2.1. INFORMACIÓN GEORREFERENCIADA

La información es un conjunto de datos con un significado, que reduce la incertidumbre o que aumenta el conocimiento de algo. En verdad, la información es un mensaje con significado en un determinado contexto, disponible para uso inmediato y que proporciona orientación a las acciones por el hecho de reducir el margen de incertidumbre con respecto a nuestras decisiones. (Chiavenato, 2006)

Geo significa 'tierra' o 'la Tierra' (Real Academia Española, 2001)

Según RAE (2001), referencia significa “En un escrito, indicación del lugar de él mismo o de otro al que se remite al lector.”

De acuerdo a los conceptos anteriores de geo y referencia, se puede decir que el término georreferencia significa la indicación de un lugar sobre la Tierra.

Entonces Información georreferenciada, es un conjunto de datos que nos indican un lugar sobre la tierra.

A. GESTIÓN DE INFORMACIÓN

Según Bartle (2009) es el proceso de analizar y utilizar la información que se ha recabado y registrado para permitir a los administradores (de todos los niveles) tomar decisiones documentadas. La información para la gestión es la información necesaria para tomar decisiones de gestión.

Curto (2006) afirma que es un proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma.

B. MAPA CUANTITATIVO

Mori (s.f) afirma que un mapa cuantitativo se elabora a partir de datos estadísticos y muestra la relación que tiene la información.

Usandivaras (s.f.) indica que un mapa cuantitativo muestra aspectos espaciales de datos numéricos. Estos mapas muestran los datos en una escala ordinal (más que, menos que), y en escalas de intervalo y proporción (cuánto más que).

Según ATLAS SIG HYPARION (s.f.) es un mapa temático que representa la distribución de fenómenos y hechos de acuerdo con su importancia numérica expresada de forma absoluta o relativa.

Un mapa temático es “Mapa que, sobre una base topográfica elemental de referencia, destaca, mediante la utilización de diversos recursos de las técnicas cartográfica, correlaciones, valoraciones o estructuras de distribución de algún tema concreto y específico.” (ATLAS SIG HYPARION, s.f., párr. 3)

2.2.2. ESTADO NUTRICIONAL

El estado nutricional de una persona o un colectivo es el resultado de la interrelación entre el aporte nutricional que recibe y las demandas nutritivas del mismo, necesarias para permitir la utilización de nutrientes, mantener las

reservas y compensar las pérdidas. (Majem y Aranceta, 2006, p. 114)

La Organización Mundial de la Salud, ha establecido tres medidas básicas para evaluar un estado nutricional en niños, estas son: peso para la edad, talla para la edad y peso para la talla.

2.2.2.1. DESNUTRICIÓN

Una de las formas de malnutrición es la desnutrición, resultado de un consumo insuficiente de alimentos o por la presencia de enfermedades, como las infecciones virales, bacterianas o parasitarias, que afectan el aprovechamiento de los nutrientes en la persona. Estas dos causas se pueden presentar de manera conjunta, agravando la situación nutricional, manifestándose no sólo en una disminución del crecimiento físico, sino, principalmente en el deterioro de la capacidad productiva, mental y física. Esta situación es más desventajosa en los niños, debido a que el riesgo de padecer desnutrición está relacionado a la velocidad del crecimiento, es decir, cuanto más rápido está creciendo un niño, mayores son sus necesidades nutricionales. (FAO, 2002).

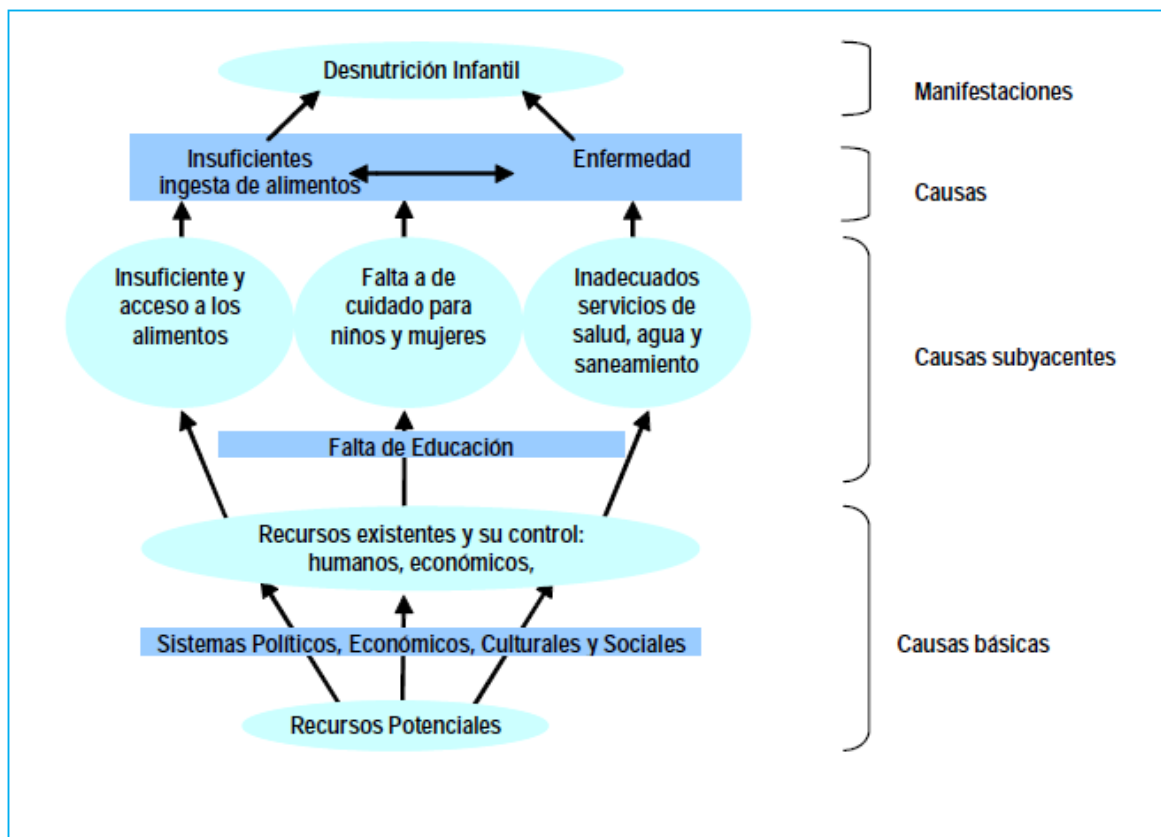
Según el Instituto Nacional de Salud (INS, 2011) la desnutrición es una afección que ocurre cuando su cuerpo no recibe los nutrientes suficientes. Hay muchas causas de desnutrición. Éstas pueden surgir a raíz de:

- Dieta inadecuada o balanceada
- Problemas con la digestión o la absorción
- Ciertas afecciones médicas

Para evaluar el estado nutricional de una persona, modalidad más utilizada por su bajo costo es la medición antropométrica, principalmente las medidas de talla y peso, que asociadas a la edad y sexo, y; confrontadas con el patrón de referencia, permiten contar con un diagnóstico del estado nutricional del niño. Con esta información se obtiene tres tipos de indicadores: desnutrición crónica, desnutrición aguda y desnutrición global. (Segura, Montes, Hilario, Asenjo, Baltazar, 2002)

Cuando se trata de evaluar el estado nutricional de una persona el uso de los tres indicadores tienen un carácter complementario.

Figura N° 2.1: Marco conceptual de la desnutrición infantil. Fuente: UNICEF – Estado mundial de la infancia (1998)



A. DESNUTRICIÓN CRÓNICA (Talla para la edad o retardo del crecimiento)

Consiste en un menor crecimiento lineal. Por depender del desarrollo de los huesos largos, tiene una dinámica menos flexible que el peso para la talla. De hecho, excepto bajo condiciones muy particulares, se considera que el déficit de talla para la edad es una condición prácticamente irreversible. (Segura et al., 2002)

Retardo de altura para la edad (A/E). Asociada normalmente a situaciones de pobreza, y relacionada con dificultades de aprendizaje y menos desempeño económico. (Programa Especial para la Seguridad Alimentaria, 2002)

Según la INEI (2009a), la desnutrición crónica se determina al comparar la talla del niño con la esperada para su edad y sexo. La desnutrición crónica refleja el resultado de la influencia de factores socioeconómicos, nutricionales y de salud.

El nivel de desnutrición crónica en niños próximos a cumplir los cinco años, es un indicador

de los efectos acumulativos del retardo en el crecimiento. La desnutrición crónica refleja el resultado de la influencia de factores socioeconómicos, nutricionales y de salud. Es un indicador para medir los problemas de desarrollo de la niñez, por su estrecha relación con problemas de aprendizaje, deserción escolar y, a la larga, déficit en la productividad de la persona adulta.

La información para elaborar este indicador es la talla (estatura o longitud en centímetros), edad y sexo. Para ello, se requiere de personal capacitado y estandarizado en la aplicación de las técnicas de recojo de información antropométrica. (INEI, 2009a)

B. DESNUTRICIÓN AGUDA (Peso para la talla)

Deficiencia de peso por altura (P/A). Delgadez extrema. Resulta de una pérdida de peso asociada con periodos recientes de hambruna o enfermedad que se desarrolla muy rápidamente y es limitada en el tiempo. (Programa Especial para la Seguridad Alimentaria, 2002)

Consiste en la disminución de masa corporal, inicialmente a expensas de tejido graso pero que puede afectar incluso la masa muscular. Generalmente es consecuencia de un episodio agudo de enfermedad infecciosa o severa disminución en la ingesta calórica. (Segura et al., 2002)

Según la INEI (2009a), el peso para la talla es útil para determinar el estado nutricional actual de los niños. Permite diferenciar si un niño se encuentra delgado (desnutrición aguda). La información para elaborar este indicador es el peso en kilogramos, estatura/longitud en centímetros, edad y sexo.

C. DESNUTRICIÓN GLOBAL (Peso para la edad)

Es una combinación de los dos anteriores. Por haber sido el primero en contar con tablas de referencia, fue muy usado en las primeras cartillas para evaluar el estado nutricional infantil. Consideramos no recomendable su uso como único método de evaluación, ya que niños con déficit en talla pueden ser calificados como normales por contar con una masa corporal que les permite alcanzar el peso esperado para su edad. (Segura et al., 2002)

Deficiencia de peso para la edad. Insuficiencia ponderal. Índice compuesto por $(P/A \times A/E = P/E)$. (Programa Especial para la Seguridad Alimentaria, 2002).

La INEI (2009a) resalta que la desnutrición global es considerada como un indicador general de la desnutrición, pues no diferencia la desnutrición crónica, que se debería a factores estructurales de la sociedad, de la aguda, que corresponde a pérdida de peso reciente. Para elaborar este indicador se requiere de información sobre edad, peso en kilogramos y sexo.

2.2.2.2. ANEMIA

RAE (2001) define a la anemia como el empobrecimiento de la sangre por disminución de su cantidad total, como ocurre después de las hemorragias, o por enfermedades, ya hereditarias, ya adquiridas, que amenguan la cantidad de hemoglobina o el número de glóbulos rojos.

Rodríguez (2007) manifiesta que la anemia se define como un estado clínico caracterizado por un descenso de la masa de eritrocitos o bien una disminución de la hemoglobina. La masa de hematíes puede expresarse como número por unidad de volumen, o como hematocrito, que es la relación existente entre el volumen eritrocitario y el volumen del plasma.

2.2.2.3. OBESIDAD

Baltazar (2007) afirma que la obesidad es una condición corporal caracterizada por el almacenamiento de una cantidad excesiva de grasa bajo la piel y en el interior de ciertos órganos como: músculos, corazón, hígado, etc.

La obesidad se define como un exceso de grasa, general o localizada, que se manifiesta en un valor de peso elevado comparado con el de personas de la misma edad y sexo (Ortega y Requejo, 2000)

2.2.2.4. SOBREPESO

La RAE (2001) define la palabra sobrepeso como excesiva acumulación de grasa en el cuerpo.

2.2.3. SOFTWARE

“Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.” (IEEE, 1993, p. 18).

Programas de ordenador y la documentación asociada. Los productos de software se pueden desarrollar para algún cliente en particular o para un mercado general (Sommerville, 2011).

Según Winblad y King (2003) software es el equipamiento lógico o soporte lógico de una computadora digital; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware.

2.2.3.1. INGENIERÍA DE SOFTWARE

“La ingeniería de software es el establecimiento y uso de principios sólidos de ingeniería, orientados a obtener software económico que sea fiable y trabaje de manera eficiente en máquinas reales.” (Bauer, 2002, p. 94).

Según Sommerville (2011) la ingeniería de software es una disciplina de ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento después de que se utiliza. La ingeniería de software comprende las formas prácticas para desarrollar y entregar un software útil.

“La ingeniería de software permite elaborar consistentemente productos correctos, utilizables y costo-efectivos.” (Según Cota, 1994, p. 5)

“La ingeniería de software incluye un proceso, métodos y herramientas para administrar y hacer ingeniería con el software. El fundamento en el que se apoya la ingeniería de software es el compromiso con la calidad.” (Pressman, 2010, p. 11)

Figura N° 2.2. Capas de la Ingeniería de Software. Fuente: Pressman (2010)



2.2.4. SISTEMA DE INFORMACIÓN

“Un sistema de información es un conjunto de personas, datos, procesos y tecnología de la información que interactúan para recoger, procesar, almacenar y proveer la información necesaria para el correcto funcionamiento de la organización.” (Whitten, Bentley y Dittman, 2004, p. 38).

Un sistema de información es un sistema que reúne, almacena, procesa y distribuye conjuntos de información entre los diferentes elementos que configuran una organización, y entre la organización misma y su entorno (Pastor, 2005).

“Un sistema de información posee cuatro actividades básicas que son de entrada, almacenamiento, procesamiento y salida de información.” (Pressman, 2005, p. 5).

2.2.4.1. ACTIVIDADES DE UN SISTEMA DE INFORMACIÓN

a. Entrada de Información

Amaya (2010), afirma que es el proceso mediante el cual el sistema de información toma los datos que requiere para procesar la información.

“Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos.” (Tamayo, 2000, p. 46).

b. Almacenamiento de Información

“El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior.” (Amaya, 2005, p. 24).

c. Procesamiento de Información

“Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida” (Fernández, 2006, p. 75).

El procesamiento de Información permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base. (Tamayo, 2000, p. 47)

d. Salida de Información

“La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior.” (Fernández, 2006, p. 27).

“Un sistema de información es el sistema de personas, registros de datos y actividades que procesa los datos y la información en cierta organización, incluyendo manuales de procesos o procesos automatizados.” (Amaya, 2005, p. 38).

2.2.5. SISTEMA DE INFORMACIÓN GEOGRÁFICA (SIG)

El Instituto de Investigación de Sistemas ambientales (ESRI, 2004), define a un SIG como una colección organizada de hardware, software, datos geográficos y de personal diseñada para capturar, almacenar, actualizar, manipular, analizar y mostrar eficientemente todas las formas de información referenciada geográficamente.

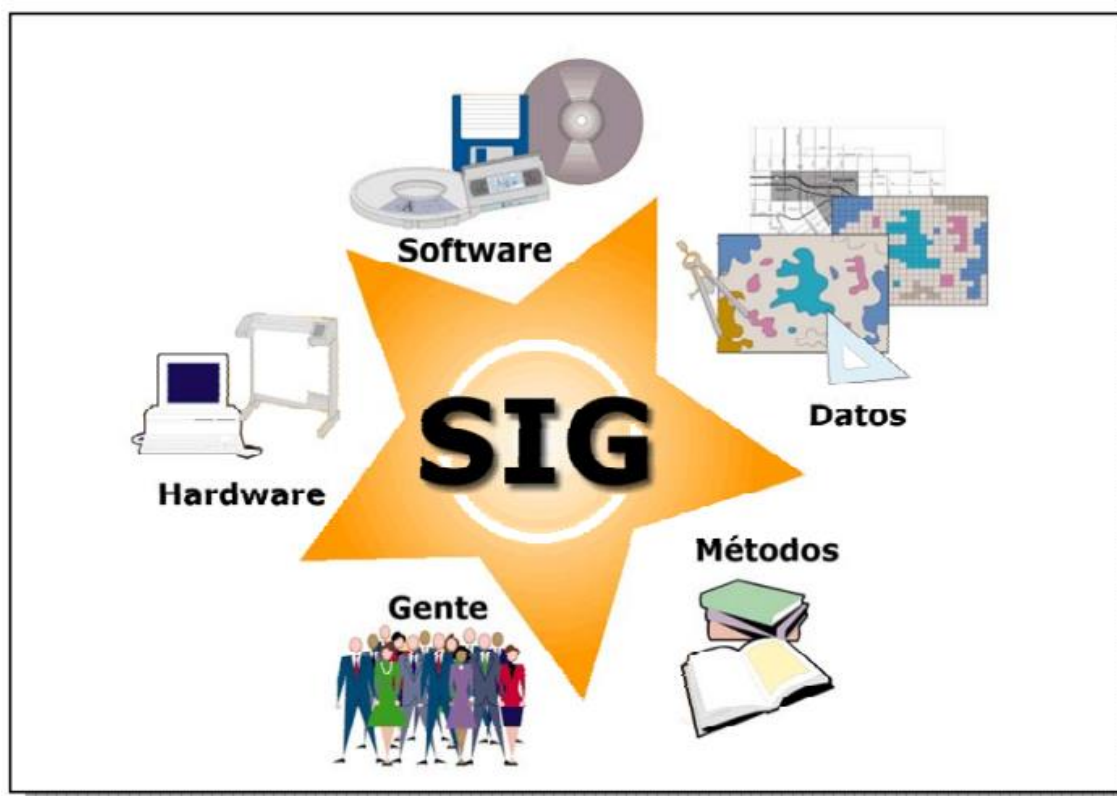
El sistema corporativo de información geográfica (SICORI, 2001), define a un SIG como una disciplina basada en conocimientos, metodologías y procedimientos asistidos por

computadora, que permiten la incorporación, almacenamiento, manipulación, procesamiento, consulta y presentación de información referenciada geográficamente en formatos gráficos y no gráficos.

2.2.5.1. COMPONENTES:

Para comprender mejor el funcionamiento de un SIG es importante conocer cuáles son los componentes que lo constituyen.

Figura N° 2.3: Componentes SIG. Fuente: ESRI (2004)



En la figura N° 2.3 podemos observar los componentes que conforman un SIG, y a continuación se explica cada uno de estos componentes:

- a. **Hardware.-** Es el equipo y los periféricos en los cuales el SIG opera, puede ser desde una estación de trabajo hasta un servidor centralizado. Los equipos pueden operar en modo stand-alone o networked-configuration. Algunos ejemplos de hardware pueden ser: computadoras, redes, dispositivos periféricos, impresora, plotters y digitalizadores.
- b. **Software.-** Es el que provee las funciones y herramientas que el usuario necesita para

almacenar, analizar, y mostrar la información geográfica. Los componentes de software clave son: Software SIG, Software de BD, Software de SO y Software de red, recordando que el Software SIG se considera al Software comercial y al Software abierto.

- c. **Datos.-** Unos de los componentes más importantes de un SIG son los datos. Es absolutamente esencial que la información sea precisa. Algunos diferentes tipos de datos son: datos vectoriales, Datos raster, Datos de imágenes Datos de atributos.
- d. **Gente.-** La tecnología SIG es claramente de valor limitado sin la gente que administre el sistema y desarrolle planes para su aplicación. Los usuarios SIG pueden ser desde especialistas técnicos calificados hasta planeadores y analistas de mercado quienes usen los SIG para ayudarse con sus tareas diarias. Ejemplo de gente SIG pueden ser: Administradores, Gerentes, Técnicos SIG, Expertos de aplicación, Usuarios finales, Consumidores, Geólogos, Urbanistas.
- e. **Métodos.-** Los métodos son los planes bien diseñados y reglas de negocio específicas de la aplicación que describen como la tecnología es aplicada. Esto incluye: Pautas, Especificaciones, Estándares y Procedimientos.

2.2.5.2. FUNCIONES DE UN SISTEMA DE INFORMACIÓN GEOGRÁFICA

Según Comas y Ruiz (1993), las funciones de un Sistema de Información Geográfica son:

- a. **Funciones de entrada de información.-** Son los procedimientos que convierten la información geográfica en formato analógico al formato digital propio de los ordenadores, mediante escaneo, digitalización, integración de ortoimágenes, integración de ficheros CAD, importaciones de base de datos, etc.
- b. **Funciones de gestión de la información espacial.-** Capacidad de organización de base de datos relacionales, asociación de datos gráficos y datos alfanuméricos, organización interna.
- c. **Funciones analíticas.-** Constituyen el elemento más característico de los Sistemas de Información Geográfica. A través del procesamiento de los datos, se consigue obtener mayor información de la que se disponía en un principio. Esta nueva información es almacenada y puede ser utilizada para nuevos procesos de análisis. Tareas analíticas propias de los SIG son la clasificación y superposición de coberturas.
- d. **Funciones de salida de información.-** Un SIG debe permitir obtener mapas, gráficos,

tablas de datos numéricos, etc. de los datos obtenidos tras los distintos procesos de análisis o de aquellos que simplemente se almacenan en la base de datos.

2.2.5.3. DATOS GEOGRÁFICOS

Bosque (2000) define los datos geográficos como datos relacionados a características y recursos del planeta tierra, así como a actividades humanas basadas en o asociadas a estas características y recursos.

2.2.5.4. BASE DE DATOS GEOGRÁFICA

Ochoa (2003) indica que es un almacén de datos ya sea gráficos o descriptivos y éstos están organizados generalmente en tablas relacionales.

Según ONGEI (2009) define como colección de datos interrelacionados y un conjunto de programas para acceder a los mismos. Contienen datos alfanuméricos y cartográficos.

2.2.6. SISTEMA WEB

Según Kendall, K. y Kendall, J. (2011) se denomina sistema web al sistema que se hace migrar a la World Wide Web o si se concibe e implementan originalmente como tecnologías en web, agregándole con esto una mayor funcionalidad.

Existen muchos beneficios relacionados con el proceso de implementar o mejorar un sistema en web, a continuación se muestra los siguientes:

- a. Un sistema web aumenta el número de usuarios que se enteran de la disponibilidad de un servicio, producto, industria, persona o grupo.
- b. Un sistema web hace que los usuarios tengan la posibilidad de acceder al sistema las 24 horas del día.
- c. Un sistema web mejora la utilidad y capacidad de uso del diseño de la interfaz.
- d. Un sistema web hace que se pueda expandir a un sistema globalmente en vez de permanecer en el entorno local, con lo cual se puede establecer contacto con personas en ubicaciones remotas sin preocuparse por la zona horaria en la que se encuentren.

2.2.7. GOOGLE MAPS

Google Maps nace en febrero del 2005 como desarrollo de Google Labs. Herramienta planeada para trabajar útilmente y en conjunto con Google Local.

Google, dentro de sus amplios servicios gratuitos tiene a *Google Maps* como un API el cual nos permite insertar mapas en nuestras aplicaciones Web, brindándonos diversas utilidades y eventos para manipularlos.

Google Maps es un servicio que ofrece el portal Google, se define a sí mismo como tecnología amigable para recibir indicaciones de cómo llegar a diversos lugares, ubicación de diferentes negocios, así como mapas detallados de diferentes países en el mundo.

Algunas de las funciones que contiene Google Maps son:

- Resultados integrales de búsqueda de negocios.
Te permite encontrar diferentes negocios.
- Mapas arrastrables (draggable).
Aplicación que te permite observar a los alrededores de las áreas seleccionadas en el mapa sin tener que esperar a que se cargue la aplicación.
- Vista de imágenes vía satélite.
Permite el acercamiento y alejamiento de la visión.
- Visión del terreno.
Permite ver relieves en los mapas mediante sombreados así como montañas y demás vegetación.
- Visión de calle.
Permite la navegación a través de avenidas y calles.
- Direcciones a detalle.
Le permite al usuario especificar una dirección a la cual desea llegar y obtener instrucciones de cómo trasladarse de un punto a otro mediante las calles y avenidas señaladas.

2.2.8. METODOLOGÍA ICONIX

ICONIX proceso es un análisis de casos de uso basada en el diseño y la metodología. Su foco principal está en cómo llegar de forma fiable a partir de casos de uso de código en menor número de pasos posible. (Rosenberg, Stephens y Collins-Cope, 2005).

EL PROCESO ICONIX

Proceso de desarrollo de software práctico, tiene un enfoque racionalizado y minimalista centrado en la zona entre los casos de uso y el código. ICONIX establece un procedimiento sencillo y un conjunto mínimo de pasos que dan lugar a buenos resultados; resultados que han demostrado ser coherentes y repetibles. Es un proceso simplificado en comparación con otros procesos más tradicionales, que unifica un conjunto de métodos de orientación a objetos con el objetivo de abarcar todo el ciclo de vida de un proyecto. Presenta claramente las actividades de cada etapa y exhibe una secuencia de pasos que deben ser seguidos. Está entre la complejidad del RUP (Rational Unified Processes) y la simplicidad de XP (Extreme Programming), sin eliminar las tareas de análisis y diseño que XP no contempla.

Características de ICONIX

- a. Iterativo e incremental**, varias iteraciones ocurren entre el desarrollo del modelo del dominio y la identificación de los casos de uso. El modelo estático es incrementalmente refinado por los modelos dinámicos.
- b. Trazabilidad**, cada paso está referenciado por algún requisito. Se define trazabilidad como la capacidad de seguir una relación entre los diferentes “artefactos de software” producidos.
- c. Dinámica del UML**, La metodología ofrece un uso “dinámico” del UML por que utiliza algunos diagramas del UML, sin exigir la utilización de todos, como en el caso de RUP.

Rosenberg y Stephens (2007), destacan un análisis de requisitos, un análisis y diseño preliminar, un diseño y una implementación como las principales tareas.

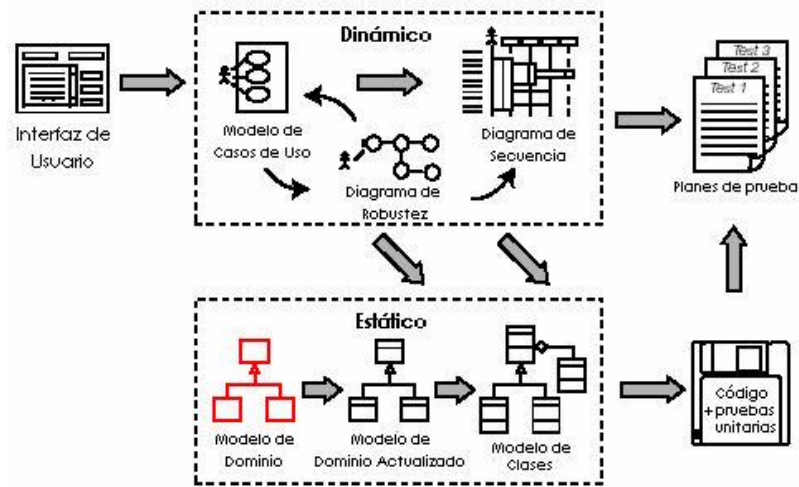
TÉCNICAS DE LA METODOLOGÍA ICONIX

Según Rosenberg y Stephens (2007), las técnicas de la metodología ICONIX son:

a. MODELO DE DOMINIO

El Modelo de Dominio es un artefacto colaborativo vivo. Es refinado y actualizado en cada parte el proyecto, de modo que refleja siempre la comprensión actual del espacio del problema. (Rosenberg y Stephens, 2007)

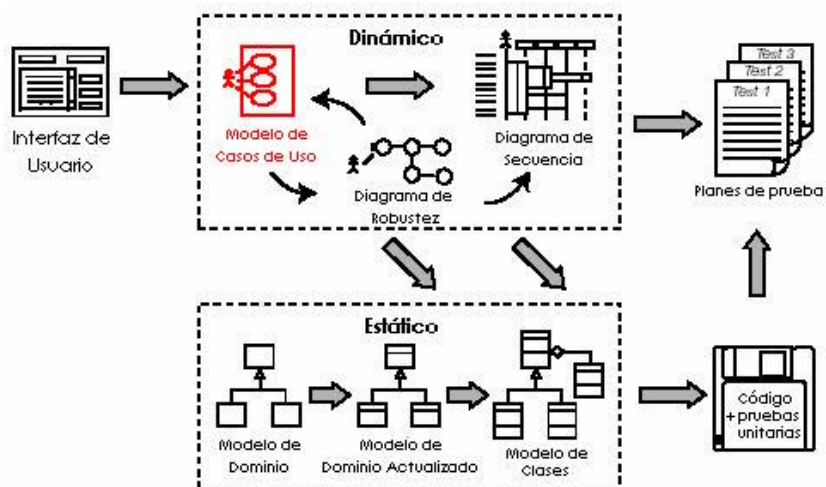
Figura N° 2.4: Modelo de dominio, adaptado. Fuente: Doug Rosenberg and Matt Stephens (2007)



b. MODELO DE CASOS DE USO

Con un primer modelo de dominio inicial en su lugar, es hora de comenzar a escribir los casos de uso. Los casos de uso dan un modo estructurado de capturar los requisitos de comportamiento de un sistema, de modo que puede razonablemente crear un diseño desde ellos. Le ayudan a responder ciertas preguntas fundamentales: *¿Qué están tratando de hacer los usuarios del sistema?* *¿Cuál es la experiencia del usuario?* Una cantidad sorprendente de lo que su software debe hacer se dicta por el modo en que los usuarios deben interactuar con él. (Rosenberg y Stephens, 2007).

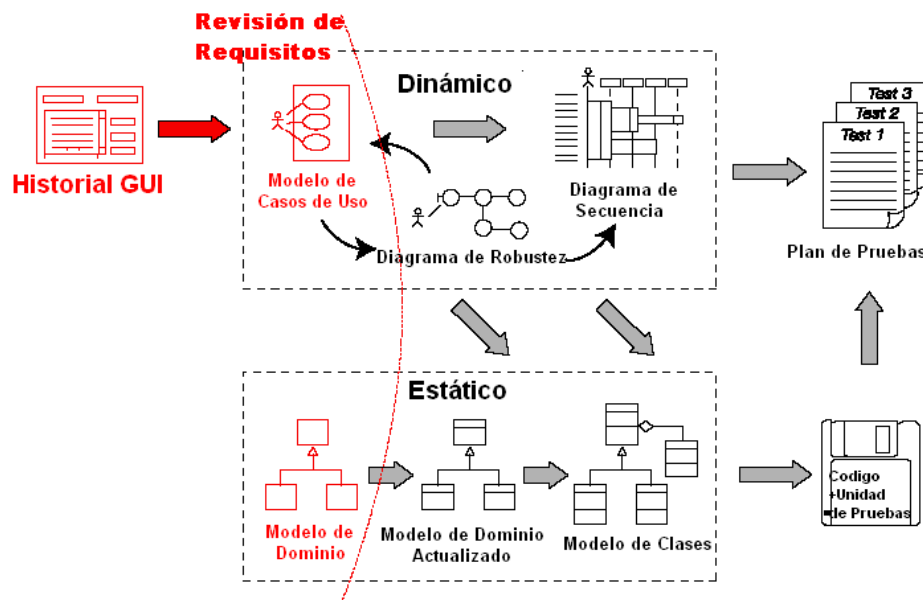
Figura N° 2.5: Modelo de casos de uso, adaptado. (Doug Rosenberg and Matt Stephens; 2007)



c. REVISIÓN DE REQUISITOS

La sesión de revisión de requisitos garantiza que el sistema tal y como se describe coincide con los requisitos. Se trata de un período de sesiones de colaboración que impliquen al representante(s) del cliente, los usuarios finales (es decir, las personas que realmente van a utilizar el sistema, o quien está usando el sistema actual que se sustituirá), y las personas de marketing-básicamente, todos los stakeholders que tienen un interés en asegurar que los requisitos encajen con su punto de vista del sistema. (Rosenberg y Stephens, 2007).

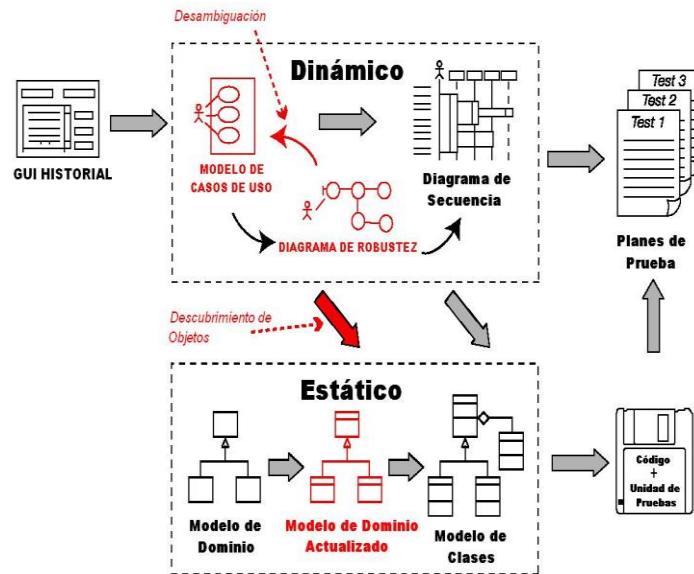
Figura N° 2.6: Revisión de requisitos, adaptado. (Doug Rosenberg and Matt Stephens; 2007)



d. ANÁLISIS DE ROBUSTEZ

Para obtener a partir de los casos de uso un diseño detallado (y luego el código), lo que se necesita es enlazar los casos de uso a los objetos. La técnica que se describe en este capítulo, el análisis de robustez, le ayuda a superar la brecha que existe del análisis al diseño. En pocas palabras, es una manera de analizar sus casos de uso e identificar un primer conjunto de objetos para cada caso de uso. Estos se clasifican en objetos interfaz, objetos entidad, y controladores (que son a menudo más como funciones que como objetos). (Rosenberg y Stephens, 2007).

Figura N° 2.7: Análisis de Robustez, adaptado. (Doug Rosenberg and Matt Stephens; 2007)

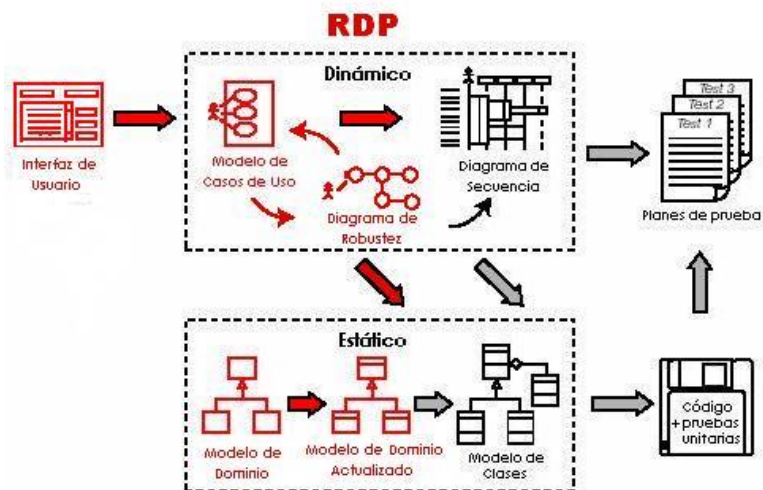


e. REVISIÓN PRELIMINAR DEL DISEÑO

Rosenberg y Stephens (2007) indica que:

Las sesiones de Revisión del Diseño Preliminar (RDP) ayudan a asegurarse que los diagramas de robustez, el modelo de dominio, y la descripción de casos de uso coincidan entre sí. Esta revisión es el "puente" entre el diseño preliminar y en las etapas del diseño detallado, para cada paquete de casos de uso.

Figura N° 2.8: Revisión Preliminar del diseño, adaptado. (Doug Rosenberg and Matt Stephens; 2007)

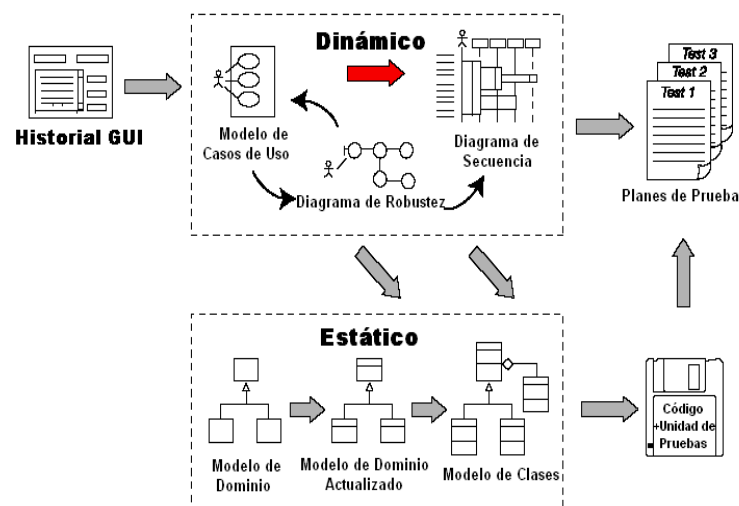


f. ARQUITECTURA TÉCNICA

El objetivo de arquitectura técnica (AT) es obtener un sentido general del sistema que vas a desarrollar. ¿Será un sistema basado en Internet? O un sistema en VB, NET o Java Swing, para un cliente muy rico? Es necesario utilizar un framework de aplicación específico (por ejemplo, un framework de una compañía estándar).

No hay una notación estándar o un formato para documentar la AT, la profundidad y el formato de la arquitectura técnica-y los convenios para crearla-varían mucho de empresa a empresa, por lo que no insistiremos en esta área demasiado tiempo. (Rosenberg y Stephens, 2007).

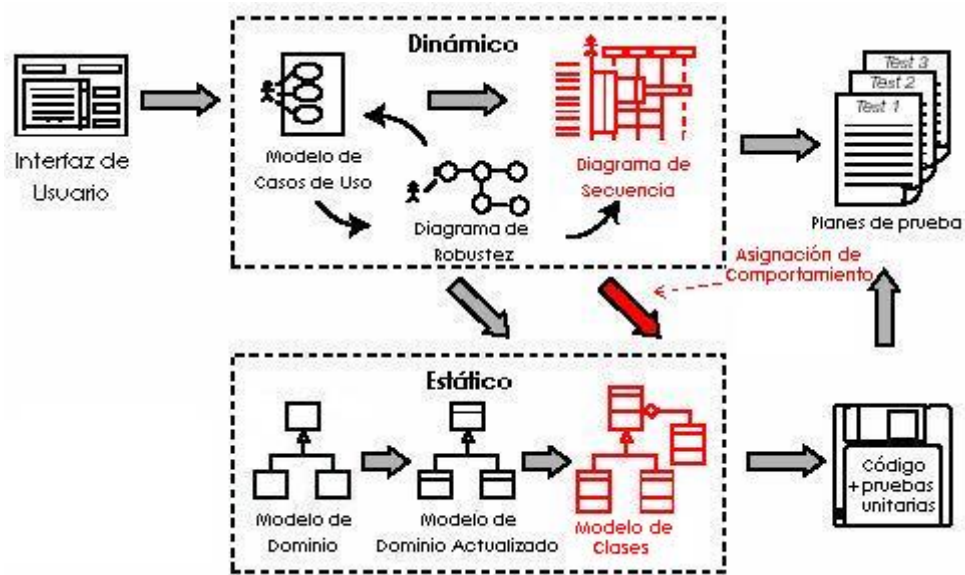
Figura N° 2.9: arquitectura técnica, adaptado. (Doug Rosenberg and Matt Stephens; 2007)



g. DIAGRAMAS DE SECUENCIA

Una vez que se ha finalizado el análisis de robustez, y ha celebrado una Revisión del Diseño Preliminar, es tiempo de iniciar el diseño detallado. En este momento, la descripción de sus casos de uso debe ser completa, correcta, detallada y explícita. En resumen, los casos de uso deben estar en un estado del que se pueda crear un diseño detallado. (Rosenberg y Stephens, 2007).

Figura N° 2.10: diagrama de secuencia, adaptado. (Doug Rosenberg and Matt Stephens; 2007)

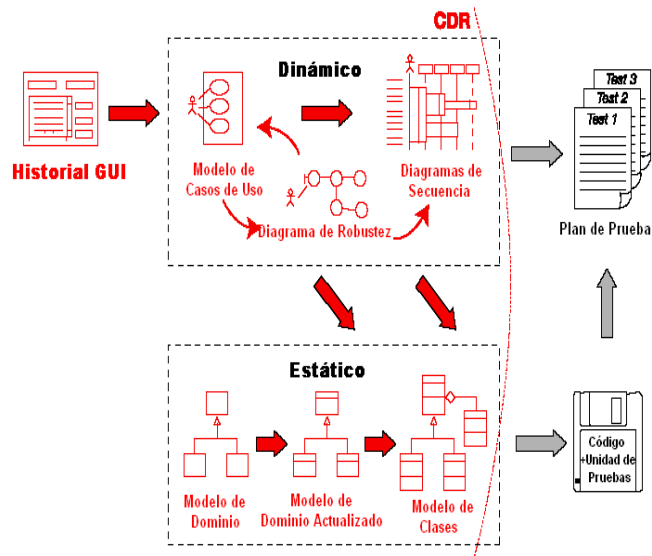


h. REVISIÓN CRÍTICA DEL DISEÑO

Tu proyecto debería estar ahora en mucha mejor condición que muchos otros proyectos que están en esta etapa. Por ahora, has utilizado el análisis de robustez para desambiguar la descripción de los casos de uso y descubrir las clases de dominio faltantes, has mantenido una revisión de diseño preliminar (PDR) para asegurarte que los casos de uso coinciden con lo que el cliente realmente quiere, y has elaborado cuidadosamente un diseño detallado de los casos de uso que has implementado para esta versión.

Por tanto, estás casi listo para comenzar la codificación-hay sólo una etapa rápida (pero vital) para comprobar el primero de la lista: Revisión del Diseño Crítico (CDR). (Rosenberg y Stephens, 2007)

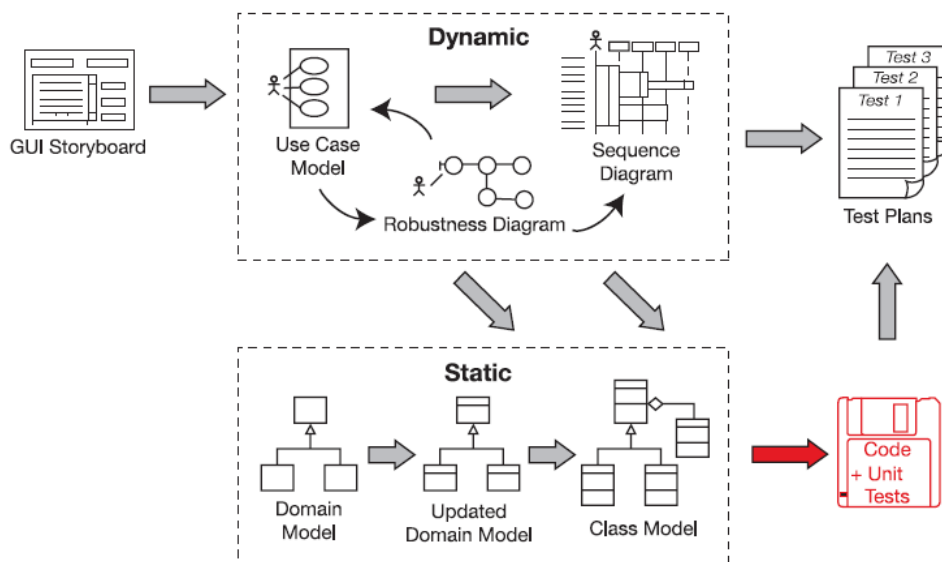
Figura N° 2.11: revisión Crítico del diseño, adaptado. (Doug Rosenberg and Matt Stephens; 2007)



i. IMPLEMENTACIÓN: PASO DEL DISEÑO DETALLADO AL CÓDIGO

Si usted ha pasado por todo el esfuerzo para crear un diseño agradable y detallado, vale la pena tener una buena idea de cómo traducir ese diseño detallado en el código fuente y las pruebas unitarias, por supuesto. (Rosenberg y Stephens, 2007)

Figura N° 2.12: Implementación, adaptado. (Rosenberg y Stephens; 2007)

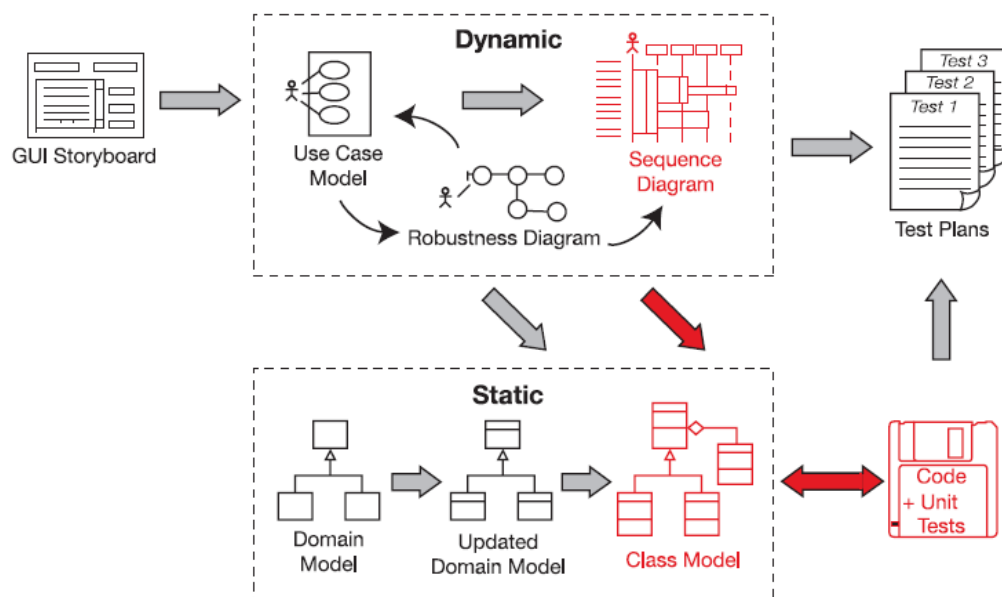


j. REVISIÓN DEL CÓDIGO Y ACTUALIZACIÓN DEL MODELO

Durante la codificación, lo más probable es que te han hecho algunos cambios en el diseño, por lo que el código ahora estar ligeramente fuera de sincronía con los diagramas de diseño. Una reacción tristemente común en esta etapa consiste en considerar que la documentación de diseño obsoleto, tirar a la basura, y seguir todas las posteriores desarrollo del trabajo sin hacer ningún trabajo de diseño más. (Rosenberg y Stephens, 2007)

Figura N° 2.13: Revisión del código y actualización del modelo, adaptado.

(Rosenberg y Stephens; 2007)

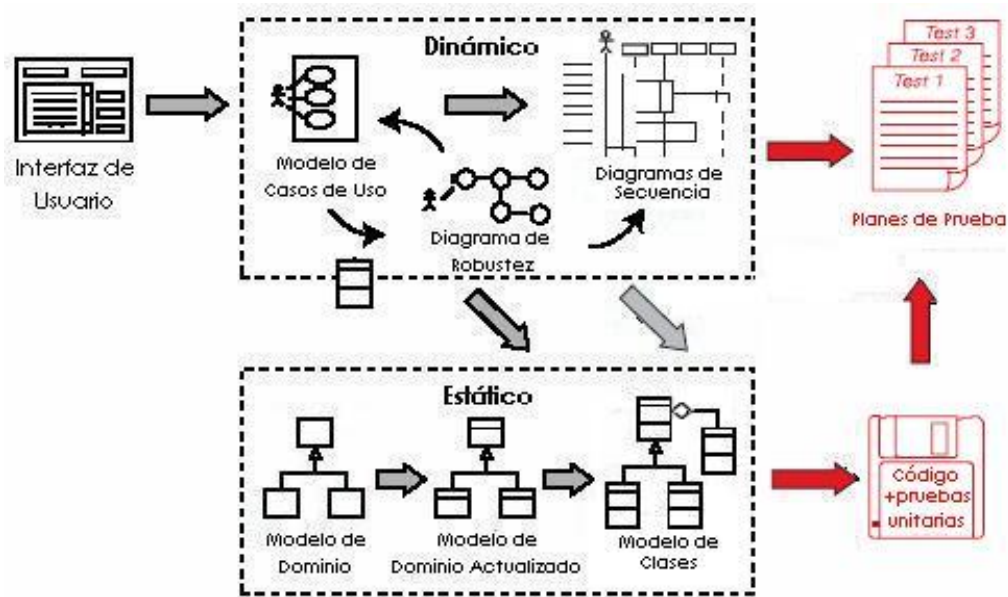


k. DISEÑO GUIADO POR PRUEBAS

Es fácil ver el módulo de un programa y decir; “Bien, ya terminé”, pero este sentido de completación puede decepcionarnos. ¿Cómo estás seguro que el código reúne todos los escenarios de casos de uso –no solo los cursos básicos, sino también los cursos alternos?

Las *Prueba basadas en el diseño* (DDT, por sus siglas en inglés) provee un método ‘a prueba de balas’ para producir casos de prueba y verificar que todos los escenarios específicos están completos. Puedes además usar este proceso para escribir pruebas unitarias ejecutables de estos casos de prueba. (Rosenberg y Stephens, 2007)

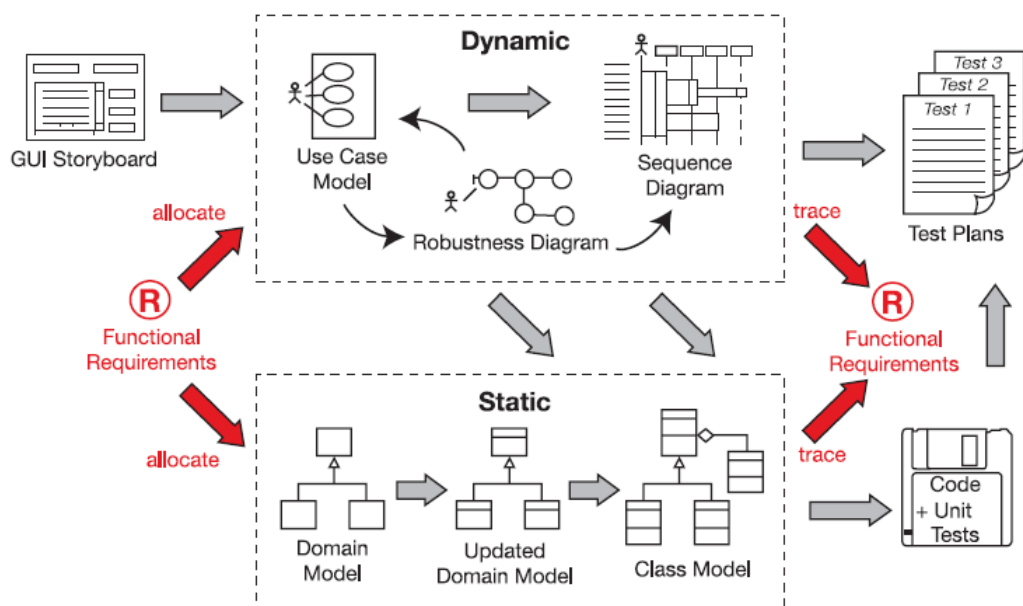
Figura N° 2.14: Diseño de pruebas, adaptado. (Rosenberg y Stephens; 2007)



I. ATENDIENDO REQUISITOS

Rosenberg y Stephens (2007) no es una parte fundamental del proceso de simplemente porque las diferentes organizaciones tienen diferentes estrategias para el manejo de los requisitos.

Figura N° 2.15: atención de requisitos, adaptado. (Rosenberg y Stephens; 2007)



2.2.9. LENGUAJE DE PROGRAMACIÓN C#

“Es un lenguaje orientado a objetos creado por Microsoft especialmente para su plataforma .NET. C# combina y mejora gran parte de las características más interesantes de Java y C++” (Cerezo, Peñalba y Caballero, 2006, p. 25)

“Creación de Microsoft, es un lenguaje de programación orientado a objetos que reúne las características de alta productividad y sencillez.” (Ceballos, 2006, p. 49)

“Los programas C# se organizan en clases, lo que significa que incluso el programa más sencillo que se pueda pensar debe estar escrito dentro de una clase.” (Martín, 2007, p. 11)

A. DATOS

“Un tipo de datos es el conjunto de valores que puede tomar una variable” (Joyanes y Zahonero, 2002, p. 122)

El lenguaje C# soporta dos tipos de variables de datos: por valor y por referencia. Los tipos por valor, llamados también tipos básicos o simples, se caracterizan por el hecho de que las variables que los almacenan contienen el valor del dato (Martín, 2007, p. 15)

B. VARIABLES

Una variable es una posición con nombre en memoria donde se almacena un valor de un cierto tipo de dato. Las variables de tipos de datos básicos, tipos primitivos, almacenan datos correspondientes con el tipo. Las variables de clases, variables de objetos, son referencias al objeto. Una constante por el contrario, es una variable cuyo valor no puede ser modificado. (Joyanes y Zahonero, 2002, p. 132)

“Las variables se utilizan para manejar los datos en un programa, tanto de los tipos simple como los objetos de la clase” (Martín, 2007, p. 17)

C. OPERADORES

C# proporciona numerosos operadores para tratar con los datos en un programa, la tabla 2.1 contiene los operadores más importantes:

Tabla N° 2.1: Operadores más importantes del lenguaje C#

TIPOS	OPERADORES
Aritméticos	+, -, *, /, %, ++ y --
Asignación	=, +=, -=, *=, /= y %=
Comparación	==, <, >, <=, >=, y !=
Lógicos	&&, , y !
Otros	New, is, typeof

D. INSTRUCCIONES DE CONTROL

D.1. INSTRUCCIÓN IF

“Establece el bloque de instrucciones que se deben ejecutar en función del resultado de la condición, cuyos únicos valores posibles pueden ser true o false. Su formato es:

```
if(){
    //instrucciones”
}
else{
    //instrucciones
}”
```

(Martín, 2007, p. 27)

D.2. INSTRUCCIÓN SWITCH

Permite establecer diferentes bloques de instrucciones a ejecutar, en función de los posibles valores que puedan resultar al evaluar una determinada expresión, valores que deben pertenecer a algunos de los tipos simples soportados por el lenguaje C#. Su formato es:

```
switch (){
    case valor1:
        //instrucciones
```

```

        break;
    case valor2:
        //instrucciones
        break;
    default:
        //instrucciones
        break;
}

```

(Martín, 2007, p. 28)

D.3. INSTRUCCIÓN FOR

“La instrucción for nos permite repetir la ejecución de un bloque de sentencias un número determinado de veces. Su formato es:

```

for(inicialización; comparación; incremento){
    //instrucciones
}”

```

(Martín, 2007, p. 30)

E. ARRAYS

Un array es un conjunto de datos de un mismo tipo, agrupados bajo un único nombre de variable. Cada uno de estos datos tiene asociada una posición, pudiendo acceder a cada uno de ellos utilizando la variable y un índice numérico que represente su posición. (Antonio Martín, 2007, p. 34)

2.2.10. PROGRAMACIÓN ORIENTADA A OBJETOS CON C#

Es una forma de pensar, una filosofía, de la cual surge una cultura nueva que incorpora técnicas y metodologías diferentes. Pero estas técnicas y metodologías, y la cultura misma, provienen del paradigma. La POO como paradigma es una postura ontológica: el universo computacional está poblado por objetos, cada uno responsabilizándose por sí mismo, y comunicándose con los demás por medio de mensajes. (Greiff, 1994, p. 31)

Álvarez (2008), afirma que la programación orientada a objetos es una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real que otros tipos de programación.

“La programación orientada a objetos es realmente un nuevo estilo de programación, el cual, básicamente consiste en definir clases y poner dichas clases a comunicarse o conversar entre sí.” (Flores, 2005, p. 27).

Los principios del modelo Orientado a Objetos son: abstracción, encapsulación, modularidad y jerarquía, Según Booch (1986) dice que si un modelo que se dice OO no contiene alguno de los cuatro elementos, entonces no es OO.

A. OBJETOS

“Un objeto es una encapsulación abstracta de información, junto con los métodos o procedimientos para manipularla. Un objeto contiene operaciones que definen su comportamiento y variables que definen su estado entre las llamadas a las operaciones.” (Florez, 2005, p. 49).

“Un objeto es una instancia (un ejemplar, un caso concreto) de una clase (una categoría)” (Joyanes y Zahonero, 2002, p. 77)

Booch (1995) define un objeto como “algo que tiene un estado, un comportamiento y una identidad” (p. 56)

B. CLASES

“En programación, una clase es una estructura que contiene datos y procedimientos (o funciones) que son capaces de operar sobre esos datos” (Joyanes, 2003, p. 591)

Según Álvarez (2008), una clase es mostrada como un rectángulo con tres partes:

- Nombre de la clase en la parte superior.
- Lista de atributos en la parte central.
- Lista de métodos en la parte inferior.

“Una clase es por tanto una plantilla implementada en software que describe un conjunto de objetos con atributos y comportamiento similares” (Álvarez, 2008, p. 64).

“Las clases contienen la definición de las propiedades y métodos que los objetos de la misma tendrán. En C# una clase se crea de la siguiente forma:

```
[public] class Nombre_clase
{
    //definición de propiedades
    //definición de métodos
}
```

(Según Martín, 2007, p. 40)

C. MÉTODOS

Los métodos (operaciones o servicios) describen el comportamiento asociado a un objeto. Representan las acciones que pueden realizarse por un objeto y sobre un objeto. La ejecución de un método puede conducir a cambiar el estado del objeto o dato local de un objeto. (Joyanes, 2003, p. 590)

Un método es un bloque de instrucciones encaminado a realizar una tarea determinada. Los métodos a diferencia de las propiedades no almacenan información sino que ejecutan las instrucciones asociadas cuando son invocados, para lo cual se usa la sintaxis: `variable_objeto.metodo()`; (Martín, 2007, p. 45)

2.2.11. PLATAFORMA .NET

.NET Framework es un entorno de ejecución administrado que proporciona diversos servicios a las aplicaciones en ejecución. Consta de dos componentes principales: Common Language Runtime (CLR), que es el motor de ejecución que controla las aplicaciones en ejecución, y la biblioteca de clases de .NET Framework, que proporciona una biblioteca de código probado y reutilizable al que pueden llamar los desarrolladores desde sus propias

aplicaciones. (MICROSOFT, 2016).

Figura N° 2.16: Framework .NET – MICROSOFT (2016)



Microsoft (2016) define a .NET como un modelo de desarrollo que hace que el software sea independiente de la plataforma y de los dispositivos, y que hace que los datos estén disponibles a través de Internet. Por lo consiguiente, el .NET Framework es la infraestructura básica subyacente de .NET.

Microsoft asegura que .NET fue implementado desde un principio pensando en una arquitectura abierta. .NET es una plataforma que puede utilizarse para generar y ejecutar la siguiente generación de aplicaciones de escritorio y aplicaciones Web. El objetivo de la plataforma .NET de Microsoft es simplificar el desarrollo Web. Está formada por las siguientes tecnologías principales:

- .NET Framework
- Windows Server System

- Servicios *building block* fundamentales
- Visual Studio .NET

Según Martín, (2007) define a .Net como el conjunto de tecnologías Microsoft destinadas al desarrollo de aplicaciones informáticas. Estas tecnologías se basan en la utilización de un modelo de desarrollo unificado, lo que significa que podrán ser utilizadas tanto para el desarrollo de aplicaciones que se ejecuten en entorno Windows, como para las que se vayan a ejecutar a entorno Web o incluso para aquellas cuyo entorno de ejecución se encuentra en un dispositivo electrónico.

2.2.12. PROGRAMACIÓN CON ASP.NET

“Debido a la potencia y cortos tiempos de desarrollo que ofrece ASP.NET, se trata de una tecnología por la que están optando cada vez más programadores y empresas software para acometer sus desarrollos.” (Martín, 2007, p. 107).

A. APLICACIÓN WEB

Se entiende por aplicaciones web que:

Son un conjunto de archivos de hipertexto ligados que presenta información en texto y algunas gráficas. Sin embargo, a medida que el comercio electrónico y las aplicaciones B2B adquieren mayor importancia, las WebApps evolucionan hacia ambientes computacionales sofisticados que no solo proporcionan características, funciones de cómputo y contenidos independientes del usuario final, sino que están integradas con base de datos corporativas y aplicaciones de negocio. (Pressman, 2005, p. 501)

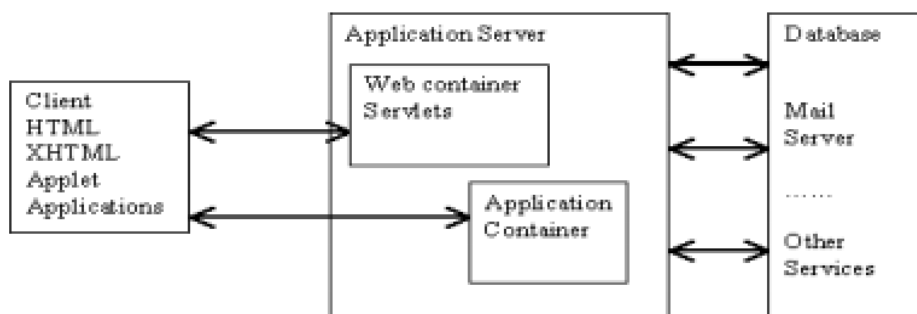
“La creación de aplicaciones web, en consecuencia, requiere la existencia de software ejecutándose en el servidor que genere automáticamente los ficheros HTML que se visualizan en el navegador del usuario.” (Berzal, Cortijo y Cubero, 2003, p. 9)

Una aplicación web es un programa informático que puede dar servicio simultáneamente a múltiples usuarios que lo ejecutan a través de internet. Este tipo de aplicaciones se basa en lo que se conoce como una arquitectura de 3 capas, donde los diferentes actores y elementos implicados en la misma se encuentran distribuidos en tres bloques o tres capas. (Martín, 2007, p. 107)

B. PATRÓN DE ARQUITECTURA MVC

“La arquitectura cliente/servidor a dos-capas es el punto de partida para la arquitectura cliente/servidor a “n-capas” la cual se caracteriza por contener en el servidor de aplicaciones un contenedor Web, así como un contenedor de aplicaciones.” (Allamaraju, 2002, p. 50).

Figura N° 2.17: Arquitectura cliente/servidor de n-capas



En el caso de los sistemas de información, una de las arquitecturas más utilizadas es la de Modelo, Vista, Controlador (MVC – Model, View, Control), popularizado por los ambientes de desarrollo para los lenguajes de programación Smalltalk esta arquitectura se basa en tres dimensiones principales: Modelo Correspondiente a la *Información*, Vista correspondiente a la presentación o interacción con el *usuario*, y Control correspondiente al comportamiento. (Weitzenfeld, 2002, p.254).

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC (según CMU), se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

B.1. MODELO

Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la *vista* y su *controlador* facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.

“La capa de datos tiene como misión el almacenamiento permanente de la información manejada por la aplicación y la gestión de la seguridad de los mismos” (Martín, 2007, p. 112)

B.2. VISTA

Esta capa presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Según Antonio Martín (2007) se trata de la capa con la que interactúa el usuario de la aplicación. Realiza principalmente dos funciones. Por un lado se encarga de capturar los datos de usuario con los que opera la aplicación y enviárselos a ésta. La segunda función es la de presentar al usuario los resultados generados por la aplicación.

B.2.1. NAVEGADOR

En una arquitectura web de 3 capas la capa cliente está implementada por el navegador. Un navegador es un tipo especial de software basado en una interfaz gráfica que utilizan los usuarios para interactuar con una aplicación web, tanto en un entorno Internet como Intranet. (Martín, 2007, p. 109)

B.2.2. PÁGINA WEB

Una página web es un documento de texto que contiene información para ser visualizada por un navegador; es enviada a éste desde la capa intermedia y está codificada en un lenguaje especial capaz de ser interpretado por todos los tipos de navegadores. Este lenguaje es el HTML. (Martín, 2007, p. 109)

B.2.3. INTERNET

“Internet es un inmenso conjunto de redes diferentes que usan ciertos protocolos comunes y proporcionan ciertos servicios comunes.” (Tanenbaum, 2003, p. 50).

Internet ha supuesto una revolución sin precedentes en el mundo de la informática y de las comunicaciones. Los inventos del telégrafo, teléfono, radio y ordenador sentaron las bases para esta integración de capacidades nunca antes vivida. Internet es a la vez una oportunidad de difusión mundial, un mecanismo de propagación de la información y un medio de colaboración e interacción entre los individuos y sus ordenadores independientemente de su localización geográfica. (Berners-Lee, 2010, p. 55).

B.2.4. WEB

World Wide Web (o la "Web") o Red Global Mundiales un sistema de documentos de hipertexto y/o hipermedias enlazados y accesibles a través de Internet. Con un navegador Web, un usuario visualiza páginas Web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces. (Berners-Lee, 2010, p. 94)

B.2.5. PROTOCOLOS USADOS EN INTERNET

Los navegadores, y en general cualquier cliente web, se comunican con la capa intermedia mediante un protocolo conocido como HHTP. Utilizando este protocolo, el navegador envía a la capa intermedia los datos suministrados por el usuario para la ejecución de la aplicación y esta a su vez lo emplea para enviar al navegador la información que debe presentar en pantalla. (Martín, 2007, p. 110)

B.3. CONTROLADOR

Esta capa responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

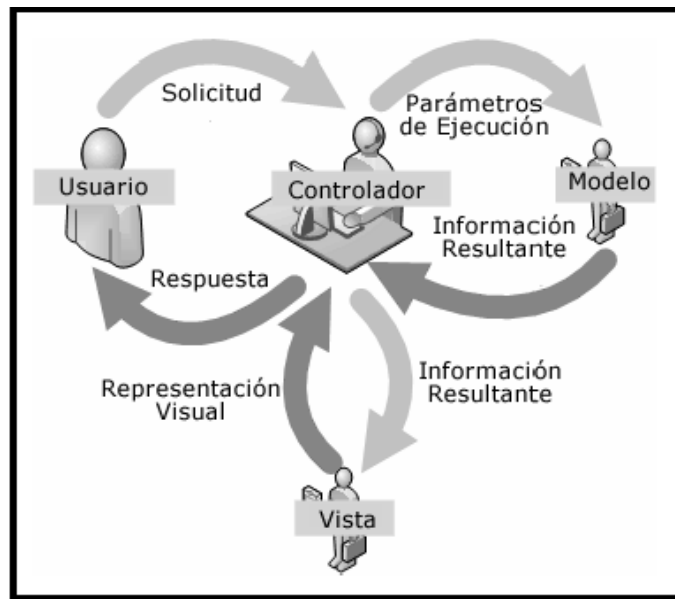
“En una arquitectura de tres capas la capa intermedia está constituida por la aplicación en sí. Esta se encuentra instalada en una maquina independiente, conocida como servidor, a la que acceden los clientes a través de la red.” (Martín, 2007, p. 111)

“Son numerosos las distintas tecnologías y lenguajes proporcionados por los fabricantes software para la construcción de aplicaciones de capa intermedia en una arquitectura de tres capas.” (Martín, 2007, p. 112)

B.4. CICLO DE VIDA DE MVC

El ciclo de vida de MVC es normalmente representado por las 3 capas presentadas anteriormente y el cliente (usuario). La siguiente figura representa el ciclo de vida de manera sencilla.

Figura N° 2.18: Ciclo de vida MVC



El primer paso en el ciclo de vida empieza cuando el usuario hace una solicitud al controlador con información sobre lo que el usuario desea realizar. Entonces el Controlador decide a quién debe delegar la tarea y es aquí donde el Modelo empieza su trabajo. En esta etapa, el Modelo se encarga de realizar operaciones sobre la información que maneja para cumplir con lo que le solicita el controlador. Una vez que termina su labor, le regresa al controlador la información resultante de sus operaciones, el cual a su vez redirige a la Vista. La Vista se encarga de transformar los datos en información visualmente entendible para el usuario. Finalmente, la representación gráfica es transmitida de regreso al Controlador y éste se encarga de transmitírsela al usuario. El ciclo entero puede empezar nuevamente si el usuario lo requiere.

C. COMPONENTES DE UNA APLICACIÓN ASP.NET

Antonio Martín (2007) afirma que:

Para construir una aplicación ASP.NET, la plataforma .NET proporciona una serie de elementos y componentes que nos ofrecen gran potencia y facilidad a la hora de acometer los desarrollos. Estos son:

- **Formularios web.** Representan las páginas web de la aplicación a través de las cuales se producirá la interacción con el usuario
- **Controles web y controles HTML.** Son los elementos gráficos que permiten realizar el diseño de la interfaz de usuario.

- **Objetos ASP.** Permiten resolver los diferentes aspectos de la problemática que se presenta en la programación de aplicaciones Web, tales como el mantenimiento del estado, el acceso a los datos del cliente, etc.

D. PAGINAS ASPX

Desde un punto de vista global, una aplicación ASP.NET se compone de una serie de módulos interrelacionados conocidos como páginas aspx. Estos módulos contienen la parte visual como el código de la aplicación”.

Cada página aspx está compuesta por dos partes:

- **El formulario Web.** Es un archivo de texto con extensión .aspx que contiene la interfaz gráfica de la aplicación. Aparentemente, es similar a una página HTML aunque además de etiquetas y controles HTML pueden contener controles Web.
- **El archivo de código asociado.** En este archivo se incluirá la parte del código de la aplicación asociado a la página aspx que deba ser ejecutado en el servidor. Dependiendo del lenguaje utilizado para la creación de la aplicación, su extensión será .aspx.cs para C# o aspx.vb para Visual Basic. (Martín, 2007, pp. 117-118)

E. INCLUSIÓN DE CÓDIGO DE SERVIDOR EN EL FORMULARIO WEB

A estos bloques de código se les conoce como scripts de servidor y pueden aparecer en cualquier parte de la página. Alguno de éstos bloques de código o scripts tienen como misión el insertar el resultado de una excepción o variable del programa en la página de respuesta, por lo que son conocidos como expresiones ASP y siguen el formato:

`<%=expresión%>` (Martín, 2007, p. 121)

F. COMPONENTES DE LA INTERFAZ DE USUARIO

“Las aplicaciones ASP.NET no solamente permiten utilizar controles Web para la construcción de la interfaz de usuario; también es posible emplear controles HTML clásicos.” (Antonio Martín, 2007, p. 144)

G. OBJETOS ASP

Los controles Web proporcionan un potente mecanismo para el desarrollo de la interfaz gráfica de una aplicación web, facilitando la captura y presentación de datos en la misma.

No obstante, las aplicaciones web también necesitan realizar otras operaciones como es el mantenimiento del estado de las aplicaciones, el acceso a la información de la petición HTTP, la gestión de las respuestas HTTP, etc. Para ello, ASP.NET proporciona una serie de clases cuyos métodos y propiedades permiten realizar todas estas operaciones.

Estas clases son instanciadas con cada petición y almacenadas en propiedades del objeto Page, exponiéndose al programador como objetos intrínsecos a la aplicación. Estos objetos son conocidos como objetos ASP. Las características principales de estos objetos son:

- **Request.** Proporciona acceso a la petición actual. A través de las propiedades de este objeto pueden recuperarse los datos enviados por el usuario, acceder a los encabezados de la petición o recuperar las cookies del cliente.
- **Response.** El objeto Response está asociado al stream de salida de la página, pudiéndose utilizar para escribir texto en la respuesta que va a ser enviada al cliente o para insertar una cookie en la misma.
- **Application.** Permite el almacenamiento de la información compartida por todos los usuarios de la aplicación. (Martín, 2007, pp. 149-150)

H. DATOS INSERTADOS EN LA URL

No toda la información suministrada a la aplicación Web desde una página cliente es proporcionada a través de controles Web/HTML de usuario. Estos datos deben ser insertados directamente en las URLs de los enlaces o

formularios que apuntan a la página aspx que debe escribirlos en la petición, siguiendo el formato:

`Nombre_pagina.aspx?nombre_dato=valor&nombre_dato=valor`

Para recoger esta información desde las clases Page de la página solicitada, volvemos a recurrir al objeto Request, sólo que ahora debemos emplear su propiedad QueryString en vez de Form. QueryString contiene la colección de datos enviados en la URL de la petición, disponiendo de un indizador para acceder de forma individualizada a los valores de los mismos a partir de su nombre:

`this.Request.QueryString["nombre_dato"]` (Martín, 2007, p. 161)

I. REDIRECCIONAMIENTO

La transferencia de una petición desde una página a otra solo puede realizarse si ambas página pertenecen a la misma aplicación ASP.NET (Web Site). Si la página aspx que recibe la petición tiene que pasar el control a otra página (aspx, htm o asp) de una aplicación diferente, debería realizar un direccionamiento.

Como su nombre indica, un redireccionamiento consiste en redireccionar al usuario a otra URL diferente a la que esta ha solicitado. Esta operación se lleva a cabo mediante el método Redirect() del objeto Response de ASP y su formato es el siguiente: void Redirect(String url)(Martín, 2007,p. 172)

J. OBJETO SESSION Y VARIABLES DE SESIÓN

“La sesión es un concepto asociado al uso que un determinado cliente realiza de una aplicación Web” (Martín, 2007, p. 221)

El objeto Session de ASP permite gestionar las sesiones de usuarios durante la ejecución de una aplicación ASP.NET, su principal funcionalidad es la de proporcionar acceso a las variables de sesión.

Las variables de sesión son espacios de memoria del servidor, donde una aplicación puede almacenar datos que estén accesibles para todas las instancias de las páginas a las que un determinado usuario accede durante la sesión.

Desde cualquiera de las páginas aspx de la aplicación, se puede acceder a la colección de variables de sesión del usuario actual a través del indizador del objeto Session:

```
this.Session["nombre_variable"] (Martín, 2007, p. 222)
```

K. CONTROLES DE VALIDACIÓN

Los controles de validación permiten a los programadores ahorrar una gran cantidad de tiempo y líneas de código en el desarrollo de una aplicación Web, ya que una de las tareas más pesadas y rutinarias que existen, como es el de validación de los datos de usuario antes de su procesamiento, puede ser realizada mediante este tipo de controles sin incluir una sola línea de código. (Martín, 2007, p. 202)

L. CONFIGURACIÓN DE APLICACIONES ASP.NET

“Además de clases y páginas aspx, una aplicación ASP.NET incluye un archivo de configuración en el que se definen distintos parámetros relativos a la misma y que el servidor de aplicaciones utiliza durante su ejecución.” (Martín, 2007, p. 182)

L.1. EL ARCHIVO Web.config

Este archivo de configuración consiste en un documento XML en el que, mediante una serie de etiquetas predefinidas, se establecen los valores de los distintos parámetros configurables de la aplicación. El nombre del archivo es Web.config y debe estar situado en el directorio raíz de la aplicación Web. (Martín, 2007, p. 182)

M. JQUERY

jQuery se encuentra escrito en JavaScript, un lenguaje de programación muy rico y expresivo.

El jQuery es una librería de JavaScript para acceder a los objetos del DOM de un modo simplificado. Las aplicaciones en internet son cada vez más complejas, ya que incorporan efectos visuales, drag and drop, auto-completar, animaciones etc. el desarrollar todos estos conceptos desde cero puede resultar complicado sobre todo si tenemos que presentar la solución con muy poco tiempo, en este tipo de situaciones el empleo de librerías como el jQuery nos facilitan el desarrollo de la aplicación. Otra ventaja paralela es despreocuparnos cuando codificamos en la compatibilidad de navegadores, ya que la librería resolverá esto.

Para utilizar la librería como dijimos debemos descargarla del sitio oficial y en cada página que lo requiera agregar:

```
<script type="text/javascript" src="jquery.js"></script>
```

Del sitio oficial de jQuery descargaremos la versión descomprimida que ocupa alrededor de 60 Kb (es el archivo jquery.js) y cuando subamos nuestro sitio es recomendable descargar la versión comprimida que tiene un peso de 20 Kb.

N. MASTER PAGES

Una Master Page es una página aspx singular, que funciona como una especie de plantilla para la aplicación. Su misión es definir la estructura que deben

tener todas las páginas de la misma, así como establecer los elementos comunes que todas ellas van a contener.

A través de las Master Pages podemos conseguir la funcionalidad proporcionada por los Frames, de una forma mucho más cómoda y sencilla y sin los inconvenientes que estos presentan. (Martín, 2007, p. 334)

O. HOJAS DE ESTILO CSS

Como solución al problema del estilo, el W3C desarrolló un estándar bajo el nombre de Cascading Style Sheet (CSS), que tiene como objetivo poder definir hojas de estilo para ser aplicadas sobre un documento HTML, permitiendo separar los datos de la información de formato y estilo de los mismos.

Las hojas de estilo CSS se basan en la definición de una serie de propiedades de estilo asociadas a determinados tipos de etiquetas e independientes de su contenido, de forma que al ser utilizadas estas etiquetas en el documento se apliquen automáticamente las diferentes propiedades y opciones de estilo definido para las mismas. (Martín, 2007, p. 406)

P. ACCESO A DATOS

La mayoría de las aplicaciones web trabajan con información almacenada en alguna fuente de datos, ya sea porque tienen que mostrar contenidos al usuario que han de actualizarse dinámicamente, porque tienen que realizar algún tipo de búsqueda de información o porque necesitan almacenar datos suministrados por el usuario a través de una página web.

Toda esta información, como decimos, no se guarda dentro de la aplicación, sino que reside en algún tipo de fuente de datos externa (normalmente una

base de datos) con la que esta debe interaccionar, todo ello de forma transparente para el usuario. Esta interacción entre una aplicación Web ASP.NET y un origen de datos es posible gracias a ADO.NET.

ADO.NET representa el modelo de objetos incluido dentro del .NET Framework que permite a una aplicación acceder a bases de datos relacionales y a otro tipo de fuentes de datos, como por ejemplo documentos XML. (Martín, 2007, p. 249)

P.1. FUNDAMENTOS DE ADO.NET

“ADO.NET proporciona a las aplicaciones dos modos de trabajar con un origen de datos: modo conectado y modo desconectado.” (Martín, 2007, p. 252)

P.1.1. ACCESO A DATOS CON ADO.NET EN MODO CONECTADO

“Cuando hablamos de acceso a datos en modo conectado, nos estamos refiriendo a acceder a los datos reales existentes en una base de datos a través de una conexión establecida con la misma.” (Martín, 2007, p. 252)

P.1.1.1. PROVEEDOR DE DATOS

“Representa el conjunto de clases específicamente creadas para trabajar con bases de datos de SQL Server, tanto en la versión 2000 como en la 2005. Estas clases se encuentran en el espacio de nombres System.Data.SqlClient.” (Martín, 2007, p. 253)

P.1.1.2. LAS CLASES ADO.NET EN MODO CONECTADO

Las características de las clases ADO.NET

- **Connection.** Permite establecer una conexión con la base de datos. Los objetos Connection representan conexiones activas con la base de datos y deben permanecer abiertas para poder transferir información entre ésta y la aplicación.

- **Command.** Los objetos Command permiten enviar consultas SQL a la base de datos a través de la conexión. También pueden ejecutar procedimientos almacenados de la base de datos.
- **Parameter.** Un objeto Parameter contiene información relativa a un parámetro que va a ser enviado a la base de datos durante la ejecución de un parámetro almacenado. (Martín, 2007, p. 255)

P.1.1.3. PROCESO DE ACCESO A LOS DATOS

Cuando una aplicación .NET necesita acceder a una base de datos utilizando el modo conectado de ADO.NET, según Antonio Martín (2007) debe seguir los siguientes pasos en el orden indicado:

- Establecimiento de una conexión.
- Ejecución de la consulta SQL.
- Manipulación de resultados (si procede)
- Cierre de la conexión

P.1.1.4. ESTABLECIMIENTO DE UNA CONEXIÓN

“Para poder tener acceso a la base de datos es necesario establecer una conexión con la misma, operación que se realiza a través de un objeto Connection.” (Martín, 2007, p. 256)

P.1.1.5. CREACIÓN DE UN OBJETO CONNECTION

En todos los proveedores de datos, el constructor de clase Connection requiere que se le pase como parámetro una cadena de caracteres con los datos de conexión a la base de datos.

Las cadenas de conexión varían de un proveedor a otro. Por ejemplo, para el proveedor de SQL Server la cadena de conexión tendría el siguiente aspecto:

“Data Source=servidor; Initial Catalog=basedatos; user=usuario; pwd=password:”

Los cuatro parámetros son:

- **Data Source:** nombre del servidor de la base de datos.
- **Initial Catalog.** Nombre de la base de datos.
- **user.** Nombre del usuario con quien va a establecer la conexión. Debe ser un usuario registrado en la base de datos.
- **pwd.** Contraseña del usuario. Si quiere utilizar la seguridad integrada con Windows para conectar con la base de datos, este parámetro y la anterior deberían sustituirse por la expresión Integrated Security=true o Integrated Security=SSPI. (Martín, 2007, pp. 256-257)

P.1.1.6. APERTURA DE LA CONEXIÓN

“Una vez creada el objeto Connection debemos proceder a abrir la conexión, para lo cual empleamos el método Open() del objeto” (Martín, 2007, p. 257)

P.1.1.7. EJECUCIÓN DE UNA CONSULTA SQL

A partir del momento en que se abre la conexión con la base de datos ya es posible ejecutar consultas SQL sobre ella.

En ADO.NET las instrucciones SQL y las llamadas a procedimientos almacenados son enviados desde la aplicación a la base de datos utilizando objetos de la clase Command. (Martín, 2007, p. 258)

P.1.1.8. CREACIÓN DE UN OBJETO COMMAND

“El constructor de la clase Command requiere dos parámetros. Por un lado, el objeto Connection abierto a través del cual se va a ejecutar la consulta y por otro, la cadena de caracteres con la consulta SQL.” (Martín, 2007, p. 258)

“La clase Command dispone de los siguientes métodos para ejecutar una consulta: ExecuteNonQuery(). Se utiliza para ejecutar consultas de acción que no devuelven resultados, es decir, del tipo Insert, Update, Delete.” (Martín, 2007, p. 259)

P.1.1.9. CIERRE DE UNA CONEXIÓN

“Una vez finalizadas las operaciones sobre la base de datos, a fin de liberar recursos en el servidor, debemos proceder al cierre de la conexión tan pronto sea posible. Esta operación se realiza invocando al método Close() del objeto Connection.” (Martín, 2007, p. 259)

P.1.2. ACCESO EN MODO DESCONECTADO

Durante el acceso a datos en modo desconectado, la aplicación trabaja con una copia de los datos en memoria sin necesidad de mantener ninguna conexión abierta con la base de datos mientras se produce la manipulación de los mismos.

Las copias de los datos son gestionados a través de un objeto DataSet que proporciona todo lo necesario para manipular los datos desde código. (Martín, 2007, p. 275)

P.1.2.1. LAS CLASES DE ADO.NET EN MODO DESCONECTADO

Las clases utilizadas por ADO.NET para trabajar en modo desconectado se encuentran en el espacio de nombres System.Data, las copias de datos son independientes del proveedor, por lo que se utilizaran las mismas clases para manipular cualquier tipo de origen de datos. (Martín, 2007, p. 258)

Para Antonio Martín (2007) las clases de ADO.NET en modo desconectado son:

- **DataSet.** Sin duda es la clase más importante de las clases ADO.NET para la manipulación de datos en modo desconectado. Los objetos DataSet son los

encargados de albergar las copias de los datos, proporcionando todo lo necesario para poder realizar la manipulación de los mismos.

- **DataTable.** Al igual que sucede con las bases de datos reales, la información almacenada en un DataSet se organiza en tablas, cada una de las cuales está representada por un objeto DataTable.
- **DataAdapter.** Los objetos DataAdapter son los encargados de generar las copias de los datos, así como de llevar a cabo la actualización de la base de datos con las modificaciones realizadas en las copias, al ser un objeto que hace de interfaz entre el DataSet y la base de datos, cada proveedor deberá proporcionar su propia clase DataAdapter.

P.1.2.2. CREACIÓN Y CONFIGURACIÓN DEL OBJETO DATAADAPTER

El objeto DataAdapter es el encargado de conectar con la base de datos, crear la copia de los registros adecuados y de almacenarla en el DataSet. Por ello, necesita que se le proporcione al constructor de la clase tanto la cadena de conexión de la base de datos como la consulta SQL de tipo Select que determina los datos a copiar. (Martín, 2007, p. 279)

P.1.2.3. GENERACIÓN DE LA COPIA

“Se invoca al método Fill() del objeto DataAdapter. El método Fill() está sobrecargado, presentando diversos formatos para la generación de la copia.” (Martín, 2007, p. 279)

P.1.3. PROCEDIMIENTOS ALMACENADOS

Un procedimiento almacenado es una rutina SQL que reside en la base de datos y que realiza una o varias operaciones sobre la misma.

Los procedimientos almacenados no solamente incluyen sentencias SQL, sino que también se pueden emplear para su construcción de variables e instrucciones de control. Cada gestor de base de datos proporciona su propio dialecto SQL que, además de las sentencias SQL estándar, consta de un

amplio abanico de instrucciones que permiten crear procedimientos almacenados de una gran complejidad. En el caso de SQL Server, el lenguaje utilizado para la creación de procedimientos almacenados se llama Transac-SQL. (Martín, 2007, p. 270)

P.1.3.1. EJECUCIÓN DE UN PROCEDIMIENTO ALMACENADO CON ADO.NET

Como sucede con las consultas SQL estándares, la ejecución de un procedimiento almacenado desde una aplicación ADO.NET se realiza a través del objeto Command.

Los pasos a seguir para la ejecución de un procedimiento almacenado son los siguientes:

1. Creación y configuración del objeto Command.
2. Establecimiento de parámetros.
3. Ejecución.

(Martín, 2007, p. 271)

P.1.3.1.1. CREACIÓN Y CONFIGURACIÓN DEL OBJETO COMMAND.

Para crear un objeto Command que permita ejecutar un procedimiento almacenado, se debe especificar en el constructor el nombre del procedimiento almacenado a ejecutar, en vez de una consulta SQL. El siguiente objeto Command se crea para ejecutar el procedimiento almacenado “actualizarPrecio”:

```
SqlCommand cmd = new SqlCommand("actualizarPrecio", cn)
```

Una vez creado el objeto Command, es necesario indicarle que la cadena especificada en el constructor debe ser tomada como el nombre de un procedimiento almacenado. Esto se hace a través de la propiedad

CommandType del objeto, a la que habrá que asignar el valor adecuado de los que se encuentran en la enumeración CommandType:

```
cmd.CommandType = CommandType.StoredProcedure
```

(Martín, 2007, pp. 271-272)

P.1.3.1.2. ESTABLECIMIENTO DE PARÁMETROS

Los procedimientos almacenados pueden aceptar parámetros. Estos deben ser creados como objetos Parameter en la aplicación ADO.NET y añadirlos a la colección de parámetros del objeto Command antes de proceder a la ejecución del procedimiento almacenado.

Por cada parámetro que se quiera pasar al procedimiento almacenado habrá que crear un objeto Parameter correspondiente. (Martín, 2007, p. 272)

P.1.3.1.3. EJECUCIÓN

“Para ejecutar el procedimiento almacenado utilizaremos los métodos del objeto Command como el ExecuteNonQuery()” (Martín, 2007, p. 273)

P.1.4. CONTROL GRIDVIEW

El control GridView representa una versión mejorada del control DataGrid incluida en las versiones anteriores a ASP.NET 2.0. Básicamente este control permite visualizar en una página web el contenido completo del DataTable con el que está vinculado, aunque también proporciona los medios necesarios para que el usuario pueda modificar, añadir y eliminar registros directamente sobre el control. (Martín, 2007, p. 293)

2.2.13. LENGUAJE SQL

“El lenguaje SQL surge ante la necesidad de disponer de un mecanismo para operar con la información almacenada en base de datos relacionales de diferentes fabricantes.”
(Martín, 2007, p. 411)

A. SENTENCIAS PARA MANIPULACIÓN DE DATOS (DML)

A.1. SENTENCIA SELECT

La sentencia SELECT es la más importante y la más compleja de todas las sentencias que forman SQL. Se utiliza para extraer datos de una o varias tablas de la base de datos. Su forma general operando sobre una única tabla es:

```
SELECT campo1, campo2,...  
  
FROM tabla  
  
[WHERE criterios]  
  
[ORDER BY campo]           (Martín, 2007, pp. 412-413)
```

A.1.1. ORDENACIÓN DE REGISTROS

“La cláusula ORDER BY determina cómo se van a ordenar los registros según la forma:

```
Campo1 ASC/DESC, campo2 ASC/DESC,...”   (Martín, 2007, p. 414)
```

A.1.2. CONSULTA SELECT SOBRE VARIAS TABLAS

Para obtener datos incluidos en distintas tablas basta con indicar los nombres de estas en la cláusula FROM, separadas por comas. Si alguno de los campos seleccionados se encuentra en más de una tabla, hay que preceder a su nombre el de la tabla en la que se encuentra, separado por un punto. (Martín, 2007, p. 414)

A.1.3. OPERADORES

Además de los operadores simples (<, >, =, ...) la cláusula WHERE puede incluir otros tipos de operadores:

- **LIKE.** Se utiliza para buscar campos que contengan combinaciones de caracteres que cumplan ciertas condiciones:

Campo LIKE constante_alfanum

La constante alfanumérica puede contener caracteres cualesquiera y puede incluir comodines con significado especial:

% cadena de longitud aleatoria

- **BETWEEN.** comprueba si un valor está comprendido entre dos datos:

Exp1 [NOT] BETWEEN exp2 AND exp3 (Martín, 2007, pp. 415-416)

A.2. SENTENCIA INSERT

Permite añadir una o más filas a una tabla. Puede utilizarse tanto para añadir una única fila como para copiar en una tabla un subconjunto de registros proveniente de otra. En el caso de añadir filas individuales se utiliza el formato:

INSERT INTO tabla (campo1, campo2,...)

VALUES (valor1, valor2,...) (Martín, 2007, p. 416)

A.3. SENTENCIA UPDATE

Su función es modificar los valores de ciertos campos en aquellos registros que cumplan una determinada condición. El formato de esta instrucción es el siguiente:

UPDATE tabla

SET campo1=expr1, campo2=expr2,...

WHERE condiciones (Martín, 2007, p. 418)

2.2.14. SISTEMA GESTOR DE BASE DE DATOS

Los sistemas de gestión de bases de datos (en inglés DATABASE MANAGEMENT SYSTEM, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Los hechos general de los sistemas de gestor de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización (PostgreSQL Global Development Group, s.f.).

De acuerdo a Capote (2008) el objetivo de un SGBD es garantizar la independencia de sus datos. Otros objetivos que deben satisfacer los SGBD son:

2.2.14.1. Diseño y utilización orientada al usuario

Los datos y aplicaciones deben ser accesibles a los usuarios de la manera más amigable posible. Para ello los SGBD, habrán de fundamentarse en un modelo de datos teórico coherente y proporcionar lenguajes que permitan definir la estructura de la BD y acometer la generación, mantenimiento y acceso a los datos en los términos que resulten apropiados.

2.2.14.2. Centralización

Los datos deben gestionarse de forma centralizada e independiente de las aplicaciones. Para satisfacer este objetivo un SGBD habrá de proporcionar una serie de utilidades que faciliten la administración de ese fondo común. Adicionalmente, aparece la figura del administrador de BD, como responsable de las tareas que permiten mantener la disponibilidad de ese fondo común.

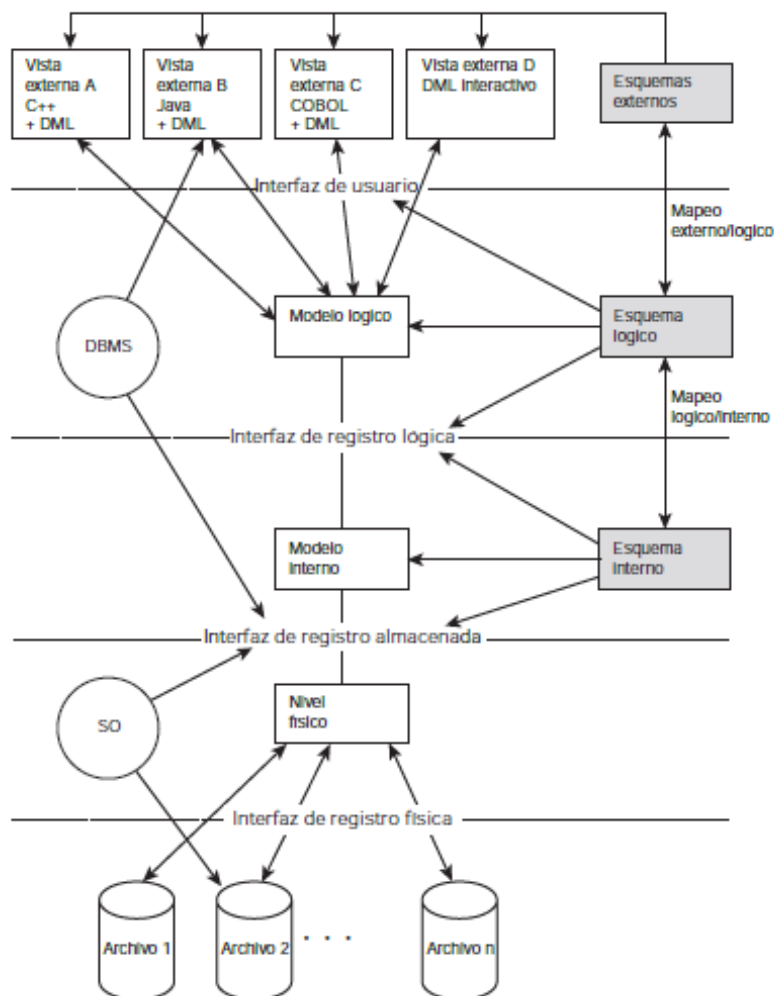
2.2.14.3. Evitar la redundancia y gestionar la concurrencia

La redundancia es fuente de problemas, no sólo por el desaprovechamiento del espacio de almacenamiento, sino porque acarrea problemas de inconsistencia. Como un SGBD provee un fondo de datos común que puede ser compartido por varias aplicaciones, no es preciso por tanto, duplicar datos. Sin embargo, puesto que varias aplicaciones pueden acceder a los mismos datos al mismo tiempo, el SGBD debe disponer de los mecanismos adecuados para “gestionar esas concurrencias”.

El propósito de la arquitectura de tres niveles es separar el modelo del usuario de la estructura física de la base de datos.

La forma en que los usuarios piensan acerca de los datos se llama **nivel externo**. El **nivel interno** es la forma en que los datos se almacenan realmente usando estructuras de datos y organizaciones de archivo estándar. Sin embargo, existen muchas diferentes visiones de usuario y muchas estructuras físicas, de modo que debe haber algún método para mapear las visiones externas a las estructuras físicas. Un mapeo directo es indeseable, pues los cambios hechos a las estructuras físicas o dispositivos de almacenamiento requerirían un cambio correspondiente en el exterior al mapeo físico. Por tanto, hay un nivel medio que proporciona tanto el mapeo como la independencia deseada entre los niveles externo y físico. Éste es el nivel **lógico**. (Ricardo, 2009)

FIGURA N° 2.18: Arquitectura de base de datos en tres niveles – Ricardo (2009)



2.2.15. ENTERPRISE ARCHITECT

Enterprise Architect es una herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas de análisis, modelos de diseño, pruebas y mantenimiento. EA es una herramienta multiusuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad.

El Lenguaje Unificado de Modelado provee beneficios significativos para ayudar a construir modelos de sistemas de software rigurosos y donde es posible mantener la trazabilidad de manera consistente. Enterprise Architect soporta este proceso en un ambiente fácil de usar, rápido y flexible; ésta herramienta provee trazabilidad completa desde el análisis de requerimientos hasta los artefactos de análisis y diseño, a través de la implementación y despliegue. (Lagos, 2010)

2.2.16. .NET Framework

Para describir los principales conceptos relacionados con la plataforma de desarrollo escogida para esta investigación, se comienza con un panorama general del .NET Framework desde su surgimiento, sus principales componentes hasta su implementación para entornos libres mediante el proyecto Mono8 para después centrarse en ASP.NET que es su módulo para el desarrollo Web y finalizar con la descripción de su modelo de conexión a datos desconectado llamado ADO.NET.

2.2.17. MINISTERIO DE SALUD

Según el reglamento de órganos y funciones Ministerio de Salud (s.f.) es un órgano del Poder Ejecutivo. Es el ente rector del Sector Salud que conduce, regula y promueve la intervención del Sistema Nacional Coordinado y Descentralizado de Salud, con la finalidad de lograr el desarrollo de la persona humana, a través de la promoción, protección, recuperación y rehabilitación de su salud y del desarrollo de un entorno saludable, con pleno respeto de los derechos fundamentales de la persona.

2.2.18. INSTITUTO NACIONAL DE SALUD

El Instituto Nacional de Salud es un Organismo Público Ejecutor del Ministerio de Salud dedicado a la investigación de los problemas prioritarios de salud y de desarrollo tecnológico.

El Instituto Nacional de Salud tiene como mandato el proponer políticas y normas, promover, desarrollar y difundir la investigación científica-tecnológica y brindar servicios de salud en los campos de salud pública, control de enfermedades transmisibles y no transmisibles, alimentación y nutrición, producción de biológicos, control de calidad de alimentos, productos farmacéuticos y afines, salud ocupacional, protección del medio ambiente y salud intercultural, para contribuir a mejorar la calidad de vida de la población.

2.2.19. CENTRO NACIONAL DE ALIMENTACIÓN Y NUTRICIÓN

Es el encargado de promover, ejecutar y evaluar las investigaciones y el desarrollo de tecnologías en el ámbito de la alimentación y nutrición humana, Asimismo conduce el sistema de vigilancia nutricional y es responsable de realizar el control de calidad de alimentos; proponiendo acciones que generen hábitos alimenticios saludable y que contribuyan a la reducción de los niveles de desnutrición crónica en el Perú.

2.2.20. DIRECCIÓN REGIONAL DE SALUD

Según el Ministerio de Salud (s.f.), las Direcciones de Salud, son los Órganos Desconcentrados del Ministerio de Salud que ejercen la autoridad de salud por delegación de la Alta Dirección y tienen a su cargo, las siguientes funciones generales en sus respectivas jurisdicciones:

- a. Implementar la visión, misión, política, objetivos y normas sectoriales, en su jurisdicción.
- b. Brindar, en forma eficaz y oportuna, la asistencia, apoyo técnico y administrativo a la gestión de las Direcciones de Red de Salud y de los Hospitales bajo su dependencia y jurisdicción.
- c. Mantener informadas a las entidades públicas y organizaciones en general, que desarrollen actividades afines para el Sector Salud sobre los dispositivos legales para la Salud, evaluando su cumplimiento.

2.2.21. RED DE SALUD

Según el Ministerio de Salud (2001) la Red de Salud es un conjunto de establecimientos y servicios de salud, de diferentes niveles de complejidad y capacidad de resolución, interrelacionados por una red vial y corredores sociales, articulados funcional y

administrativamente, cuya combinación de recursos y complementariedad de servicios asegura la provisión de un conjunto de atenciones prioritarias de salud, en función de las necesidades de la población.

2.2.22. MICRORRED DE SALUD

Según el Ministerio de Salud (2001) la Microrred de Salud es un conjunto de establecimientos de salud del primer nivel de atención cuya articulación funcional, según criterios de accesibilidad, ámbito geográfico y epidemiológico, facilita la organización de la prestación de servicios de salud. Constituye la unidad básica de gestión y organización de la prestación de servicios. La agregación de microrredes conformará una red de salud.

2.2.23. ESTABLECIMIENTOS DE SALUD

Según R.M. N° 769 2004 CADM la unidad operativa de la oferta de servicios de salud, clasificada en una categoría e implantada con recursos humanos, materiales y equipos destinada a realizar procesos asistenciales y administrativos. Tiene como finalidad brindar atenciones sanitarias preventivas, promocionales, recuperativas o de rehabilitación intramurales y extramurales de acuerdo a su capacidad. (DIRESA, 2008)

Tipos de establecimientos de Salud

Entiéndase por Establecimiento de Salud a los hospitales cualquiera sea su tamaño o especialidad, públicos o privados, policlínicos, Centros de Salud Hospital, Puestos Médicos, Puestos Sanitarios, clínicas y Consultorios Privados y todo aquel en el que se realizan actividades concernientes a la salud de las personas. (DIRESA, 2008)

Aquellos que realizan en régimen ambulatorio o de internamiento atención de salud con fines de prevención, promoción, diagnóstico, tratamiento y rehabilitación, dirigidas a mantener el estado de salud de las personas, la familia y la comunidad. (MINSAL, 2001)

a. Finalidad de los Establecimientos de Salud

La finalidad de los Establecimientos de Salud, es la de proveer a las personas servicios integrales que tiendan a prevenir las enfermedades, promover la salud, reparar las enfermedades y rehabilitar a los impedidos, así como servir de centros de investigación y de enseñanza para la formación de los recursos humanos que requiere el país. (DIRESA, 2008)

b. Funciones de los Establecimientos de Salud

Los Establecimientos de salud según tamaño y su naturaleza deben cumplir las siguientes funciones:

- a) Servicios de medicina preventiva
- b) Servicio de promoción o fomento de la Salud
- c) Medicina curativa
- d) Educación para la salud
- e) Servicios de rehabilitación
- f) Servicios de control de medio ambiente
- g) Enseñanza e investigación

2.2.24. ORGANIZACIÓN DE ENTIDADES POR NIVELES

Figura N°2.19: Organización de entidades por niveles: Fuente: Elaboración propia

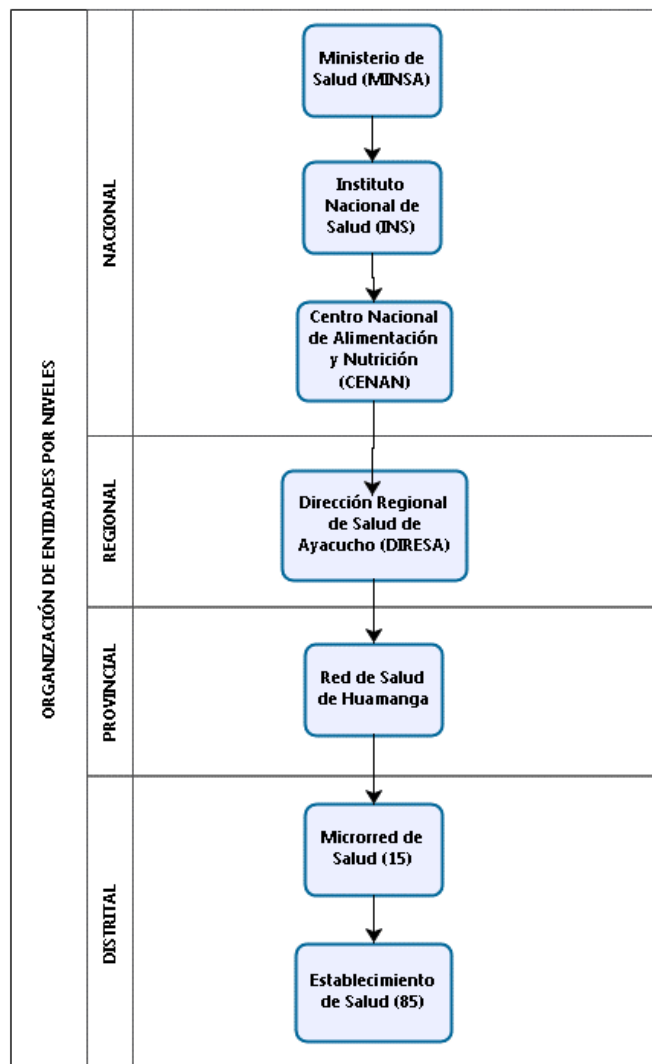


Figura N°2.20: Establecimientos de salud que pertenecen a cada Microrred: Fuente:

Elaboración propia

ESTABLECIMIENTOS DE SALUD QUE PERTENECEN A CADA MICRORRED	BELEN	C.S. BELEN	P.S. BARRIOS ALTOS	P.S. HUASCAHURA	P.S. MORRO DE ARICA	P.S. RANCHA	P.S. SANTA ANA		
	CARMEN ALTO	C.S. CARMEN ALTO	C.S. VISTA ALEGRE	P.S. POKRAS	P.S. YANAMA				
	CHEREA	P.S. ALLPACHACA	P.S. CHIARA	P.S. MANALLASACC	P.S. SACHABAMBA				
	CHONTACA	P.S. ACOCRO	P.S. CCOLLCCA	P.S. PAMPAMARCA	P.S. SECCELAMBRAS				
	LICENCIADOS	C.S. LICENCIADOS	P.S. MOLLEPATA	P.S. VILLA SAN CRISTOBAL					
	MUYURINA	C.S. SIMPAPATA	P.S. LA COMPAÑIA	P.S. MUYURINA	P.S. NIÑO YUCAY	P.S. PACAYCASA	P.S. SANTIAGO DE PESCHA		
	OCROS	C.S. OCROS	P.S. MAYABAMBA						
	PARAS	C.S. PARAS	P.S. CCAHUJACCOCCO	P.S. CRUZ PAMPA	P.S. IGLESIA HIASI				
	PUTACCA	C.S. PUTACCA	P.S. CATALINAYOC	P.S. CUCHUQUESERA	P.S. MILLPO	P.S. PUNKUPATA	P.S. ROSASPATA		
	QUINUA	C.S. QUINUA	P.S. HUAYCHAO						
	SAN JUAN BAUTISTA	C.S. LOS OLIVOS	C.S. MIRAFLORES	C.S. SAN JUAN BAUTISTA	P.S. NAHUJNPUQUIO				
	SANTA ELENA	C.S. SANTA ELENA	P.S. HUAYHUACONDO	P.S. TAMBILLO	P.S. YANAMILLA				
	SOCOS	P.S. LUYANTA	P.S. MANZANAYOCC	P.S. SAN JOSE DE TICCLAS	P.S. SAN RAFAEL	P.S. SANTA ROSA DE COCHABAMBA	P.S. SOCOS		
	TOTOS	P.S. VILCANCHOS							
	VINCHOS	C.S. VINCHOS	P.S. ARIZONA	P.S. CCAHUJACCPAMPA	P.S. CCOÑANI	P.S. ÑAUPALLACCTA	P.S. OCCOLLO	P.S. SAN JUAN CULLUHUANCCA	P.S. PACCHA

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

3.1. TIPO DE INVESTIGACIÓN

La investigación es de tipo aplicada, ya que es una actividad que tiene por finalidad la búsqueda y consolidación del saber, y la aplicación de los conocimientos para resolver problemas de la vida cotidiana y problemas prácticos que beneficia a la sociedad.

3.2. NIVEL DE LA INVESTIGACIÓN

Explicativa, ya que está dirigido a responder por las causas de los eventos y fenómenos físicos o sociales. Se enfoca en explicar por qué ocurre un fenómeno y en qué condiciones se manifiesta, o por qué se relacionan dos o más variables.

3.3. MÉTODO DE LA INVESTIGACIÓN

No experimental, porque solo observamos los fenómenos tal como se dan en su contexto natural, para posteriormente analizarlos.

3.4. DISEÑO DE LA INVESTIGACIÓN

No experimental, longitudinal.

El diseño de la investigación es no experimental debido que no se manipularon de manera deliberada las variables y solo se observaron los fenómenos en su ambiente natural.

Además el diseño de investigación fue de tipo longitudinal ya que se recabaron datos en diferentes puntos del tiempo, para realizar inferencias acerca de la evolución del problema de investigación o fenómeno, sus causas y sus efectos.

3.5. POBLACIÓN Y MUESTRA

POBLACIÓN

Niños menores de 5 años y gestantes de la región de Ayacucho.

MUESTRA

Niños menores de 5 años y gestantes que pertenecen a la Red de Salud de Huamanga (Provincia de Huamanga)

3.6. VARIABLES E INDICADORES

VARIABLE 1

V1: Información Georreferenciada.- Conjunto de datos pertenecientes a un elemento situado en un determinado lugar de la Tierra.

INDICADORES DE LA VARIABLE 1

I1: Gestión de información.- Proceso que incluye operaciones como recolección, manipulación, tratamiento, conservación y acceso de la información

I2: Mapa cuantitativo.- Mapa con información representada numéricamente.

VARIABLE 2

V2: Estado nutricional de niños y gestantes.- Resultante final del balance entre ingesta y requerimiento de nutrientes.

INDICADORES DE LA VARIABLE 2

I1: Desnutrición.- Afección que ocurre cuando el cuerpo no recibe los nutrientes suficientes para su bienestar.

I2: Anemia.- Empobrecimiento de la sangre por disminución de su cantidad total.

I3: Obesidad.- Exceso de grasa, general o localizada, que se manifiesta en un valor de peso elevado comparado con el de personas de la misma edad y sexo

3.7. TÉCNICAS E INSTRUMENTOS PARA LA RECOLECCIÓN DE DATOS

TÉCNICAS

a. **Observación.-** Se observará cómo es la gestión de información del estado nutricional de niños y gestantes.

- b. **Análisis documental.-** Se analizará la documentación que se hizo en el establecimiento de salud del estado nutricional de niños y gestantes.

Tabla N° 3.2: Registro de gestantes



**SISTEMA DE INFORMACION DEL ESTADO NUTRICIONAL
FORMATO DE REGISTRO DIARIO DE LA GESTANTE**

DIRESA:	
RED:	
MICRORED:	
ESTABLECIMIENTO:	

V°B° de Jefatura de EESS:

Nota: Sólo se registrará una visita de la gestante en el mes.

FECHA DE ATENCION (dd/mm/aa)	SEMANA EPIDEMIOLOGICA (S.E.)	N°	DNI	N°HISTORIA CLINICA (H. C.)	EDAD (años)	EDAD GESTACIONAL (semanas)	PESO ACTUAL (Kg.) (----)	TALLA (cm.) (----)	PESO PRE-GESTAC (Kg.) (----)	TIPO DE EMBARAZO		LABORATORIO (Hb OBSERVADA)*		LUGAR DE RESIDENCIA				
										Unico (U)	Múltiple (M)	HEMOGLOBINA (gr/dl) (---)	FECHA DE RESULTADO HEMOGLOBINA (dd/mm/aa)	PROVINCIA	DISTRITO	LOCALIDAD**	ALTITUD DE LA LOCALIDAD (msnm)	
		1										
		2										
		3										
		4										
		5										

b. Análisis de contenido

Figura N° 3.1. Ficha de control de crecimiento de niño - Análisis de contenido de datos.

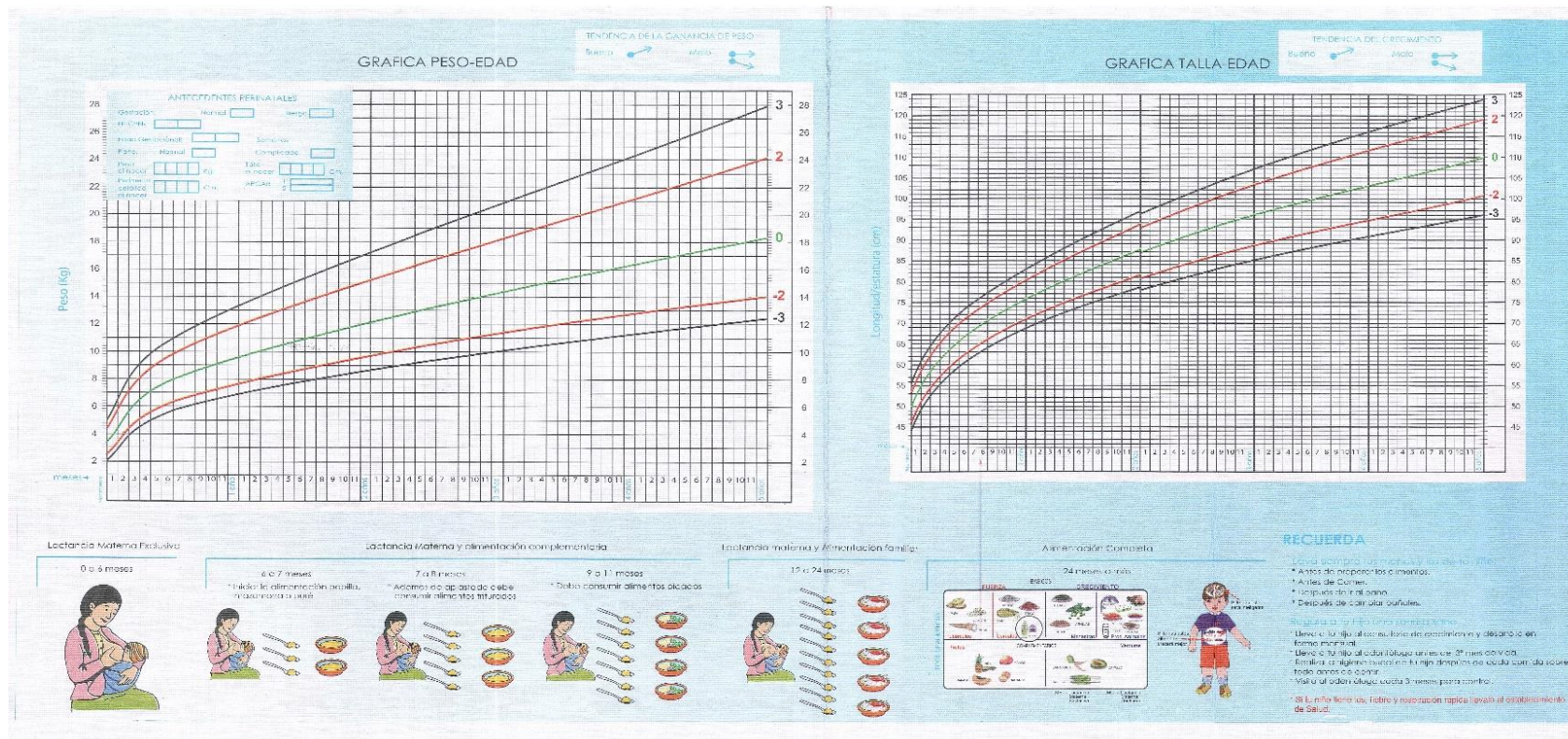


Figura N° 3.2. Ficha de control de crecimiento de niña - Análisis de contenido de datos.

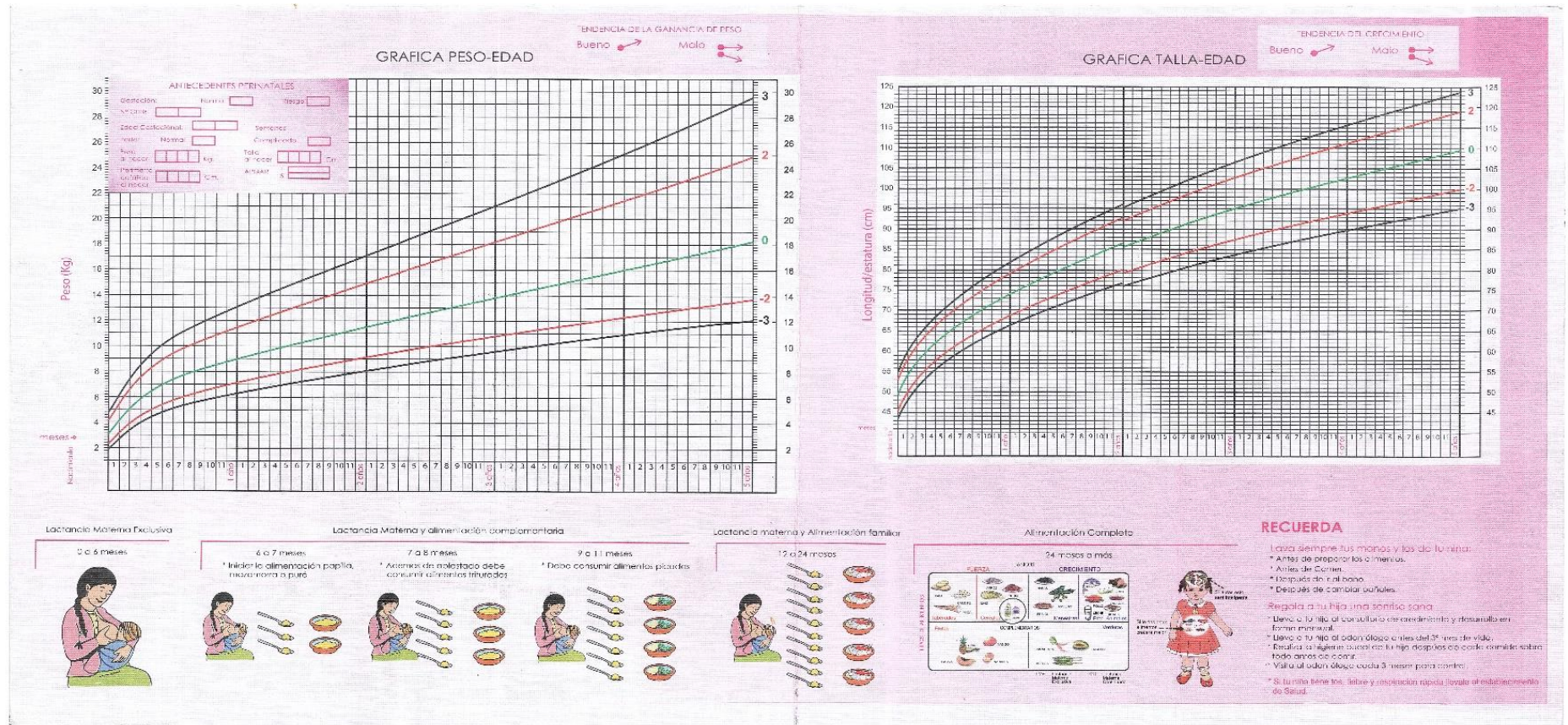


Figura N° 3.3. Tabla de recomendaciones para gestantes- Análisis de contenido de datos.

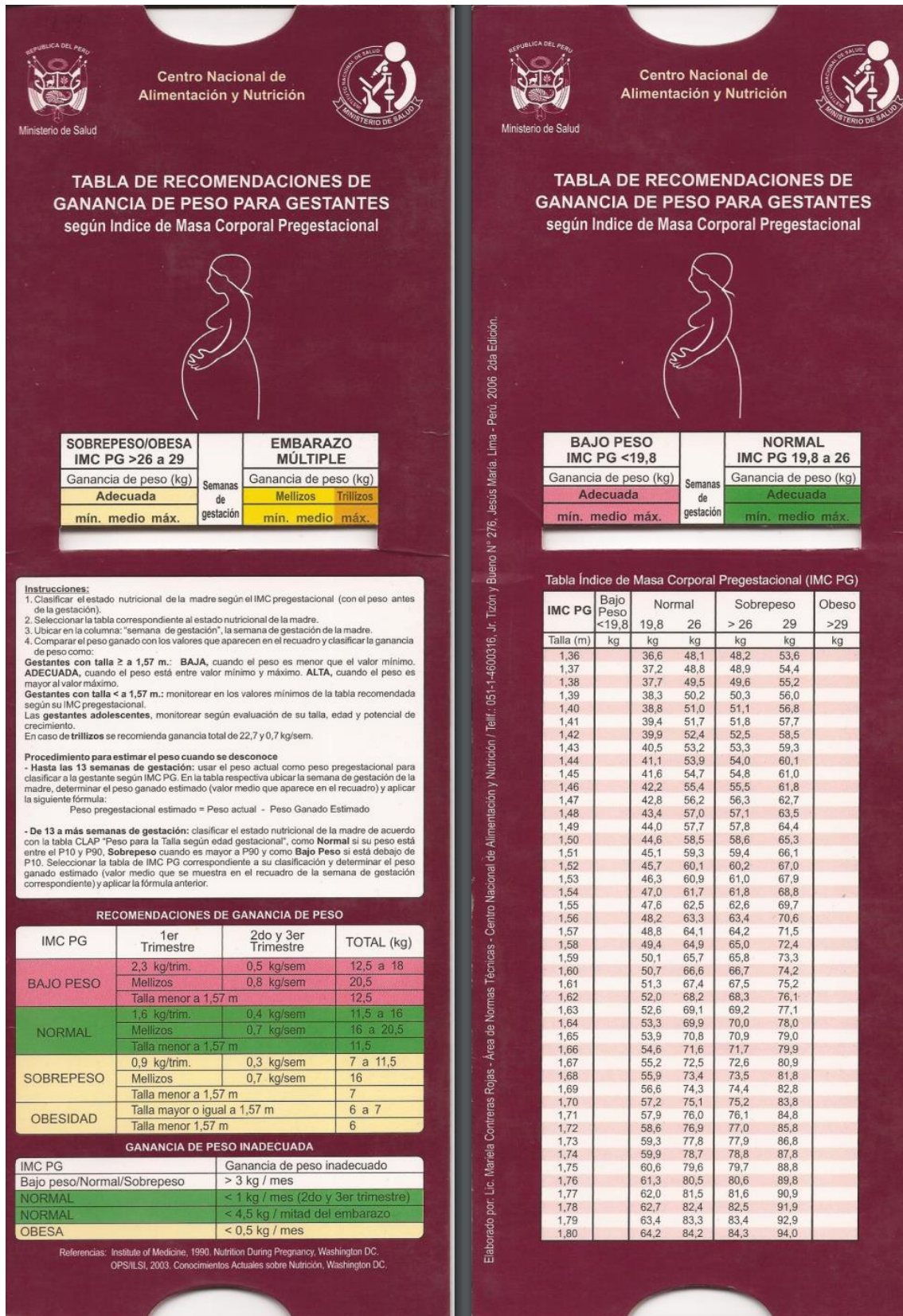


Figura N° 3.3. Tabla de recomendaciones para gestantes- Análisis de contenido de datos.

SOBREPESO IMC PG >26 a 29				Semanas de gestación	EMBARAZO MÚLTIPLE		
Ganancia de peso (kg)			Ganancia de peso (kg)				
Adecuada			Mellizos Trillizos				
mín. medio máx.			mín. medio máx.				
0,1			1	0,1			
0,1			2	0,2			
0,2			3	0,4			
0,3			4	0,5			
0,3			5	0,6			
0,4			6	0,7			
0,5			7	0,9			
0,6			8	1,0			
0,6			9	1,1			
0,7			10	1,2			
0,8			11	1,4			
0,8			12	1,5			
0,9			13	1,6			
1,1	1,2	1,3	14	2,1	2,3	2,4	
1,4	1,5	1,7	15	2,7	3,0	3,2	
1,6	1,8	2,1	16	3,2	3,7	3,9	
1,8	2,1	2,5	17	3,7	4,4	4,7	
2,0	2,5	2,9	18	4,3	5,1	5,5	
2,3	2,8	3,3	19	4,8	5,8	6,3	
2,5	3,1	3,6	20	5,3	6,5	7,1	
2,7	3,4	4,0	21	5,9	7,2	7,9	
2,9	3,7	4,4	22	6,4	7,9	8,6	
3,2	4,0	4,8	23	6,9	8,6	9,4	
3,4	4,3	5,2	24	7,5	9,3	10,2	
3,6	4,6	5,6	25	8,0	10,0	11,0	
3,8	4,9	6,0	26	8,5	10,7	11,8	
4,1	5,2	6,4	27	9,1	11,4	12,5	
4,3	5,5	6,8	28	9,6	12,1	13,3	
4,5	5,9	7,2	29	10,1	12,8	14,1	
4,7	6,2	7,6	30	10,7	13,5	14,9	
5,0	6,5	8,0	31	11,2	14,2	15,7	
5,2	6,8	8,4	32	11,7	14,9	16,4	
5,4	7,1	8,8	33	12,3	15,6	17,2	
5,6	7,4	9,1	34	12,8	16,3	18,0	
5,9	7,7	9,5	35	13,3	17,0	18,8	
6,1	8,0	9,9	36	13,9	17,7	19,6	
6,3	8,3	10,3	37	14,4	18,4	20,4	
6,5	8,6	10,7	38	14,9	19,1	21,1	
6,8	8,9	11,1	39	15,5	19,8	21,9	
7,0	9,3	11,5	40	16,0	20,5	22,7	

BAJO PESO IMC PG <19,8				Semanas de gestación	NORMAL IMC PG 19,8 a 26		
Ganancia de peso (kg)			Ganancia de peso (kg)				
Adecuada			Adecuada				
mín. medio máx.			mín. medio máx.				
0,2			1	0,1			
0,4			2	0,2			
0,5			3	0,4			
0,7			4	0,5			
0,9			5	0,6			
1,1			6	0,7			
1,2			7	0,9			
1,4			8	1,0			
1,6			9	1,1			
1,8			10	1,2			
1,9			11	1,4			
2,1			12	1,5			
2,3			13	1,6			
2,7	2,8	2,9	14	2,0	2,1	2,1	
3,1	3,3	3,5	15	2,3	2,5	2,7	
3,4	3,8	4,0	16	2,7	3,0	3,2	
3,8	4,2	4,6	17	3,1	3,4	3,7	
4,2	4,7	5,2	18	3,4	3,9	4,3	
4,6	5,2	5,8	19	3,8	4,3	4,8	
4,9	5,7	6,4	20	4,2	4,8	5,3	
5,3	6,2	7,0	21	4,5	5,2	5,9	
5,7	6,6	7,5	22	4,9	5,7	6,4	
6,1	7,1	8,1	23	5,3	6,1	6,9	
6,5	7,6	8,7	24	5,6	6,6	7,5	
6,8	8,1	9,3	25	6,0	7,0	8,0	
7,2	8,5	9,9	26	6,4	7,5	8,5	
7,6	9,0	10,4	27	6,7	7,9	9,1	
8,0	9,5	11,0	28	7,1	8,4	9,6	
8,3	10,0	11,6	29	7,5	8,8	10,1	
8,7	10,5	12,2	30	7,8	9,3	10,7	
9,1	10,9	12,8	31	8,2	9,7	11,2	
9,5	11,4	13,3	32	8,6	10,2	11,7	
9,9	11,9	13,9	33	8,9	10,6	12,3	
10,2	12,4	14,5	34	9,3	11,1	12,8	
10,6	12,9	15,1	35	9,7	11,5	13,3	
11,0	13,3	15,7	36	10,0	12,0	13,9	
11,4	13,8	16,3	37	10,4	12,4	14,4	
11,7	14,3	16,8	38	10,8	12,9	14,9	
12,1	14,8	17,4	39	11,1	13,3	15,5	
12,5	15,3	18,0	40	11,5	13,7	16,0	

Elaborado por: Lic. Mariela Contreras Rojas - Área de Normas Técnicas - Centro Nacional de Alimentación y Nutrición | Telf.: 051-1-4600316, Jr. Trzón y Buero N° 276, Jesús María, Lima - Perú, 2006. 2da Edición.

IMC PG = Índice de Masa Corporal Pregestacional
 Referencias: Institute of Medicine, 1990. Nutrition During Pregnancy, Washington DC.
 OPS/ILSI, 2003. Conocimientos Actuales sobre Nutrición, Washington DC.

IMC PG = Índice de Masa Corporal Pregestacional
 Referencias: Institute of Medicine, 1990. Nutrition During Pregnancy, Washington DC.
 OPS/ILSI, 2003. Conocimientos Actuales sobre Nutrición, Washington DC.

3.8. TÉCNICAS DE PROCESAMIENTO Y ANÁLISIS DE DATOS

Para el tratamiento de datos y la información se empleó las siguientes herramientas:

Tabla N° 3.3: Herramientas para el tratamiento de información

Software	Fabricante	Licencia	Servicio
Microsoft Excel 2013	Microsoft	Licencia propietaria	Hoja de cálculo
IBM SPSS Statistics	IBM	Licencia propietaria	Es una familia de software estadístico integrada que se centra en el completo proceso analítico, desde la planificación a la colección de datos y al análisis, "reporting" y despliegue.

3.9. HERRAMIENTAS PARA EL TRATAMIENTO DE DATOS E INFORMACIÓN

Las herramientas tecnológicas que se han seleccionado para el desarrollo de la investigación, están relacionados con el tratamiento y procesamiento de datos.

Tabla N° 3.4: Herramientas tecnológicas para la plataforma.

Software	Fabricante	Licencia	Servicio
Windows 7 Ultimate Edition	Microsoft Corporation	Licencia propietaria	Es la edición más versátil y eficiente de Windows. Destaca por un menor consumo de recursos, combinado con mejoras visuales que harán más cómodo el manejo diario de este sistema operativo. La seguridad es uno de los pilares más importantes de Windows 7, que cuenta con protección

			antispyware, herramientas para copias de seguridad, transferencia de archivos y configuraciones.
Windows Server 2012 R2	Microsoft Corporation	Licencia propietaria	Le permite optimizar el rendimiento de sus tareas de servidor más importantes y le ayuda a protegerse frente a caídas del servicio gracias a unas potentes opciones de recuperación.
Internet Information Server	Microsoft Corporation	Licencia propietaria	Servidor de aplicaciones, que ejecuta sistemas web, servicios web.

Tabla N° 3.5: Herramientas tecnológica – Leguaje de programación.

Software	Fabricante	Licencia	Servicio
Microsoft Visual Studio 2015	Microsoft Corporation	Licencia propietaria	Es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET.
.Net Framework 3.5	Microsoft Corporation	CDDL (Licencia Común de	ASP.NET es un Framework de aplicaciones web desarrollado y comercializado por Microsoft, para permitir a

		Desarrollo y Distribución)	los programadores crear sitios Web, aplicaciones Web y servicios Web dinámicos. ASP.NET se basa en el CommonLanguageRuntime (CLR), permitiendo a los programadores escribir código ASP.NET usando cualquier apoyo .NET.
Unit Test	Microsoft Corporation	Licencia propietaria	Componente para verificar el funcionamiento de una clase y sus métodos.
CSS 3.0	Open Source	Libre	CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML. El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.
JQuery	Open	Libre	Framework de Javascript, para incluir efectos visuales.
ReportViewer	Microsoft Corporation	Licencia propietaria	Aplicación de inteligencia empresarial utilizada para diseñar y generar informes desde una amplia gama de fuentes de datos (bases de

			datos). Varias aplicaciones, como Microsoft Visual Studio,
Api de Google Maps	Google	Libre	Una API es un conjunto de métodos y herramientas que se pueden utilizar para crear aplicaciones de software. La API de Google Maps nos permite mostrar mapas en un sitio web.

Tabla N° 3.6: Herramientas tecnológicas – Base de datos.

Software	Fabricante	Licencia	Servicio
Microsoft Sql Server 2008	Microsoft Corporation	Licencia propietaria	Sistema gestor de bases de datos relacional.
AllFusion ERwin Data Modeler	Computer Associates International Inc. (CA)	Licencia propietario	Es una herramienta de diseño de base de datos que ayuda a los usuarios a diseñar, generar y mantener alta calidad de las aplicaciones de base de datos de alta performance. AllFusion ERwin Data Modeler permite al usuario visualizar la estructura correcta, elementos claves y el diseño optimizado de su base de datos, desde los requerimientos de un modelo lógico de información y reglas de negocio que definen la base de datos, a un modelo físico optimizado para las características

			específicas de la base de datos seleccionada.
--	--	--	---

Tabla N° 3.7: Herramientas tecnológicas – Modelamiento de datos.

Software	Fabricante	Licencia	Servicio
Enterprise Architect Vs. 7.1	SPARX SYSTEM COMMUNITY	Licencia propietario	Herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento.

3.10. METODOLOGÍA ICONIX PARA EL DESARROLLO DEL SISTEMA DE INFORMACIÓN GEORREFERENCIADA DEL ESTADO NUTRICIONAL DE NIÑOS Y GESTANTES

Las siguientes tablas se obtuvieron de acuerdo a un resumen realizado del libro Agile Development with ICONIX cuyos autores son: Rosenberg, Stephenns y Collins-Cope.

a. ANÁLISIS DE REQUISITOS

Tabla N° 3.8: Análisis de Requisitos. Fuente: Elaboración propia.

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Identificar requisitos	* Requisitos funcionales y no funcionales * Casos de prueba	* Entrevistas * Definir lo que el sistema debe hacer * Escribir al menos un caso de prueba para cada requisito	Usuario Analista
Identificar objetos del mundo real y dibujar el modelo de dominio	Modelo de dominio	* Identificar clases clave del negocio * Identificar sustantivos y depurar * Identificar objetos en requisitos funcionales y asignar al modelo de dominio * Utilizar agregación y generalización	Analista
Realizar prototipo de interfaz gráfica	Prototipo GUI	* Utilizar historia de eventos del usuario (storyboards) * Utilizar los requisitos funcionales	Programador Analista

		* Diseñar interfaz gráfica básica	
Descubrir casos de uso	Lista de casos de uso	* Utilizar requisitos funcionales * Entrevistas	Usuario Analista
Dibujar y empaquetar casos de uso	* Diagramas de casos de uso * Paquete de casos de uso	* Identificar roles y responsabilidades de actores * Asociar actores con casos de uso * Relacionar casos de uso * Agrupar lógicamente casos de uso	Analista
Asignar requisitos funcionales a los casos de uso	Relación entre requisitos funcionales y casos de uso	* Asignar requisitos funcionales a los casos de uso	
Escribir el primer borrador de casos de uso	Primer borrador de casos de uso	* Utilizar glosario de objetos del modelo de dominio * Utilizar la regla de dos párrafos * Escribir el caso de uso como flujos de evento/respuesta * Escribir el caso de uso con estructura sustantivo- verbo-sustantivo * Escribir caso de uso en voz activa * Referenciar por su nombre las pantallas	

b. REVISIÓN DE REQUISITOS

Tabla N° 3.9: Revisión de Requisitos (primer hito). Fuente: Elaboración propia.

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Revisar el modelo de dominio	* Modelo de dominio	* Identificar al menos 80% de clases clave de dominio del problema	Usuario Analista Programador
Revisar el prototipo GUI	Prototipo GUI	* Diseñar con precisión la GUI relacionada al caso de uso	
Revisar modelo de casos de uso	Casos de uso revisado	<ul style="list-style-type: none"> * Eliminar clases fuera del dominio del problema * Cambiar descripción de voz pasiva activa * Describir todos los cursos alternos * Asociar todos los requisitos a los casos de uso * Describir que intenta hacer el usuario para cada caso de uso 	

c. DISEÑO PRELIMINAR

Tabla N° 3.10: Diseño preliminar. Fuente: Elaboración propia.

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Reescribir el primer borrador para cada caso de uso	Caso de uso desambiguado	* Reescribir el caso de uso durante el análisis de robustez	Analista
Identificar el primer corte de objetos que completan escenarios para cada caso de uso	Diagrama de robustez	<ul style="list-style-type: none"> * Copiar la descripción del caso de uso en el diagrama de robustez * Usar las clases del modelo de dominio * Crear un objeto interfaz por cada GUI y nombrarlo * Transformar verbos del caso de uso en controladores * Relacionar un caso de uso al diagrama de robustez cuando es invocado * Utilizar las reglas para construir el diagrama de robustez 	

Actualizar el modelo de dominio	Modelo de dominio actualizado	* Actualizar el modelo de dominio con nuevas clases y atributos durante el análisis de robustez	
Actualiza el diagrama de clases de análisis	Modelo de dominio actualizado	* Actualizar el diagrama de clases de análisis al finalizar el análisis de robustez * Asignar atributos a las clases entidad	

d. REVISIÓN DEL DISEÑO PRELIMINAR

Tabla N° 3.11: Revisión del diseño preliminar (segundo hito). Fuente: Elaboración propia.

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Revisar descripción de casos de uso	Caso de uso	* Coincidir la descripción del caso de uso con el diagrama de robustez	
Revisar diagrama de robustez	Diagrama de robustez	* Coincidir el diagrama de robustez con descripción del caso de uso	Usuario Analista Programador

		* Verificar que el diagrama de robustez tiene todos los cursos alternos	
Revisar el modelo de dominio actualizado	Modelo de dominio actualizado	* Coincidir los objetos entidad del diagrama de robustez con el modelo de dominio actualizado	

e. ARQUITECTURA TÉCNICA

Tabla N° 3.12: Tareas, artefactos, técnicas y responsables para la arquitectura técnica. Fuente: Elaboración propia.

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Diseñar diagrama de componentes	Diagrama de componentes	<ul style="list-style-type: none"> * Entrevistas * Características del negocio * Utilizar la arquitectura MVC * Integrar frameworks 	Programador
Diseñar diagrama de despliegue	Diagrama de despliegue	<ul style="list-style-type: none"> * Entrevistas * Características del negocio * Utilizar el modelo cliente servidor 	Diseñador de BD

f. DISEÑO

Tabla N° 3.13: Diseño. Fuente: Elaboración propia.

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Dividir el modelo de dominio actualizado para cada caso de uso	Parte del modelo de dominio actualizado	* Coincidir las clases entidad del diagrama robustez con parte del modelo de dominio actualizado y dibujarlo	Analista
Dibujar un diagrama de secuencia para cada caso de uso	Diagrama de secuencia	<ul style="list-style-type: none"> * Copiar la descripción del caso de uso * Copiar objetos entidad, interfaz y actores del diagrama de robustez * Verificar que un mensaje del diagrama de secuencia es verbo en el caso de uso * Hacer refactoring al diagrama de secuencia antes de codificar 	
Actualizar diagrama de clases de un caso de uso	Diagrama de clases	* Asignar operaciones a las clases a partir de mensajes del diagrama de secuencia	Programador

		<ul style="list-style-type: none"> * Establecer multiplicidad de clases * Depurar las clases, operaciones y atributos del diagrama de clases 	Diseñador de BD
Extraer controladores para pruebas unitarias	Lista de controladores	* Identificar controladores para la lógica del negocio desde un diagrama de robustez	

g. REVISIÓN CRÍTICA DE DISEÑO

Tabla N° 3.14: Revisión crítica de diseño (tercer hito). Fuente: Elaboración propia.

TAREA	ARTEFACTO	TÉCNICA	RESPONSABLES
Revisar diagrama de secuencia	Diagrama de secuencia	<ul style="list-style-type: none"> * Verificar que el diagrama de secuencia coincide con la descripción del caso de uso * Verificar que el diagrama de secuencia representa los cursos básicos y alternos 	Diseñador de BD

		* Verificar en los mensajes que los atributos y los valores de retorno son correctos	Programador
Revisar diagrama de clases	Diagrama de clases	* Verificar que el nombre, atributos y operaciones se asignaron correctamente a las clases * Asignar requisitos no funcionales a los casos de uso y clases	
Revisar modelo de dominio actualizado	Modelo de dominio actualizado	* Verificar nombres y atributos del modelo de dominio actualizado	
Revisar lista de pruebas unitarias	Lista de controladores	* Actualizar la lista de controladores	

CAPÍTULO IV

ANÁLISIS Y RESULTADOS DE LA INVESTIGACIÓN

4.1. RESULTADOS Y ARTEFACTOS DEL SOFTWARE APLICANDO EL PROCESO DE ICONIX

Aplicando las técnicas de la metodología ICONIX, obtenemos los artefactos como tablas y figuras para el modelado e implementación del Sistema Web de Información Georreferenciada del Estado Nutricional de Niños y Gestantes de la Dirección Regional de Salud de Ayacucho, 2016.

4.1.1. ETAPA 1: ANÁLISIS DE REQUISITOS

Siendo la Unidad de Estadística, Informática y Telecomunicaciones de la Red de Salud de Huamanga la encargada de procesar la información y obtener estadísticas vitales fiables que reflejan la dinámica demográfica provincial, se procede a recopilar la información necesaria, obteniendo los siguientes requisitos:

4.1.1.1. REQUISITOS FUNCIONALES

Tabla N° 4.1: Requisitos Funcionales

N°	REQUISITOS FUNCIONALES
ReqF01.	El sistema debe permitir registrar una cuenta de usuario por trabajador .
ReqF02.	El sistema debe permitir actualizar una cuenta de usuario.
ReqF03.	El sistema debe permitir eliminar una cuenta de usuario.
ReqF04.	El sistema debe permitir realizar búsqueda de cuenta de usuario.
ReqF05.	El sistema debe permitir obtener un reporte de las cuentas de usuario de los digitadores según establecimiento, microrred, red, DIRESA a la que pertenece.
ReqF06.	El sistema debe permitir ingresar al menú de atención del paciente , ya sea niño o gestante mediante un nombre de usuario y una contraseña.

ReqF07.	El sistema deber permitir registrar datos de un niño.
ReqF08.	El sistema debe permitir modificar los datos de un niño.
ReqF09.	El sistema debe permitir eliminar un niño.
ReqF10.	El sistema debe permitir realizar la búsqueda de un niño a través de filtros.
ReqF11.	El sistema debe permitir la emisión de un reporte del estado nutricional de niños por distrito, provincia y departamento .
ReqF12.	El sistema deber permitir registrar datos de una gestante.
ReqF13.	El sistema deber permitir modificar los datos de una gestante.
ReqF14.	El sistema debe permitir eliminar una gestante.
ReqF15.	El sistema debe permitir realizar la búsqueda de una gestante.
ReqF16.	El sistema debe permitir la emisión de un reporte del estado nutricional de gestantes por distrito, provincia y departamento.
ReqF17.	El sistema debe permitir generar reportes con gráficos estadísticos del estado nutricional de niños.
ReqF18.	El sistema debe permitir generar reportes con gráficos estadísticos del estado nutricional de gestantes.

4.1.1.2. REQUISITOS NO FUNCIONALES

Tabla N° 4.2: Requisitos No Funcionales

N°	REQUISITOS NO FUNCIONALES
ReqNF01.	El sistema debe estar basado en entorno web y presentar una interfaz amigable.
ReqNF02.	El sistema debe presentar una arquitectura técnica y codificación usando estándares que permita su operación y mantenimiento adecuado.
ReqNF03.	El sistema debe poseer la propiedad de escalabilidad.
ReqNF04.	El sistema debe contener niveles de seguridad adecuados y acceso a la información por roles y grupos de usuarios.
ReqNF05.	Para el desarrollo de la aplicación se usarán estándares de nomenclatura para las clases, objetos, vistas, controladores, etc.; que

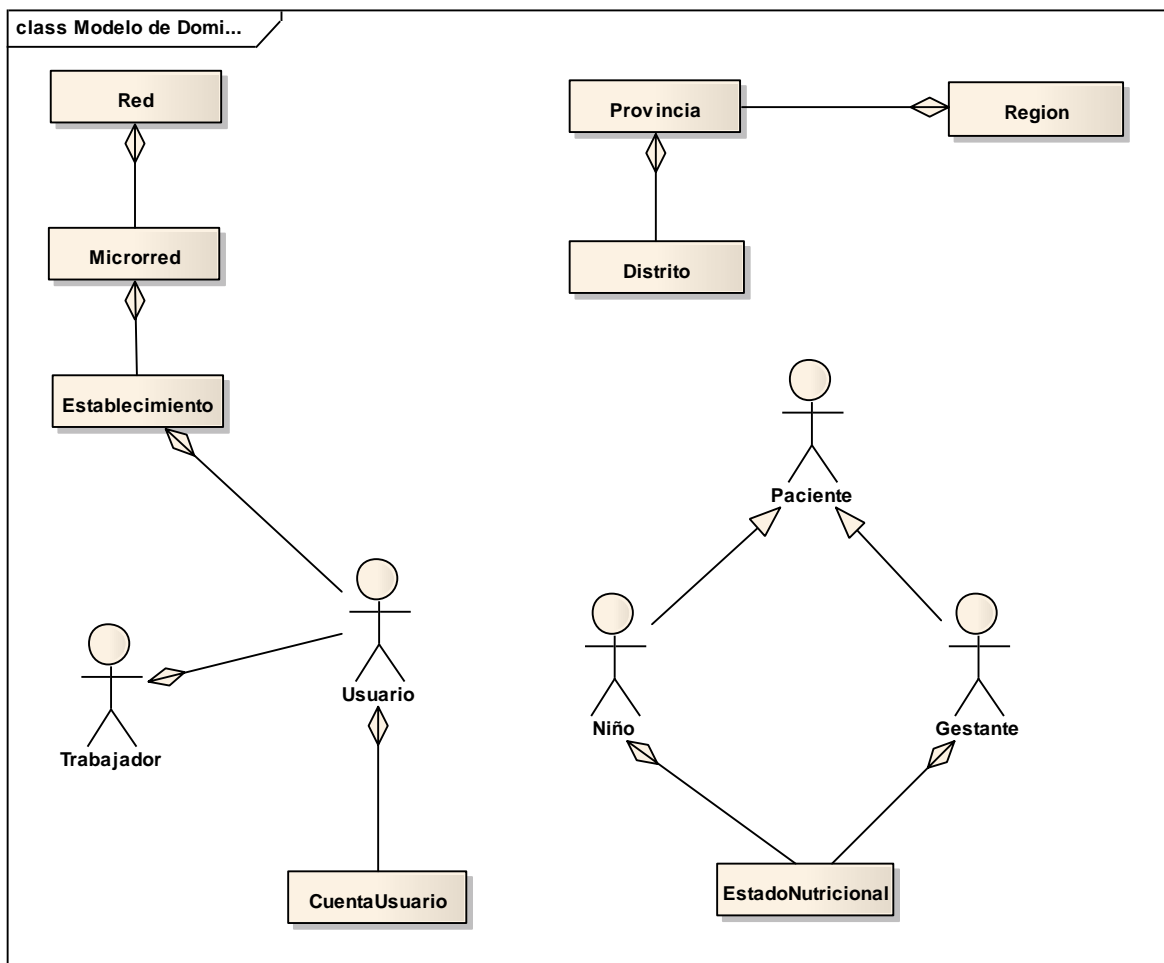
	permitan una adecuada mantenibilidad de la aplicación en las etapas de pruebas unitarias y post-producción.
ReqNF06.	El sistema mejorará los tiempos de respuesta y la seguridad en el procesamiento de la información, tanto para el procesamiento de datos así como para fines de auditoría de los mismos.

De acuerdo a los requisitos se obtiene una lista de objetos candidatos:

- Cuenta de usuario
- Paciente
- Trabajador
- Establecimiento
- Microrred
- Red
- Niño
- Gestante
- Usuario
- Estado nutricional
- Distrito
- Provincia
- Departamento

4.1.1.3. MODELO DE DOMINIO INICIAL

Figura N° 4.1: Primer intento de Modelo de Dominio



4.1.1.4. PROTOTIPO DE INTERFAZ DE USUARIO DE LOS CASOS DE USO

4.1.1.4.1. INTERFAZ DE USUARIO: Registrar niño

Figura N° 4.2: Interfaz de usuario: Registrar niño

Niños

Datos del niño

Registro de menor de 5 años (Referencia)

Fecha de atención *

Historia clínica

Semana epidemiológica

Sexo Masculino Femenino

Documento DNI CUI

Fecha de nacimiento

Número de documento

Edad (meses)

Datos antropométricos y de laboratorio

Peso (Kg)

Talla (cm)

Hemoglobina (gr/dl)

Fecha de resultado de hemoglobina

CREC - desde la consulta anterior, número recibidos de...

N° de controles

Sachets MM

Consejerías nutricionales

Sesiones demostrativas

Visitas domiciliarias

Afiliación

CRECER

SIS

PIN

JUNTOS

Lugar de residencia

Departamento

Distrito

Altitud (msnm)

Provincia

Localidad

4.1.1.4.2. INTERFAZ DE USUARIO: Mantener niño

Figura N° 4.3: Interfaz de usuario: Mantener niño

Niños

Criterios de Búsqueda

Nro documento:

+ Nuevo niño ☰

Listado de Resultados

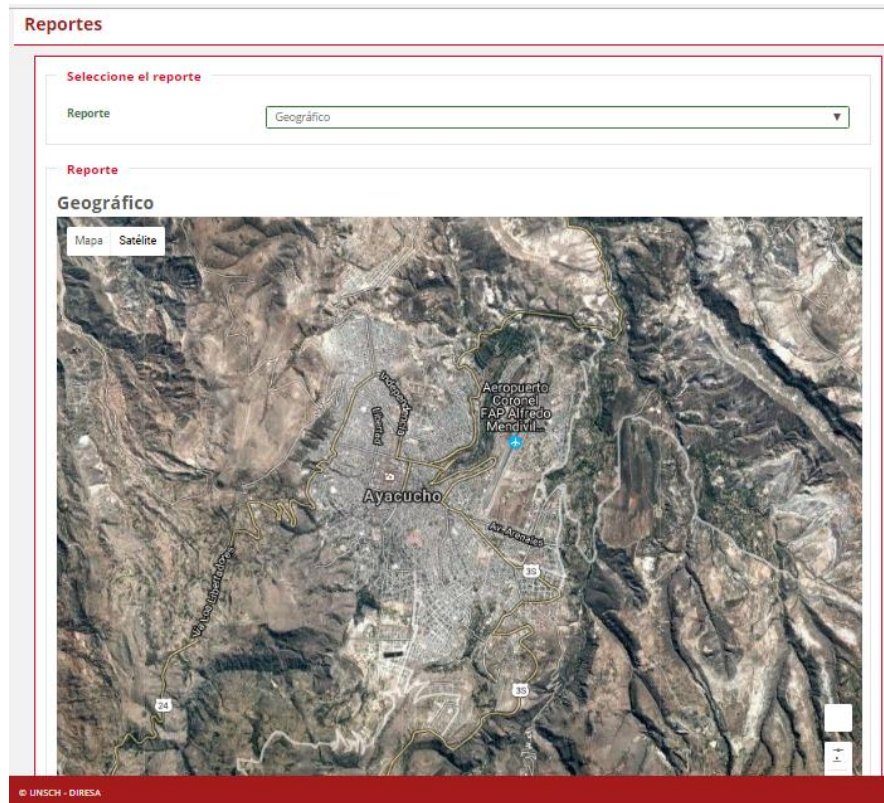
Nro	Fecha de atención	Provincia	Distrito	Documento	Sexo	Edad (meses)	Peso (Kg)	Talla (cm)	Hemoglobina (gr/dl)	Acción
1	31/12/2016	HUAMANGA	SAN JOSE DE TICLLAS	79985477	M	0	3.88	48	0	
2	31/12/2016	HUAMANGA	AYACUCHO	78753414	F	28	11.2	83.7	0	
3	31/12/2016	HUAMANGA	AYACUCHO	27770434	F	4	6.3	52	0	
4	31/12/2016	HUAMANGA	SOCOS	63740361	F	6	17.1	104.2	0	
5	31/12/2016	HUAMANGA	AYACUCHO	79328819	F	15	9.75	74	14.2	
6	31/12/2016	HUAMANGA	SAN JUAN BAUTISTA	81623362	M	22	9.7	79	13	
7	31/12/2016	HUAMANGA	SAN JUAN BAUTISTA	81637714	M	18	10.8	76	12.9	
8	31/12/2016	HUAMANGA	SAN JUAN BAUTISTA	79500126	F	11	7.6	70.2	13.9	
9	31/12/2016	HUAMANGA	SAN JUAN BAUTISTA	79557412	M	9	7.9	70.5	12.9	
10	31/12/2016	HUAMANGA	SAN JUAN BAUTISTA	79610256	F	9	8.6	69.7	13.1	

Mostrar: ▼ Total: 98705 filas

<<
<
1
2
3
4
5
>
>>

4.1.1.4.3. INTERFAZ DE USUARIO: Emitir reporte de niño

Figura N° 4.4: Interfaz de usuario: Emitir reporte de niño



4.1.1.5. IDENTIFICACIÓN DE CASOS DE USO

Tabla N° 4.3: Casos de uso identificados según requisitos

CASO DE USO	REQUISITOS FUNCIONALES
1. Crear cuenta de usuario.	1. El sistema debe permitir la creación de cuenta de usuario por establecimiento.
2. Actualizar cuenta de usuario.	2. El sistema debe permitir actualizar una cuenta de usuario.
3. Eliminar cuenta de usuario.	3. El sistema debe permitir eliminar una cuenta de usuario.
4. Buscar cuenta de usuario.	4. El sistema debe permitir realizar búsqueda de cuenta de usuario.

5. Reporte de cuentas de usuario.	5. El sistema debe permitir obtener un reporte de las cuentas de usuario de los digitadores según establecimiento, microrred, red a la que pertenece.
6. Iniciar session de usuario.	6. El sistema debe permitir ingresar al menú de niños y gestantes mediante un nombre de usuario y una contraseña.
7. Registrar niño.	7. El sistema deber permitir registrar datos de un niño.
8. Modificar niño.	8. El sistema debe permitir modificar los datos de un niño.
9. Eliminar niño.	9. El sistema debe permitir eliminar los datos de un niño.
10. Buscar niño.	10. El sistema debe permitir realizar la búsqueda de un niño.
11. Mostrar reporte del estado nutricional del niño.	11. El sistema debe permitir la emisión de un reporte del estado nutricional de niños por distrito, provincia y departamento.
12. Registrar gestante.	12. El sistema deber permitir registrar datos de una gestante.
13. Modificar gestante.	13. El sistema debe permitir modificar los datos de una gestante.
14. Eliminar gestante.	14. El sistema debe permitir eliminar los datos de una gestante.
15. Buscar gestante.	15. El sistema debe permitir realizar la búsqueda de una gestante.
16. Mostrar reporte del estado nutricional de la gestante.	16. El sistema debe permitir la emisión de un reporte del estado nutricional de gestantes por distrito, provincia y departamento.
17. Generar reportes con gráficos estadísticos de niños.	17. El sistema debe permitir generar reportes con gráficos estadísticos del estado nutricional de niños.

18. Generar reportes con gráficos estadísticos de gestantes.	18. El sistema debe permitir generar reportes con gráficos estadísticos del estado nutricional de gestantes.
--	--

4.1.1.6. DESCRIPCIÓN DE CASOS DE USO

Para la presentación del desarrollo de la aplicación se seleccionaron los siguientes casos de uso.

4.1.1.6.1. CASO DE USO: Registrar niño

Tabla N° 4.4: Caso de uso: Registrar niño

CASO DE USO: Registrar niño
ACTOR: Trabajador
DESCRIPCIÓN: Este caso de uso permite al trabajador registrar a un niño, teniendo en cuenta todos los datos del formato de registro diario del niño menor de 5 años.
ESCENARIO NORMAL
El trabajador hace clic en el botón nuevo, el sistema muestra el formulario donde se registrará la información para poder ingresar los datos del niño, luego hacer clic en el botón guardar, el sistema muestra un mensaje de confirmación indicando que los datos se guardaron correctamente.
ESCENARIO ALTERNO
El trabajador no ingresó todos los datos requeridos: El sistema muestra un mensaje indicando los datos que falta ingresar.
El trabajador ingresó datos incorrectos: El sistema muestra un mensaje indicando la característica de los datos a ingresar.
El trabajador hace clic en el botón cancelar: El sistema deshabilita todos los botones a excepción de los botones nuevo y modificar.

Error de sistema al registrar datos: El sistema muestra un mensaje de error generado.
PRECONDICIONES: El trabajador debe haberse autenticado e iniciado sesión previamente y estar en una sesión activa.
POSTCONDICIONES: Se administró correctamente el registro de niño.

4.1.1.6.2. CASO DE USO: Modificar niño

Tabla N° 4.5: Caso de uso: Mantener niño

CASO DE USO: Modificar niño.
ACTOR: Trabajador
DESCRIPCIÓN: Este caso de uso permite al trabajador modificar los datos del registro de un niño.
ESCENARIO NORMAL
El trabajador selecciona un tipo de búsqueda e ingresa el criterio de búsqueda requerido, seguidamente hace clic en el botón buscar, luego el sistema muestra una grilla con los datos obtenidos por las coincidencias de búsqueda, el trabajador selecciona un registro de la grilla y el sistema muestra la página niño habilitando las cajas de texto para poder realizar modificaciones en los datos del registro seleccionado, el trabajador hace clic en el botón actualizar, el sistema muestra un mensaje de confirmación de actualización indicando que los datos se modificaron correctamente.
ESCENARIO ALTERNO
El trabajador no ingresó todos los datos requeridos: El sistema muestra un mensaje indicando los datos que falta ingresar.
El trabajador ingresó datos incorrectos: El sistema muestra un mensaje indicando la característica específica de los datos a ingresar.

<p>El trabajador hace clic en el botón cancelar: El sistema cierra la página de mantener niño.</p>
<p>Error de sistema al registrar datos: El sistema muestra un mensaje de error generado.</p>
<p>PRECONDICIONES: El trabajador debe haberse autenticado e iniciado sesión previamente y estar en una sesión activa.</p>
<p>POSTCONDICIONES: Se dio mantenimiento a un registro de niño.</p>

4.1.1.6.3. CASO DE USO: Emitir reporte del estado nutricional de niños

Tabla N° 4.6: Caso de uso: Emitir reporte del estado nutricional de niños.

CASO DE USO: Emitir reporte del estado nutricional de niños
<p>ACTOR: Trabajador</p>
<p>DESCRIPCIÓN: Este caso de uso permite al trabajador emitir reporte del estado nutricional de niños.</p>
<p>ESCENARIO NORMAL</p> <p>El trabajador hace clic en la opción reporte, el sistema muestra la página de reporte de niños, el trabajador selecciona un tipo de búsqueda e ingresa el criterio de búsqueda requerido, seguidamente hace clic en el botón buscar, el sistema muestra una tabla con las coincidencias de la búsqueda, el trabajador hace clic en la opción exportar y selecciona el formato del archivo a exportar (Excel o PDF), el sistema muestra el reporte seleccionado por el trabajador en un tipo de archivo plano.</p>
<p>ESCENARIO ALTERNO</p> <p>El sistema no encuentra ninguna coincidencia: El sistema muestra un mensaje “Reporte no encontrado.”</p> <p>El trabajador no ingresa los datos requeridos: El sistema muestra un mensaje de error indicando que se debe ingresar al menos un dato de búsqueda.</p>

<p>El trabajador ingresó datos incorrectos: El sistema muestra un mensaje indicando la característica específica de los datos a ingresar.</p> <p>El trabajador hace clic en el botón cancelar: El sistema resetea los criterios de búsqueda para realizar otro reporte.</p> <p>El trabajador hace clic en el link salir: El sistema cierra la página de reporte de niños.</p> <p>Error de sistema al generar reporte de datos: El sistema muestra un mensaje de error generado.</p>
<p>PRECONDICIONES: El trabajador debe haberse autenticado e iniciado sesión previamente y estar en una sesión activa.</p>
<p>POSTCONDICIONES: Se generó correctamente el reporte del estado nutricional de niños.</p>

4.1.2. HITO ETAPA 1: REVISIÓN DE REQUISITOS

A partir de la información obtenida en la ETAPA 1, en este ítem se presenta la revisión de los requisitos funcionales, revisión requisitos no funcionales, revisión del modelo de dominio y revisión de casos de uso.

4.1.2.1. REQUISITOS FUNCIONALES REVISADOS

Tabla N° 4.7: Requisitos funcionales revisados

REVISIÓN DE REQUISITOS FUNCIONALES
<p>ReqF01. El sistema debe ser accedido a éste mediante un nombre de usuario y contraseña, según el rol que cumple cada empleado, dependiendo de su área de trabajo.</p>
<p>ReqF02. El sistema debe permitir registrar una cuenta de usuario para los empleados, según Establecimiento, Microrred, Red y DIRESA a la que pertenece.</p>
<p>ReqF03. El sistema debe permitir modificar una cuenta de usuario.</p>
<p>ReqF04. El sistema debe permitir eliminar una cuenta de usuario.</p>

ReqF05. El sistema debe permitir buscar una cuenta de usuario.
ReqF06. El sistema debe permitir mostrar un reporte de las cuentas de usuarios.
ReqF07. El sistema debe permitir registrar datos de la atención de un paciente .
ReqF08. El sistema debe permitir registrar el tipo de documento del paciente.
ReqF09. El sistema debe permitir registrar los datos de la atención como tipo de paciente a un niño menor de 5 años, para lo cual se debe registrar datos de la semana epidemiológica (según periodo epidemiológico), programa de afiliación .
ReqF10. El sistema debe permitir modificar los datos de la atención de un niño.
ReqF11. El sistema debe permitir eliminar la atención de un niño.
ReqF12. El sistema debe permitir realizar búsqueda de la atención de un niño.
ReqF13. El sistema debe permitir identificar los problemas del estado nutricional de un niño como el tipo de desnutrición (desnutrición crónica, desnutrición aguda, desnutrición global, sobrepeso, obesidad) y grado de desnutrición (Normal, severa, baja)
ReqF14. El sistema debe permitir mostrar a través de un mapa cuantitativo el estado nutricional de niños.
ReqF15. El sistema debe permitir generar gráficos estadísticos del estado nutricional de niños.
ReqF16. El sistema debe permitir mostrar reporte del estado nutricional de niños según región, provincia, distrito, localidad y periodo.
ReqF17. El sistema debe permitir mostrar reporte del estado nutricional de niños según diresa.
ReqF18. El sistema debe permitir mostrar reporte del estado nutricional de niños según red de salud.
ReqF19. El sistema debe permitir mostrar reporte del estado nutricional de niños según microrred de salud.

ReqF20. El sistema debe permitir mostrar reporte del estado nutricional de niños según establecimiento de salud.
ReqF21. El sistema debe permitir generar reporte del estado nutricional de niños según grupo etáreo.
ReqF22. El sistema debe permitir registrar los datos de la atención como tipo de paciente a una gestante que tenga 13 semanas de embarazo, para lo cual se debe registrar también el tipo de embarazo .
ReqF23. El sistema debe permitir modificar los datos de la atención de una gestante.
ReqF24. El sistema debe permitir eliminar la atención de una gestante.
ReqF25. El sistema debe permitir realizar búsqueda de la atención de una gestante.
ReqF26. El sistema debe permitir identificar los problemas del estado nutricional de una gestante como es el caso de normal, sobrepeso, déficit de peso y tipo de anemia .
ReqF27. El sistema debe permitir mostrar a través de un mapa cuantitativo el estado nutricional de gestantes.
ReqF28. El sistema debe permitir generar gráficos estadísticos del estado nutricional de gestantes.
ReqF29. El sistema debe permitir mostrar reporte del estado nutricional de gestantes según región, provincia, distrito, localidad y periodo.
ReqF30. El sistema debe permitir mostrar reporte del estado nutricional de gestantes según diresa.
ReqF31. El sistema debe permitir mostrar reporte del estado nutricional de gestantes según red de salud.
ReqF32. El sistema debe permitir mostrar reporte del estado nutricional de gestantes según microrred de salud.
ReqF33. El sistema debe permitir mostrar reporte del estado nutricional de gestantes

según establecimiento de salud.
ReqF34. El sistema debe permitir generar reporte del estado nutricional de gestantes según grupo etáreo.
ReqF35. El sistema debe permitir registrar los datos de una DIRESA.
ReqF36. El sistema debe permitir modificar los datos de una DIRESA.
ReqF37. El sistema debe permitir eliminar una DIRESA.
ReqF38. El sistema debe permitir realizar búsqueda de una DIRESA.
ReqF39. El sistema debe permitir mostrar reporte de las DIRESA's.
ReqF40. El sistema debe permitir registrar los datos de una Red de Salud.
ReqF41. El sistema debe permitir modificar los datos de una Red de Salud.
ReqF42. El sistema debe permitir eliminar una Red de Salud.
ReqF43. El sistema debe permitir realizar búsqueda de una Red de Salud.
ReqF44. El sistema debe permitir mostrar reporte de las Redes de Salud.
ReqF45. El sistema debe permitir registrar los datos de una Microrred de Salud.
ReqF46. El sistema debe permitir modificar los datos de una Microrred de Salud.
ReqF47. El sistema debe permitir eliminar una Microrred de Salud.
ReqF48. El sistema debe permitir realizar búsqueda una Microrred de Salud.
ReqF49. El sistema debe permitir mostrar reporte de las Microrredes de Salud.
ReqF50. El sistema debe permitir registrar los datos de un Establecimiento de Salud.
ReqF51. El sistema debe permitir modificar los datos un Establecimiento de Salud.
ReqF52. El sistema debe permitir eliminar un Establecimiento de Salud.

ReqF53. El sistema debe permitir realizar búsqueda un Establecimiento de Salud.
ReqF54. El sistema debe permitir generar reporte de los Establecimientos de Salud.
ReqF55. El sistema debe permitir registrar los datos de una semana epidemiológica.
ReqF56. El sistema debe permitir modificar los datos una semana epidemiológica.
ReqF57. El sistema debe permitir eliminar una semana epidemiológica.
ReqF58. El sistema debe permitir realizar búsqueda de una semana epidemiológica.
ReqF59. El sistema debe permitir generar reporte de las semanas epidemiológicas.
ReqF60. El sistema debe permitir registrar los datos de un programa de afiliación.
ReqF61. El sistema debe permitir modificar los datos de un programa de afiliación.
ReqF62. El sistema debe permitir eliminar un programa de afiliación.
ReqF63. El sistema debe permitir generar reporte de los programas de afiliación.
ReqF64. El sistema debe permitir realizar búsqueda de un programa de afiliación.
ReqF65. El sistema debe permitir registrar los datos del tipo de embarazo.
ReqF66. El sistema debe permitir modificar los datos del tipo de embarazo.
ReqF67. El sistema debe permitir eliminar un tipo de embarazo.
ReqF68. El sistema debe permitir mostrar reporte de los tipos de embarazo.
ReqF69. El sistema debe permitir realizar búsqueda de los tipos de embarazo.
ReqF70. El sistema debe permitir registrar los datos de región.
ReqF71. El sistema debe permitir modificar los datos de región.
ReqF72. El sistema debe permitir eliminar región.
ReqF73. El sistema debe permitir mostrar reporte de las regiones.

ReqF74. El sistema debe permitir realizar búsqueda de región.
ReqF75. El sistema debe permitir registrar los datos de provincia.
ReqF76. El sistema debe permitir modificar los datos de provincia.
ReqF77. El sistema debe permitir eliminar provincia.
ReqF78. El sistema debe permitir mostrar reporte de las provincias.
ReqF79. El sistema debe permitir realizar búsqueda de provincia.
ReqF80. El sistema debe permitir registrar los datos de distrito.
ReqF81. El sistema debe permitir modificar los datos de distrito.
ReqF82. El sistema debe permitir eliminar distrito.
ReqF83. El sistema debe permitir mostrar reporte de los distritos.
ReqF84. El sistema debe permitir realizar búsqueda de distrito.
ReqF85. El sistema debe permitir registrar los datos de localidad.
ReqF86. El sistema debe permitir modificar los datos de localidad.
ReqF87. El sistema debe permitir eliminar localidad.
ReqF88. El sistema debe permitir mostrar reporte de las localidades.
ReqF89. El sistema debe permitir realizar búsqueda de localidad.
ReqF90. El sistema debe permitir registrar los datos del empleado.
ReqF91. El sistema debe permitir modificar los datos del empleado.
ReqF92. El sistema debe permitir eliminar un empleado.
ReqF93. El sistema debe permitir mostrar reporte de los empleados.
ReqF94. El sistema debe permitir realizar búsqueda de los empleados.

ReqF95 El sistema debe permitir registrar el periodo epidemiológico.
ReqF96. El sistema debe permitir modificar periodo epidemiológico.
ReqF97. El sistema debe permitir eliminar un periodo epidemiológico.
ReqF98. El sistema debe permitir mostrar reporte de los periodos epidemiológicos.
ReqF99. El sistema debe permitir realizar búsqueda de los periodos epidemiológicos.
ReqF100. El sistema debe permitir registrar el rol del empleado.
ReqF101. El sistema debe permitir modificar el rol del empleado.
ReqF102. El sistema debe permitir eliminar un rol del empleado.
ReqF103. El sistema debe permitir mostrar reporte de los roles del empleado.
ReqF104. El sistema debe permitir realizar búsqueda de los roles del empleado.
ReqF105. El sistema debe permitir registrar el área de trabajo del empleado.
ReqF106. El sistema debe permitir modificar el área de trabajo del empleado.
ReqF107. El sistema debe permitir eliminar un área de trabajo del empleado.
ReqF108. El sistema debe permitir mostrar reporte de las áreas de trabajo del empleado.
ReqF109. El sistema debe permitir realizar búsqueda de las áreas de trabajo del empleado.
ReqF110. El sistema debe permitir modificar el tipo de documento del paciente.
ReqF111. El sistema debe permitir eliminar el tipo de documento del paciente.
ReqF112. El sistema debe permitir mostrar reporte de los tipos de documento del paciente.
ReqF113. El sistema debe permitir realizar búsqueda de los tipos de documento del paciente.

4.1.2.2. REQUISITOS NO FUNCIONALES REVISADOS

Tabla N° 4.8: Requisitos no funcionales revisados

REVISIÓN DE REQUISITOS NO FUNCIONALES
ReqNF01. La aplicación estará basada en entorno web, debe tener una arquitectura técnica y codificación usando estándares que permita realizar el mantenimiento de la aplicación periódicamente.
ReqNF02. El sistema debe contener niveles de seguridad adecuados y acceso a la información por roles y grupos de usuarios como administrador, digitador.
ReqNF03. En la fase de desarrollo de la aplicación se usará el patrón de arquitectura “n capas” para separar cada fase de ejecución de la aplicación.
ReqNF04. Para el desarrollo de la aplicación web se empleará la tecnología ASP.NET para proveer mayor agilidad en el desarrollo de aplicaciones.
ReqNF05. En el desarrollo de la aplicación se usarán estándares de nomenclatura para las clases, objetos, vistas, controladores, etc.; que permitan una adecuada mantenibilidad de la aplicación en las etapas de pruebas unitarias y post-producción.
ReqNF06. El sistema debe ser robusto que contemple la distribución de carga a nivel de componentes siendo altamente escalable.
ReqNF07. La aplicación debe ser de interfaz amigable, moderno e intuitivo para con los usuarios, mostrando ayudas que guiarán al usuario en su correcto uso del mismo.
ReqNF08. El sistema mejorará los tiempos de respuesta y la seguridad en el procesamiento de la información, tanto para el procesamiento de datos así como para fines de auditoría de los mismos.
ReqNF09. El sistema debe garantizar la continuidad del servicio en los momentos pico del procesamiento de datos con óptimos tiempos de respuesta (debe soportar alrededor de 50 usuarios como mínimo).
ReqNF10. El sistema debe garantizar que la eliminación de un registro de la atención

(niño, gestante, etc) debe ser de manera lógica y no física. Por lo tanto el registro de la atención (niño, gestante) debe tener un estado (activo o inactivo).

De los requisitos funcionales y no funcionales revisados se consiguió el siguiente **listado de términos para clases del modelo de dominio:**

- Usuario
- Rol
- Empleado
- Area de trabajo
- Establecimiento
- Microrred
- Red
- DIRESA
- Atención
- Tipo de documento
- Paciente
- Tipo de paciente
- Niño
- Semana epidemiológica
- Periodo epidemiológico
- Programa de afiliación
- Estado nutricional
- Tipo de desnutrición
- Desnutrición crónica
- Desnutrición aguda
- Desnutrición global
- Sobrepeso
- Obesidad
- Anemia
- Grado de desnutrición
- Mapa cuantitativo
- Región

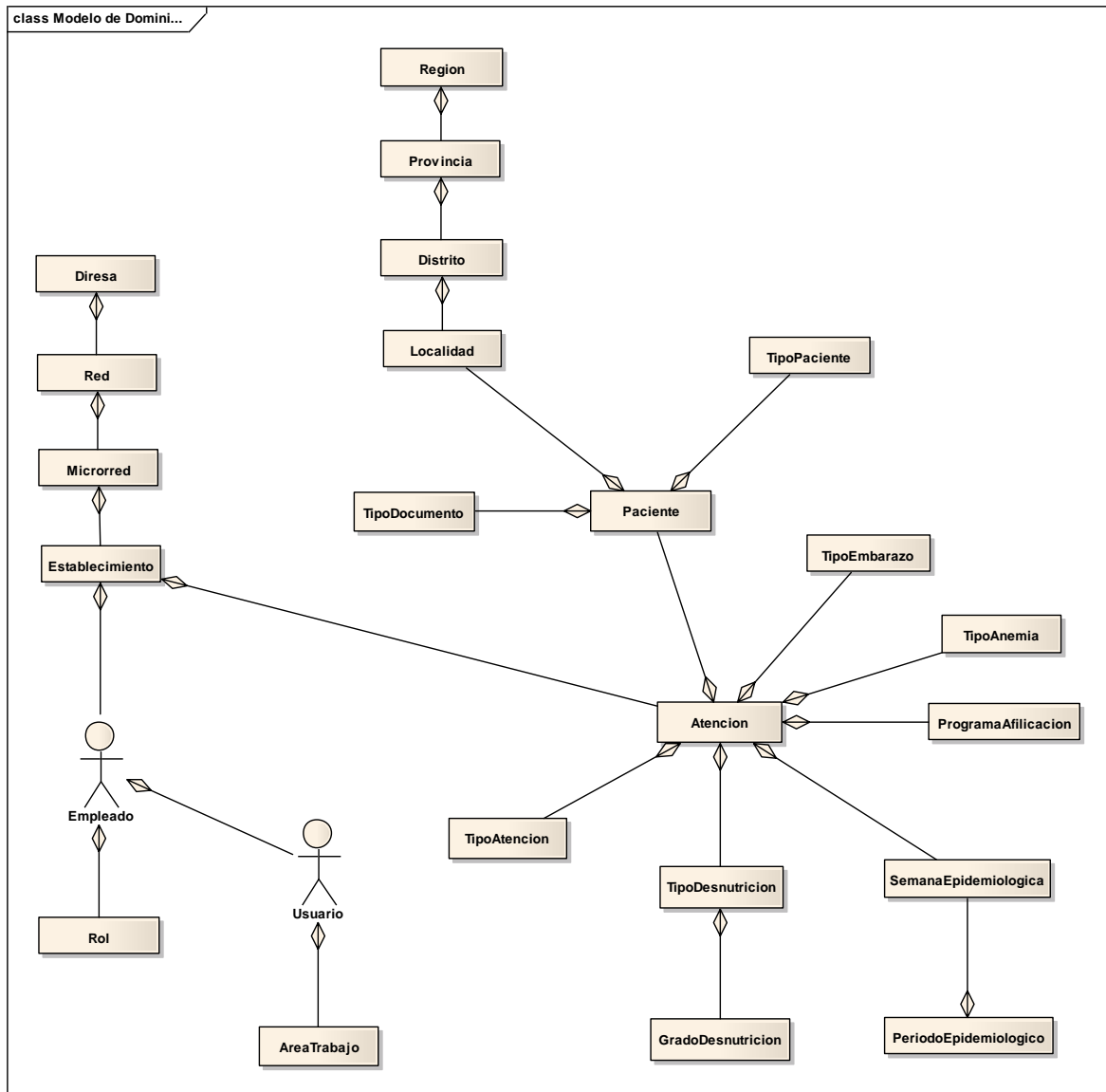
- Provincia
- Distrito
- Localidad
- Grupo etáreo
- Gestante
- Tipo de embarazo
- Sobrepeso
- Normal
- Déficit de peso
- Tipo de anemia

Desambiguación de términos:

- El término **tipo de paciente**, ya engloba a lo que es **niño** y **gestante**; por tanto sólo usaremos el término **tipo de paciente**.
- El término **tipo de desnutrición**, ya engloba a lo que es **desnutrición crónica, aguda, global**; por tanto sólo usaremos el término **tipo de desnutrición**.
- El término **grado de desnutrición**, ya engloba a lo que es **normal, severa y baja**; por tanto sólo usaremos el término **grado de desnutrición**.
- El término **estado nutricional** ya no lo emplearemos porque ya se reflejará e los reportes según los términos tipo y grado de desnutrición.
- El término **tipo de anemia**, ya engloba a lo que es **leve, moderada y severa**; por tanto sólo usaremos el término **tipo de anemia**.
- El término **mapa cuantitativo** es el tipo de visualización de un tipo de reporte, por tanto no lo emplearemos.
- El término **grupo etáreo**, lo descartamos porque es una clasificación de grupos según edades.

4.1.2.3. REVISIÓN DE MODELO DE DOMINIO

Figura N° 4.5: Modelo de dominio general revisado.



4.1.2.4. REVISIÓN DE CASOS DE USO

Tabla N° 4.9: Revisión de casos de uso

CASO DE USO	REQUISITOS FUNCIONALES
<p style="text-align: center;">Iniciar sesión de usuario</p>	<p>ReqF01. El sistema debe ser accedido a éste mediante un nombre de usuario y contraseña, según el rol que cumple cada empleado, dependiendo de su área de trabajo.</p>
<p style="text-align: center;">Registrar cuenta de usuario</p>	<p>ReqF02. El sistema debe permitir registrar una cuenta de usuario para los empleados, según Establecimiento, Microrred, Red y DIRESA a la que pertenece.</p>
<p style="text-align: center;">Modificar cuenta de usuario</p>	<p>ReqF03. El sistema debe permitir modificar una cuenta de usuario.</p>
<p style="text-align: center;">Eliminar cuenta de usuario</p>	<p>ReqF04. El sistema debe permitir eliminar una cuenta de usuario.</p>
<p style="text-align: center;">Buscar cuenta de usuario</p>	<p>ReqF05. El sistema debe permitir buscar una cuenta de usuario.</p>
<p style="text-align: center;">Emitir reporte de usuarios</p>	<p>ReqF06. El sistema debe permitir mostrar un reporte de las cuentas de usuarios.</p>
<p style="text-align: center;">Registrar atención niño</p>	<p>ReqF07. El sistema debe permitir registrar datos de la atención de un paciente.</p>

<p style="text-align: center;">Registrar tipo documento</p>	<p>ReqF08. El sistema debe permitir registrar el tipo de documento del paciente.</p>
<p style="text-align: center;">Registrar atención niño</p>	<p>ReqF09. El sistema debe permitir registrar los datos de la atención como tipo de paciente a un niño menor de 5 años, para lo cual se debe registrar datos de la semana epidemiológica (según periodo epidemiológico), programa de afiliación.</p>
<p style="text-align: center;">Modificar atención niño</p>	<p>ReqF10. El sistema debe permitir modificar los datos de la atención de un niño.</p>
<p style="text-align: center;">Eliminar atención niño</p>	<p>ReqF11. El sistema debe permitir eliminar la atención de un niño.</p>
<p style="text-align: center;">Buscar atención niño</p>	<p>ReqF12. El sistema debe permitir realizar búsqueda de la atención de un niño.</p>
<p style="text-align: center;">Emitir reporte del estado nutricional de niños</p>	<p>ReqF13. El sistema debe permitir identificar los problemas del estado nutricional de un niño como el tipo de desnutrición (desnutrición crónica, desnutrición aguda, desnutrición global, sobrepeso, obesidad) y grado de desnutrición (Normal, severa, baja)</p>
<p style="text-align: center;">Emitir reporte del estado nutricional de niños a través de un mapa cuantitativo</p>	<p>ReqF14. El sistema debe permitir mostrar a través de un mapa cuantitativo el estado nutricional de niños.</p>

<p>Generar gráficos estadísticos del estado nutricional de niños</p>	<p>ReqF15. El sistema debe permitir generar gráficos estadísticos del estado nutricional de niños.</p>
<p>Emitir reporte del estado nutricional de niños según región, provincia, distrito, localidad y periodo</p>	<p>ReqF16. El sistema debe permitir mostrar reporte del estado nutricional de niños según región, provincia, distrito, localidad y periodo.</p>
<p>Emitir reporte del estado nutricional de niños según región, provincia, distrito, localidad y periodo</p>	<p>ReqF17. El sistema debe permitir mostrar reporte del estado nutricional de niños según diresa.</p>
<p>Emitir reporte del estado nutricional de niños según red de salud</p>	<p>ReqF18. El sistema debe permitir mostrar reporte del estado nutricional de niños según red de salud.</p>
<p>Emitir reporte del estado nutricional de niños según microrred de salud</p>	<p>ReqF19. El sistema debe permitir mostrar reporte del estado nutricional de niños según microrred de salud.</p>
<p>Emitir reporte del estado nutricional de niños según establecimiento de salud</p>	<p>ReqF20. El sistema debe permitir mostrar reporte del estado nutricional de niños según establecimiento de salud.</p>
<p>Emitir reporte del estado nutricional de niños según grupo étnico</p>	<p>ReqF21. El sistema debe permitir generar reporte del estado nutricional de niños según grupo étnico.</p>

<p style="text-align: center;">Registrar atención gestante</p>	<p>ReqF22. El sistema debe permitir registrar los datos de la atención como tipo de paciente a una gestante que tenga 13 semanas de embarazo, para lo cual se debe registrar también el tipo de embarazo.</p>
<p style="text-align: center;">Modificar atención gestante</p>	<p>ReqF23. El sistema debe permitir modificar los datos de la atención de una gestante.</p>
<p style="text-align: center;">Eliminar atención gestante</p>	<p>ReqF24. El sistema debe permitir eliminar la atención de una gestante.</p>
<p style="text-align: center;">Buscar atención gestante</p>	<p>ReqF25. El sistema debe permitir realizar búsqueda de la atención de una gestante.</p>
<p style="text-align: center;">Emitir reporte del estado nutricional de gestantes</p>	<p>ReqF26. El sistema debe permitir identificar los problemas del estado nutricional de una gestante como es el caso de normal, sobrepeso, déficit de peso y tipo de anemia.</p>
<p style="text-align: center;">Emitir reporte del estado nutricional de gestantes a través de un mapa cuantitativo</p>	<p>ReqF27. El sistema debe permitir mostrar a través de un mapa cuantitativo el estado nutricional de gestantes.</p>
<p style="text-align: center;">Generar gráficos estadísticos del estado nutricional de gestantes</p>	<p>ReqF28. El sistema debe permitir generar gráficos estadísticos del estado nutricional de gestantes.</p>
<p style="text-align: center;">Emitir reporte del estado nutricional de gestantes región, provincia, distrito, localidad y periodo</p>	<p>ReqF29. El sistema debe permitir mostrar reporte del estado nutricional de gestantes según región, provincia, distrito, localidad y periodo.</p>

<p>Emitir reporte del estado nutricional de gestantes según diresa</p>	<p>ReqF30. El sistema debe permitir mostrar reporte del estado nutricional de gestantes según diresa.</p>
<p>Emitir reporte del estado nutricional de gestantes según red de salud</p>	<p>ReqF31. El sistema debe permitir mostrar reporte del estado nutricional de gestantes según red de salud.</p>
<p>Emitir reporte del estado nutricional de gestantes según microrred de salud</p>	<p>ReqF32. El sistema debe permitir mostrar reporte del estado nutricional de gestantes según microrred de salud.</p>
<p>Emitir reporte del estado nutricional de gestantes según establecimiento de salud</p>	<p>ReqF33. El sistema debe permitir mostrar reporte del estado nutricional de gestantes según establecimiento de salud.</p>
<p>Emitir reporte del estado nutricional de gestantes según grupo étnico</p>	<p>ReqF34. El sistema debe permitir generar reporte del estado nutricional de gestantes según grupo étnico.</p>
<p>Registrar diresa</p>	<p>ReqF35. El sistema debe permitir registrar los datos de una DIRESA.</p>
<p>Modificar diresa</p>	<p>ReqF36. El sistema debe permitir modificar los datos de una DIRESA.</p>
<p>Eliminar diresa</p>	<p>ReqF37. El sistema debe permitir eliminar una DIRESA.</p>

<p style="text-align: center;">Buscar diresa</p>	<p>ReqF38. El sistema debe permitir realizar búsqueda de una DIRESA.</p>
<p style="text-align: center;">Emitir reporte de diresas</p>	<p>ReqF39. El sistema debe permitir mostrar reporte de las DIRESA's.</p>
<p style="text-align: center;">Registrar red</p>	<p>ReqF40. El sistema debe permitir registrar los datos de una Red de Salud.</p>
<p style="text-align: center;">Modificar red</p>	<p>ReqF41. El sistema debe permitir modificar los datos de una Red de Salud.</p>
<p style="text-align: center;">Eliminar red</p>	<p>ReqF42. El sistema debe permitir eliminar una Red de Salud.</p>
<p style="text-align: center;">Buscar red</p>	<p>ReqF43. El sistema debe permitir realizar búsqueda de una Red de Salud.</p>
<p style="text-align: center;">Emitir reporte de redes</p>	<p>ReqF44. El sistema debe permitir mostrar reporte de las Redes de Salud.</p>
<p style="text-align: center;">Registrar microrred</p>	<p>ReqF45. El sistema debe permitir registrar los datos de una Microrred de Salud.</p>
<p style="text-align: center;">Modificar microrred</p>	<p>ReqF46. El sistema debe permitir modificar los datos de una Microrred de Salud.</p>

<p style="text-align: center;">Eliminar microrred</p>	<p>ReqF47. El sistema debe permitir eliminar una Microrred de Salud.</p>
<p style="text-align: center;">Buscar microrred</p>	<p>ReqF48. El sistema debe permitir realizar búsqueda una Microrred de Salud.</p>
<p style="text-align: center;">Emitir reporte de microrredes</p>	<p>ReqF49. El sistema debe permitir mostrar reporte de las Microrredes de Salud.</p>
<p style="text-align: center;">Registrar establecimiento</p>	<p>ReqF50. El sistema debe permitir registrar los datos de un Establecimiento de Salud.</p>
<p style="text-align: center;">Modificar establecimiento</p>	<p>ReqF51. El sistema debe permitir modificar los datos un Establecimiento de Salud.</p>
<p style="text-align: center;">Eliminar establecimiento</p>	<p>ReqF52. El sistema debe permitir eliminar un Establecimiento de Salud.</p>
<p style="text-align: center;">Buscar establecimiento</p>	<p>ReqF53. El sistema debe permitir realizar búsqueda un Establecimiento de Salud.</p>
<p style="text-align: center;">Emitir reporte de establecimientos</p>	<p>ReqF54. El sistema debe permitir generar reporte de los Establecimientos de Salud.</p>
<p style="text-align: center;">Registrar semana epidemiológica</p>	<p>ReqF55. El sistema debe permitir registrar los datos de una semana epidemiológica.</p>

<p>Modificar semana epidemiológica</p>	<p>ReqF56. El sistema debe permitir modificar los datos una semana epidemiológica.</p>
<p>Eliminar semana epidemiológica</p>	<p>ReqF57. El sistema debe permitir eliminar una semana epidemiológica.</p>
<p>Buscar semana epidemiológica</p>	<p>ReqF58. El sistema debe permitir realizar búsqueda de una semana epidemiológica.</p>
<p>Emitir reporte de semanas epidemiológicas</p>	<p>ReqF59. El sistema debe permitir generar reporte de las semanas epidemiológicas.</p>
<p>Registrar programa afiliación</p>	<p>ReqF60. El sistema debe permitir registrar los datos de un programa de afiliación.</p>
<p>Modificar programa afiliación</p>	<p>ReqF61. El sistema debe permitir modificar los datos de un programa de afiliación.</p>
<p>Eliminar programa afiliación</p>	<p>ReqF62. El sistema debe permitir eliminar un programa de afiliación.</p>
<p>Emitir reporte de programas de afiliación</p>	<p>ReqF63. El sistema debe permitir generar reporte de los programas de afiliación.</p>
<p>Buscar programa afiliación</p>	<p>ReqF64. El sistema debe permitir realizar búsqueda de un programa de afiliación.</p>





<p style="text-align: center;">Registrar tipo embarazo</p>	<p>ReqF65. El sistema debe permitir registrar los datos del tipo de embarazo.</p>
<p style="text-align: center;">Modificar tipo embarazo</p>	<p>ReqF66. El sistema debe permitir modificar los datos del tipo de embarazo.</p>
<p style="text-align: center;">Eliminar tipo embarazo</p>	<p>ReqF67. El sistema debe permitir eliminar un tipo de embarazo.</p>
<p style="text-align: center;">Emitir reporte de tipos de embarazo</p>	<p>ReqF68. El sistema debe permitir mostrar reporte de los tipos de embarazo.</p>
<p style="text-align: center;">Buscar tipo embarazo</p>	<p>ReqF69. El sistema debe permitir realizar búsqueda de los tipos de embarazo.</p>
<p style="text-align: center;">Registrar región</p>	<p>ReqF70. El sistema debe permitir registrar los datos de región.</p>
<p style="text-align: center;">Modificar región</p>	<p>ReqF71. El sistema debe permitir modificar los datos de región.</p>
<p style="text-align: center;">Eliminar región</p>	<p>ReqF72. El sistema debe permitir eliminar región.</p>
<p style="text-align: center;">Emitir reporte de regiones</p>	<p>ReqF73. El sistema debe permitir mostrar reporte de las regiones.</p>

<p style="text-align: center;">Buscar región</p>	<p>ReqF74. El sistema debe permitir realizar búsqueda de región.</p>
<p style="text-align: center;">Registrar provincia</p>	<p>ReqF75. El sistema debe permitir registrar los datos de provincia.</p>
<p style="text-align: center;">Modificar provincia</p>	<p>ReqF76. El sistema debe permitir modificar los datos de provincia.</p>
<p style="text-align: center;">Eliminar provincia</p>	<p>ReqF77. El sistema debe permitir eliminar provincia.</p>
<p style="text-align: center;">Emitir reporte de provincias</p>	<p>ReqF78. El sistema debe permitir mostrar reporte de las provincias.</p>
<p style="text-align: center;">Buscar provincia</p>	<p>ReqF79. El sistema debe permitir realizar búsqueda de provincia.</p>
<p style="text-align: center;">Registrar distrito</p>	<p>ReqF80. El sistema debe permitir registrar los datos de distrito.</p>
<p style="text-align: center;">Modificar distrito</p>	<p>ReqF81. El sistema debe permitir modificar los datos de distrito.</p>
<p style="text-align: center;">Eliminar distrito</p>	<p>ReqF82. El sistema debe permitir eliminar distrito.</p>

<p>Emitir reporte de distritos</p>	<p>ReqF83. El sistema debe permitir mostrar reporte de los distritos.</p>
<p>Buscar distrito</p>	<p>ReqF84. El sistema debe permitir realizar búsqueda de distrito.</p>
<p>Registrar localidad</p>	<p>ReqF85. El sistema debe permitir registrar los datos de localidad.</p>
<p>Modificar localidad</p>	<p>ReqF86. El sistema debe permitir modificar los datos de localidad.</p>
<p>Eliminar localidad</p>	<p>ReqF87. El sistema debe permitir eliminar localidad.</p>
<p>Emitir reporte de localidades</p>	<p>ReqF88. El sistema debe permitir mostrar reporte de las localidades.</p>
<p>Buscar localidad</p>	<p>ReqF89. El sistema debe permitir realizar búsqueda de localidad.</p>
<p>Registrar empleado</p>	<p>ReqF90 El sistema debe permitir registrar los datos del empleado.</p>
<p>Modificar empleado</p>	<p>ReqF91. El sistema debe permitir modificar los datos del empleado.</p>

<p style="text-align: center;">Eliminar empleado</p>	<p>ReqF92. El sistema debe permitir eliminar un empleado.</p>
<p style="text-align: center;">Emitir reporte de empleados</p>	<p>ReqF93. El sistema debe permitir mostrar reporte de los empleado.</p>
<p style="text-align: center;">Buscar empleado</p>	<p>ReqF94. El sistema debe permitir realizar búsqueda de los empleado.</p>
<p style="text-align: center;">Registrar periodo epidemiológico</p>	<p>ReqF95 El sistema debe permitir registrar el periodo epidemiológico.</p>
<p style="text-align: center;">Modificar periodo epidemiológico</p>	<p>ReqF96. El sistema debe permitir modificar periodo epidemiológico.</p>
<p style="text-align: center;">Eliminar periodo epidemiológico</p>	<p>ReqF97. El sistema debe permitir eliminar un periodo epidemiológico.</p>
<p style="text-align: center;">Emitir reporte de periodos epidemiológicos</p>	<p>ReqF98. El sistema debe permitir mostrar reporte de los periodos epidemiológicos.</p>
<p style="text-align: center;">Buscar periodo epidemiológico</p>	<p>ReqF99. El sistema debe permitir realizar búsqueda de los periodos epidemiológicos.</p>
<p style="text-align: center;">Registrar rol</p>	<p>ReqF100. El sistema debe permitir registrar el rol del empleado.</p>

<p style="text-align: center;">Modificar rol</p>	<p>ReqF101. El sistema debe permitir modificar el rol del empleado.</p>
<p style="text-align: center;">Eliminar rol</p>	<p>ReqF102. El sistema debe permitir eliminar un rol del empleado.</p>
<p style="text-align: center;">Emitir reporte de roles</p>	<p>ReqF103. El sistema debe permitir mostrar reporte de los roles del empleado.</p>
<p style="text-align: center;">Buscar rol</p>	<p>ReqF104. El sistema debe permitir realizar búsqueda de los roles del empleado.</p>
<p style="text-align: center;">Registrar área trabajo</p>	<p>ReqF105. El sistema debe permitir registrar el área de trabajo del empleado.</p>
<p style="text-align: center;">Modificar área trabajo</p>	<p>ReqF106. El sistema debe permitir modificar el área de trabajo del empleado.</p>
<p style="text-align: center;">Eliminar área trabajo</p>	<p>ReqF107. El sistema debe permitir eliminar un área de trabajo del empleado.</p>
<p style="text-align: center;">Emitir reporte de áreas de trabajo</p>	<p>ReqF108. El sistema debe permitir mostrar reporte de las áreas de trabajo del empleado.</p>
<p style="text-align: center;">Buscar área trabajo</p>	<p>ReqF109. El sistema debe permitir realizar búsqueda de las áreas de trabajo del empleado.</p>

	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> ReqF110. El sistema debe permitir modificar el tipo de documento del paciente. </div>
	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> ReqF111. El sistema debe permitir eliminar el tipo de documento del paciente. </div>
	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> ReqF112. El sistema debe permitir mostrar reporte de los tipos de documento del paciente. </div>
	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> ReqF113. El sistema debe permitir realizar búsqueda de los tipos de documento del paciente. </div>

4.1.2.4.1. CASO DE USO: Registrar atención niño

Tabla N° 4.10: Revisión del caso de uso registrar atención niño.

CASO DE USO: Registrar atención niño
ACTOR: Usuario
DESCRIPCIÓN: Este caso de uso permite al usuario registrar la atención de un niño, teniendo en cuenta todos los datos del formato de registro diario del niño menor de 5 años.
ESCENARIO NORMAL
<ol style="list-style-type: none"> 1. El usuario hace clic en la opción Atención Niños. 2. El sistema muestra la página Niño. 3. El usuario hace clic en el botón Nuevo niño. 4. El sistema muestra la página Detalle niño.

5. El usuario ingresa los datos de la atención del niño como fecha de atención, semana epidemiológica, número de DNI o CUI, historia clínica, nombres, apellido paterno, apellido materno, sexo, fecha de nacimiento, edad (meses); también debe ingresar datos antropométricos y de laboratorio como: peso (kg), talla (cm), hemoglobina (gr/dl), fecha de resultado de hemoglobina; asimismo debe ingresar datos de CRED (Control de Crecimiento y Desarrollo de la Niña y el Niño Menor de 5 años) como número de: controles, sachets MM, consejerías nutricionales, sesiones demostrativas, visitas domiciliarias; además el usuario debe registrar si el niño está afiliado a un seguro o más, como: Crecer, SIS, PIN, Juntos; también deberá ingresar datos de lugar de residencia como: región, provincia, distrito, localidad, altitud; seguidamente hace clic en el botón Guardar.
6. El sistema muestra un mensaje preguntando ¿Está seguro que desea grabar?
7. El usuario hace clic en el botón sí.
8. El sistema guarda la información en la base de datos y muestra un mensaje de confirmación indicando que los datos se guardaron correctamente.

ESCENARIO ALTERNO

Datos incompletos: El usuario no ingresó todos los datos requeridos, el sistema muestra un mensaje indicando los datos que falta ingresar.

Formato no válido: El usuario ingresó datos incorrectos, el sistema muestra un mensaje indicando la característica de los datos a ingresar.

Cancelar: El usuario hace clic en el botón Retornar, el sistema cancela el registro y muestra la página de atención del niño.

Error de sistema al registrar datos: El sistema muestra un mensaje de error generado por el sistema.

PRECONDICIONES: El usuario debe haberse autenticado e iniciado sesión previamente y estar en una sesión activa.

POSTCONDICIONES: Se realizó correctamente un registro de la atención de un niño.

4.1.2.4.2. CASO DE USO: Buscar atención niño

Tabla N° 4.11: Revisión del caso de uso buscar atención niño.

CASO DE USO: Buscar atención niño
ACTOR: Usuario
DESCRIPCIÓN: Este caso de uso permite al usuario buscar los datos de la atención de un niño.
ESCENARIO NORMAL
<ol style="list-style-type: none">1. El usuario hace clic en la opción Atención Niños.2. El sistema muestra la página de niños.3. El usuario ingresa el criterio de búsqueda del niño y hace clic en el botón buscar.4. El sistema muestra una tabla de datos con las posibles coincidencias.
ESCENARIO ALTERNO
<p>Formato no válido: El usuario ingresó datos incorrectos, el sistema muestra un mensaje indicando la característica específica de los datos a ingresar.</p> <p>Paciente no encontrado: El usuario ingresa campos de búsqueda no encontrados en la base de datos, el sistema muestra un mensaje indicando que no existen registros.</p> <p>Dato no ingresado para el criterio de búsqueda: El usuario hace clic en el botón Buscar, el sistema muestra un mensaje indicando que por lo menos debe ingresar un dato para el criterio de búsqueda.</p> <p>Error de sistema al modificar datos: El sistema muestra un mensaje de error generado por el sistema.</p>
PRECONDICIONES: El usuario debe haberse autenticado e iniciado sesión previamente y estar en una sesión activa.
POSTCONDICIONES: Se realizó correctamente la búsqueda de un registro de la atención del niño.

4.1.2.4.3. CASO DE USO: Modificar atención niño

Tabla N° 4.12: Revisión del caso de uso modificar atención niño

CASO DE USO: Modificar atención niño
ACTOR: Usuario
DESCRIPCIÓN: Este caso de uso permite al usuario modificar los datos de la atención de un niño.
ESCENARIO NORMAL
<ol style="list-style-type: none">1. El usuario hace clic en la opción Atención Niños.2. El sistema muestra la página de niños con una tabla de datos con los registros de las atenciones de niños.3. El usuario hace clic en el botón modificar.4. El sistema muestra la página de detalle de Niños y habilita las cajas de texto para la edición de los datos de la atención del niño.5. El usuario modifica los datos necesarios de la atención del niño y hace clic en el botón Grabar.6. El sistema muestra un mensaje preguntando ¿Está seguro que desea grabar? El usuario hace clic en el botón sí.7. El sistema guarda la información en la base de datos y muestra un mensaje de confirmación indicando que los datos se guardaron correctamente y muestra la página de atención del niño.
ESCENARIO ALTERNO
Datos incompletos: El usuario no ingresó todos los datos requeridos, el sistema muestra un mensaje indicando los datos que falta ingresar.
Formato no válido: El usuario ingresó datos incorrectos, el sistema muestra un mensaje indicando la característica específica de los datos a ingresar.
Error de sistema al modificar datos: El sistema muestra un mensaje de error generado por el sistema.

PRECONDICIONES: El usuario debe haberse autenticado e iniciado sesión previamente y estar en una sesión activa.

POSTCONDICIONES: Se realizó correctamente la modificación de un registro de la atención del niño.

4.1.2.4.4. CASO DE USO: Emitir reporte del estado nutricional de niños.

Tabla N ° 4.13: Revisión del caso de uso emitir reporte del estado nutricional de niños

CASO DE USO: Emitir reporte del estado nutricional de niños.

ACTOR: Usuario

DESCRIPCIÓN: Este caso de uso permite al usuario emitir reporte del estado nutricional de niños.

ESCENARIO NORMAL

1. El usuario hace clic en la opción Reportes.
2. El sistema muestra la página Reporte del estado nutricional de Niños.
3. El usuario ingresa los criterios de búsqueda requeridos según la necesidad del reporte de niños a procesar, además selecciona el tipo de reporte y el modo de visualización - mapa cuantitativo.
4. El sistema muestra la página de Reporte de Niños con el modo de visualización - mapa cuantitativo del estado nutricional de niños por distritos según provincia y periodo.
5. El usuario hace clic en el botón del formato del archivo donde desea exportar (Excel o PDF).
6. El sistema muestra el reporte del estado nutricional de niños en un archivo plano según el tipo de formato elegido por el usuario.

ESCENARIO ALTERNO

Cancelar reporte de niños: El usuario hace clic en el link Salir, el sistema cierra la página de reporte de niños y muestra la página de niños.

Elegir criterios de búsqueda: El usuario selecciona el tipo de reporte, el sistema muestra un mensaje indicando que antes debe elegir criterio de búsqueda.

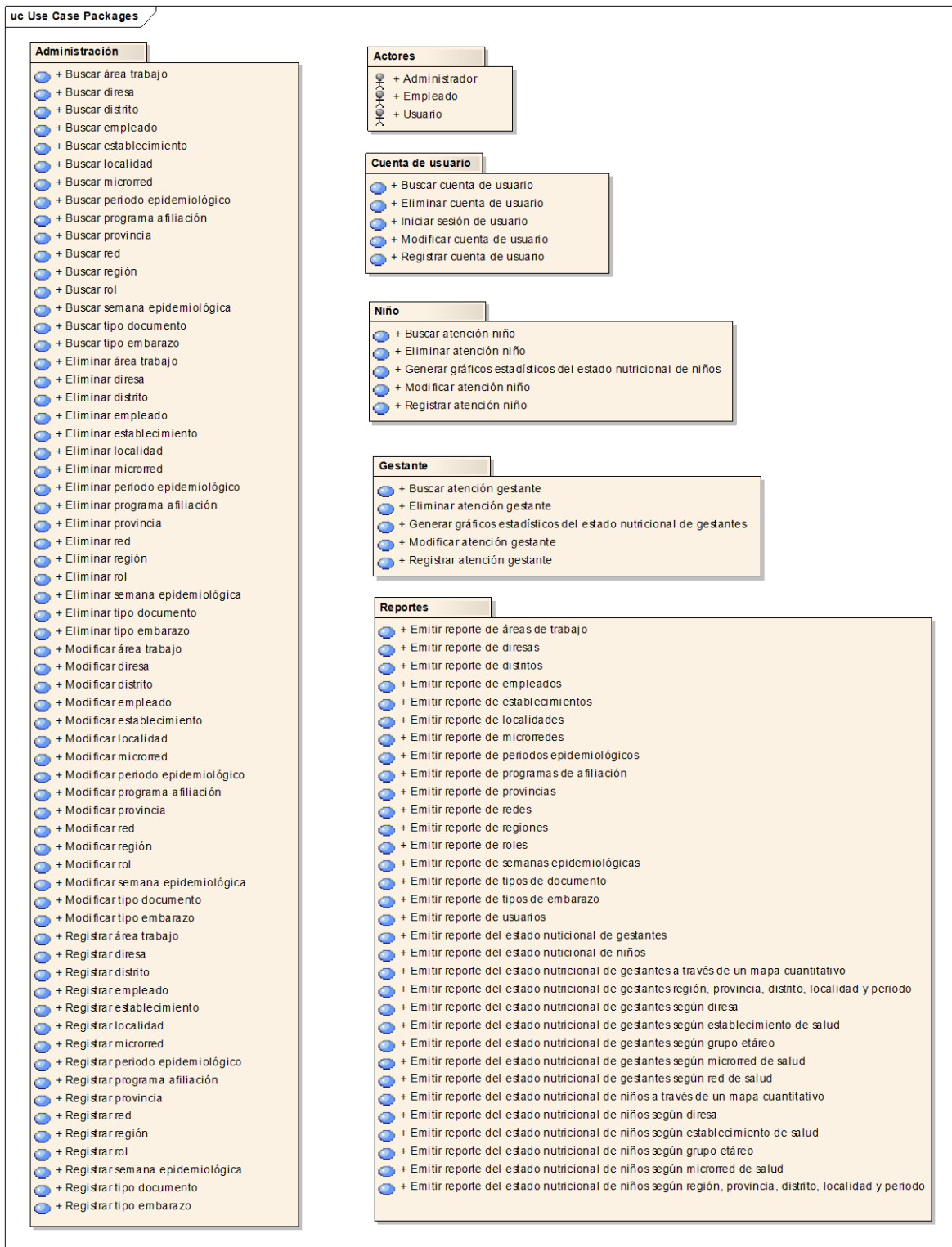
Error de sistema al generar reporte de datos: El sistema muestra un mensaje de error generado.

PRECONDICIONES: El usuario debe haberse autenticado e iniciado sesión previamente y estar en una sesión activa.

POSTCONDICIONES: Se generó correctamente el reporte del estado nutricional de niños.

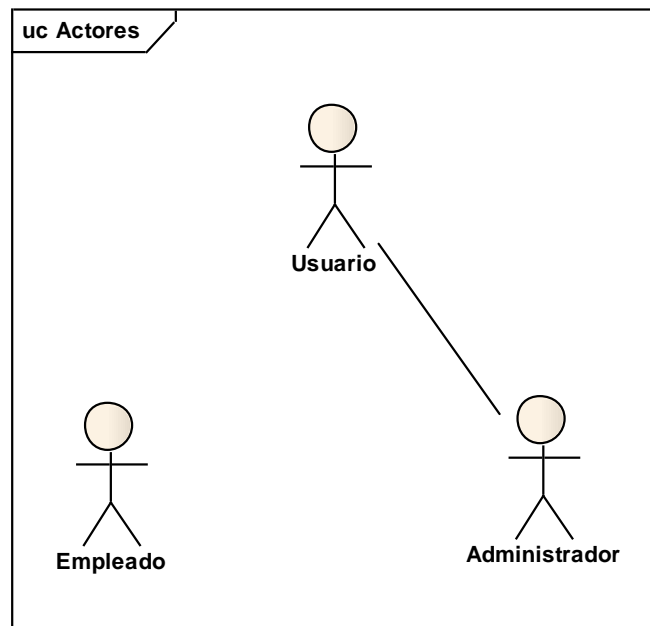
4.1.2.5. CASO DE USO POR PAQUETES

Figura N° 4.6 Empaquetado general de casos de uso.



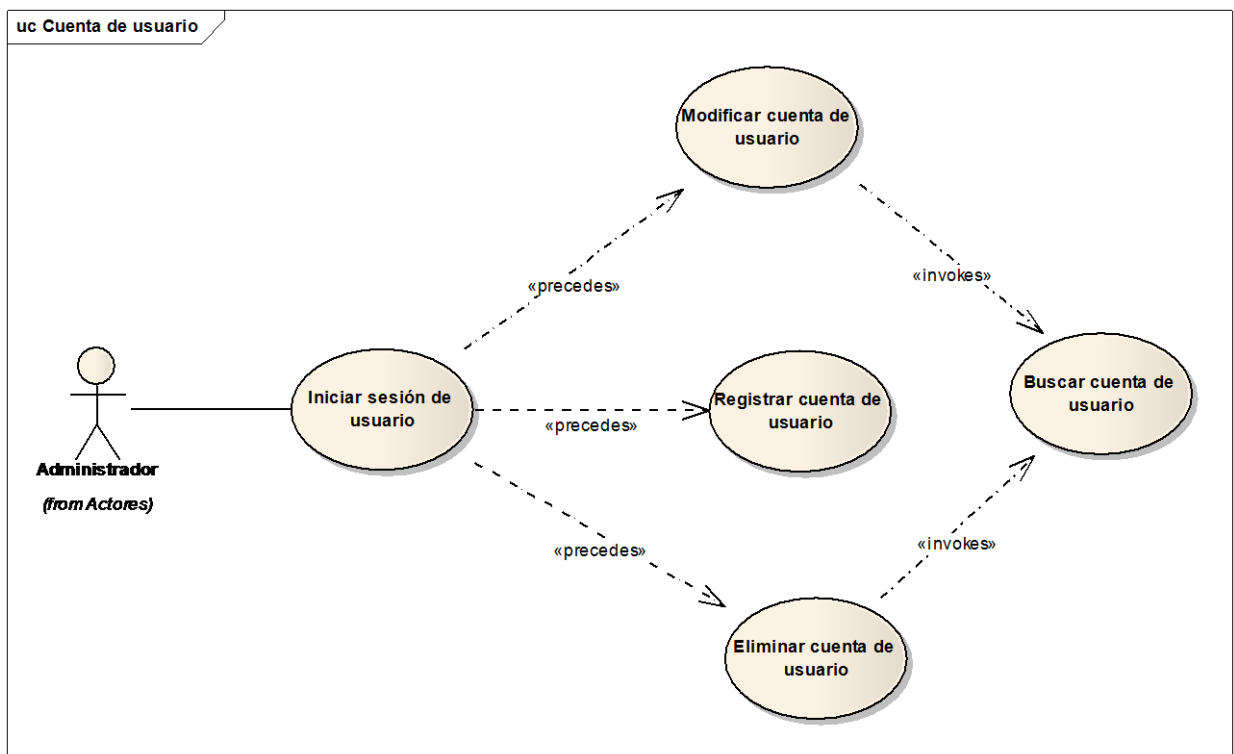
4.1.2.5.1. PAQUETE “Actores”

Figura N° 4.7: Paquete actores de sistemas.



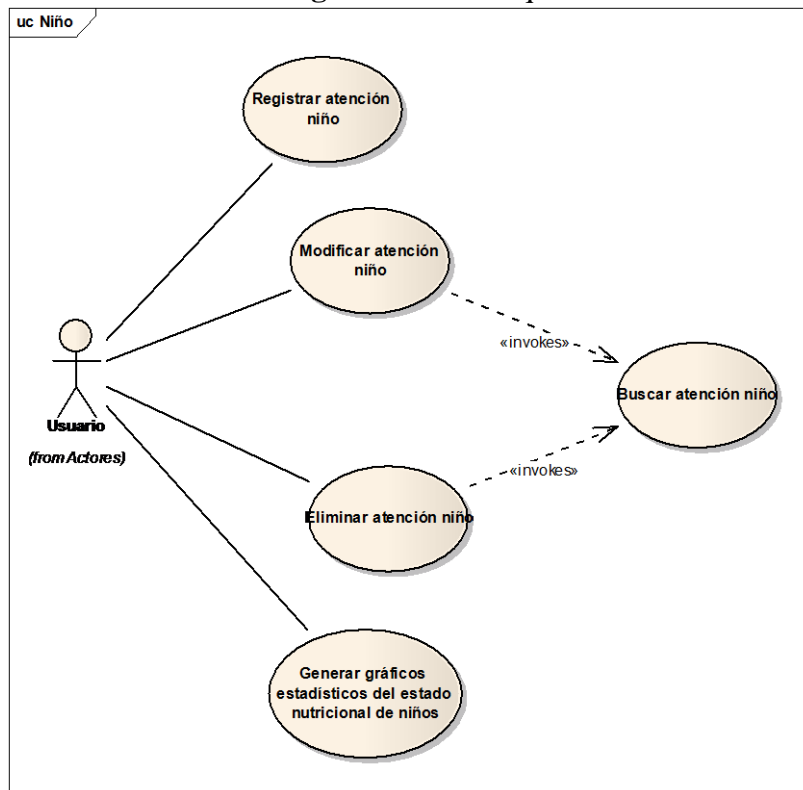
4.1.2.5.2. PAQUETE “Cuenta de usuario”

Figura N° 4.8: Paquete cuentas de usuario



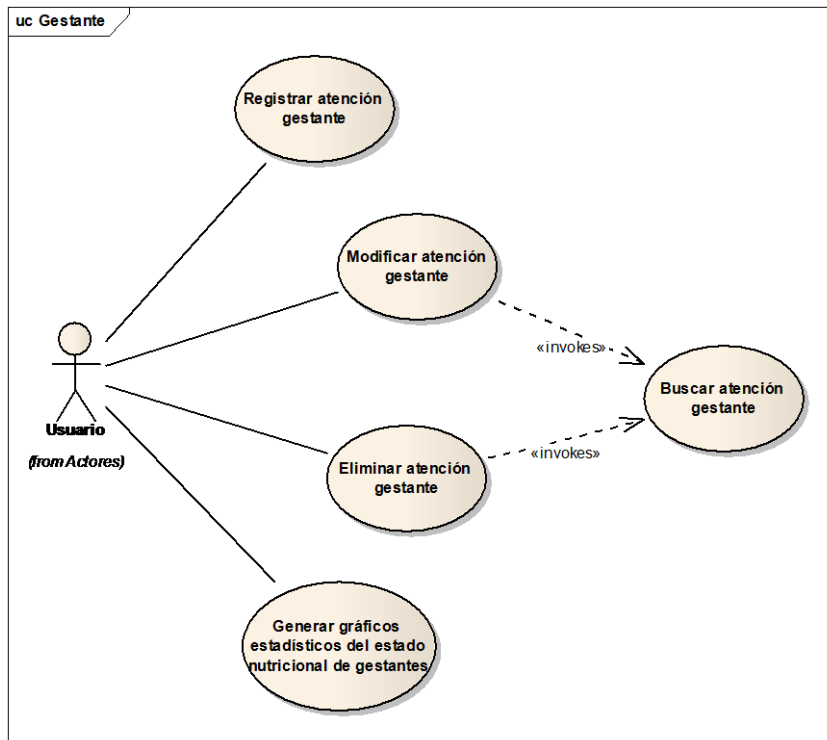
4.1.2.5.3. PAQUETE “Niño”

Figura N° 4.9: Paquete niño



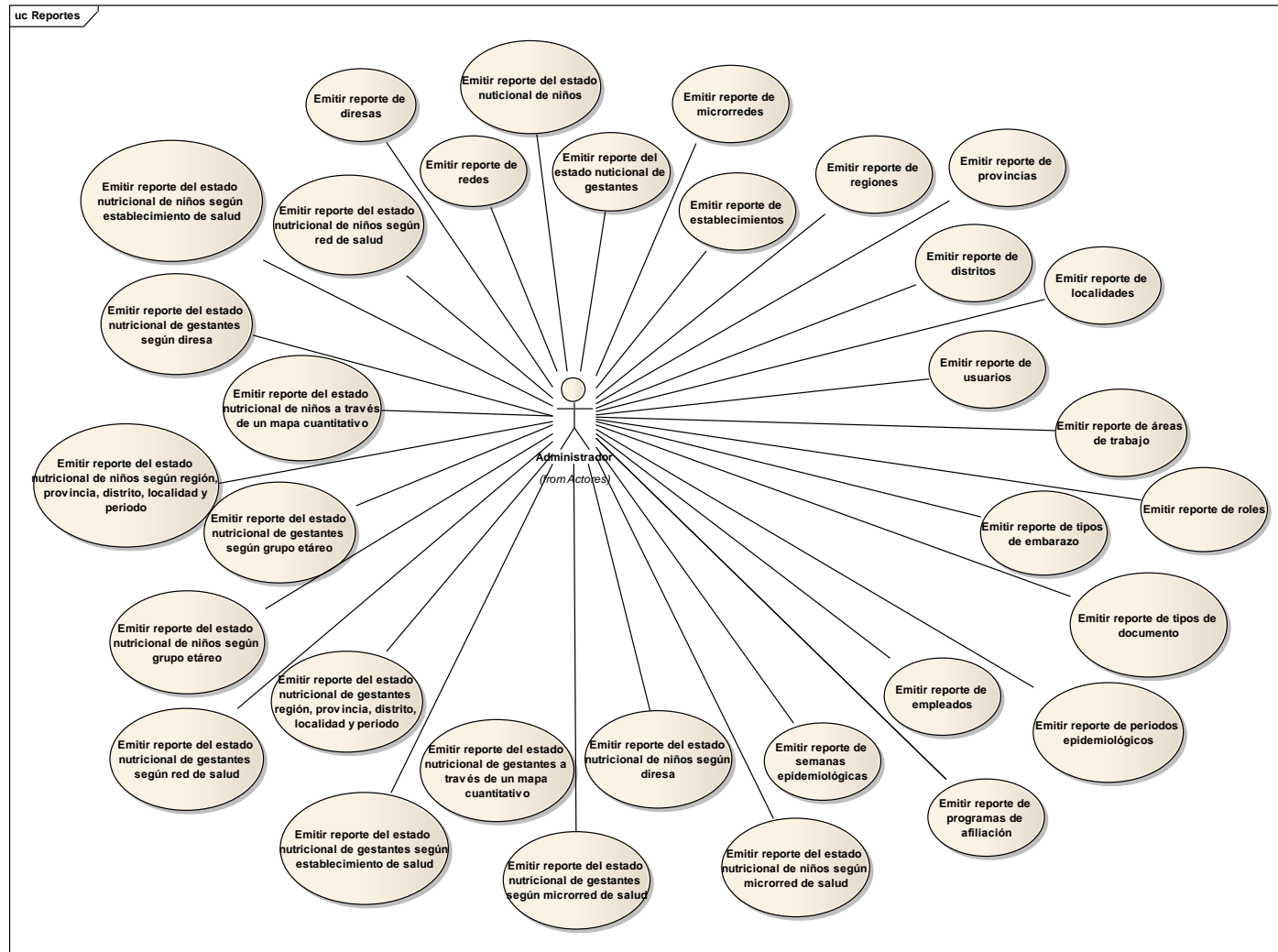
4.1.2.5.4. PAQUETE “Gestante”

Figura N° 4.10: Paquete gestante



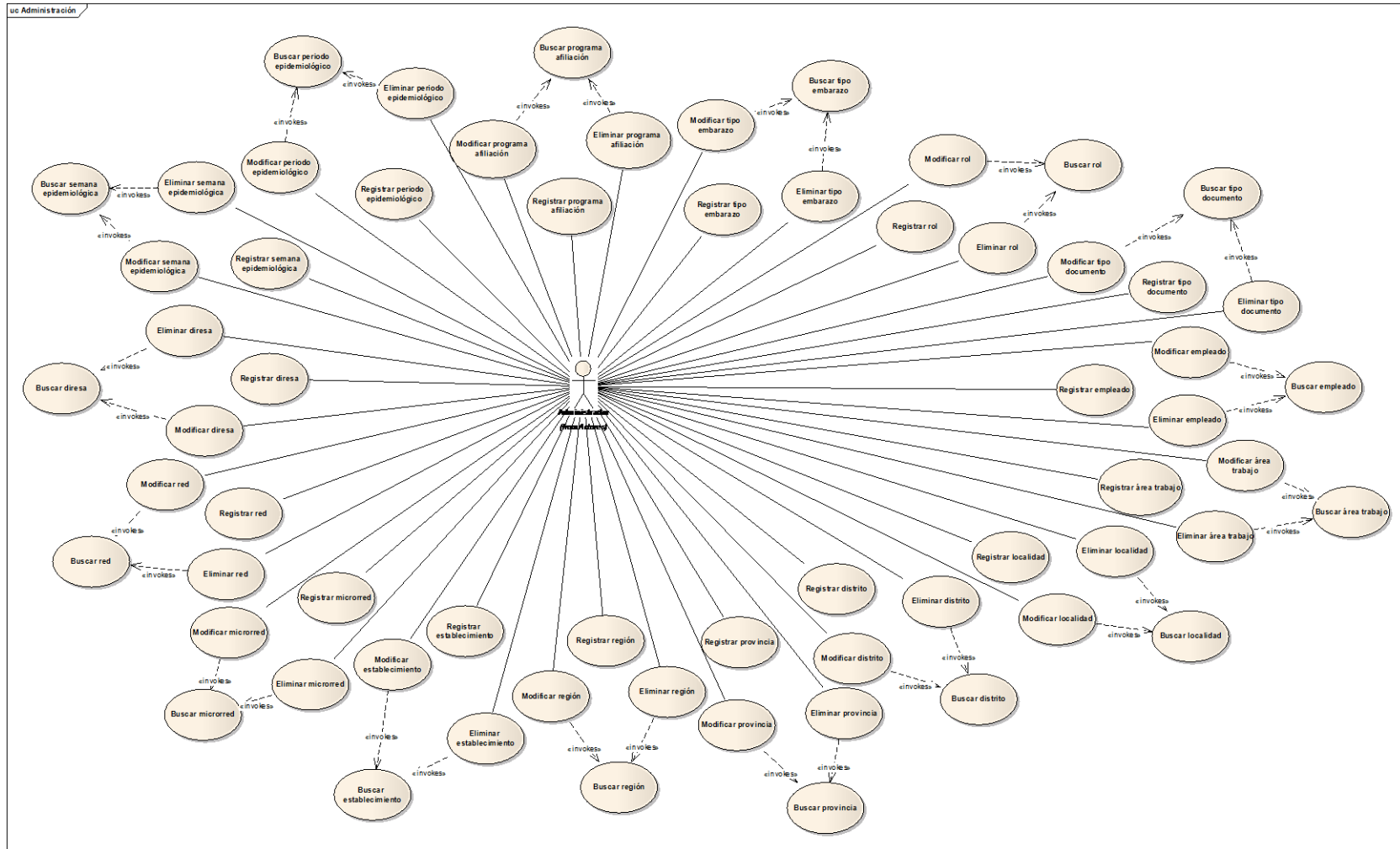
4.1.2.5.5. PAQUETE “Reportes”

Figura N° 4.11: Paquete reportes



4.1.2.5.6. PAQUETE “Administración”

Figura N° 4.12: Paquete administración



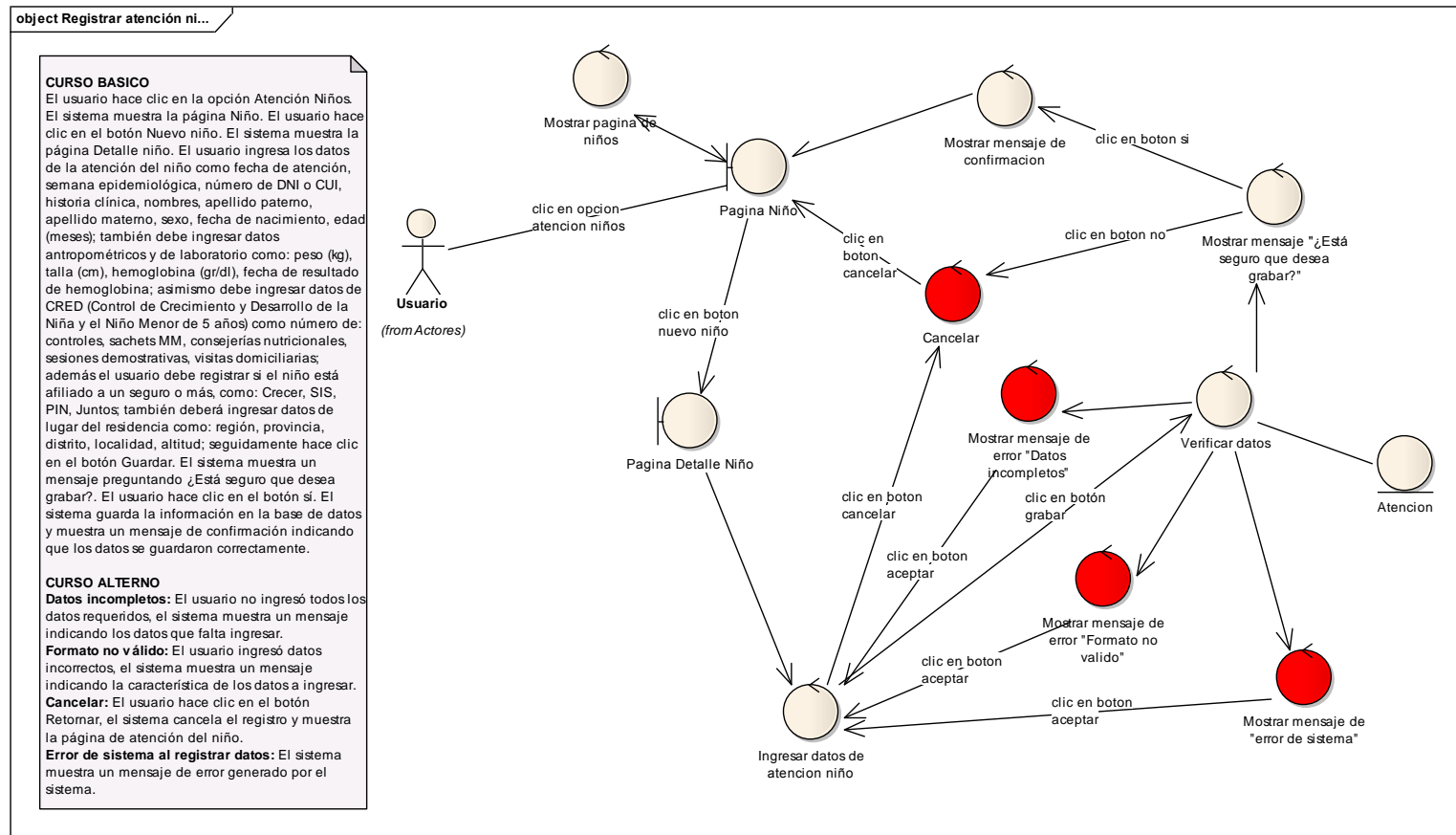
4.1.3. ETAPA 2: ANÁLISIS Y DISEÑO PRELIMINAR

4.1.3.1. DIAGRAMA DE ROBUSTEZ

Se elaboran diagramas de robustez para los casos de usos seleccionados:

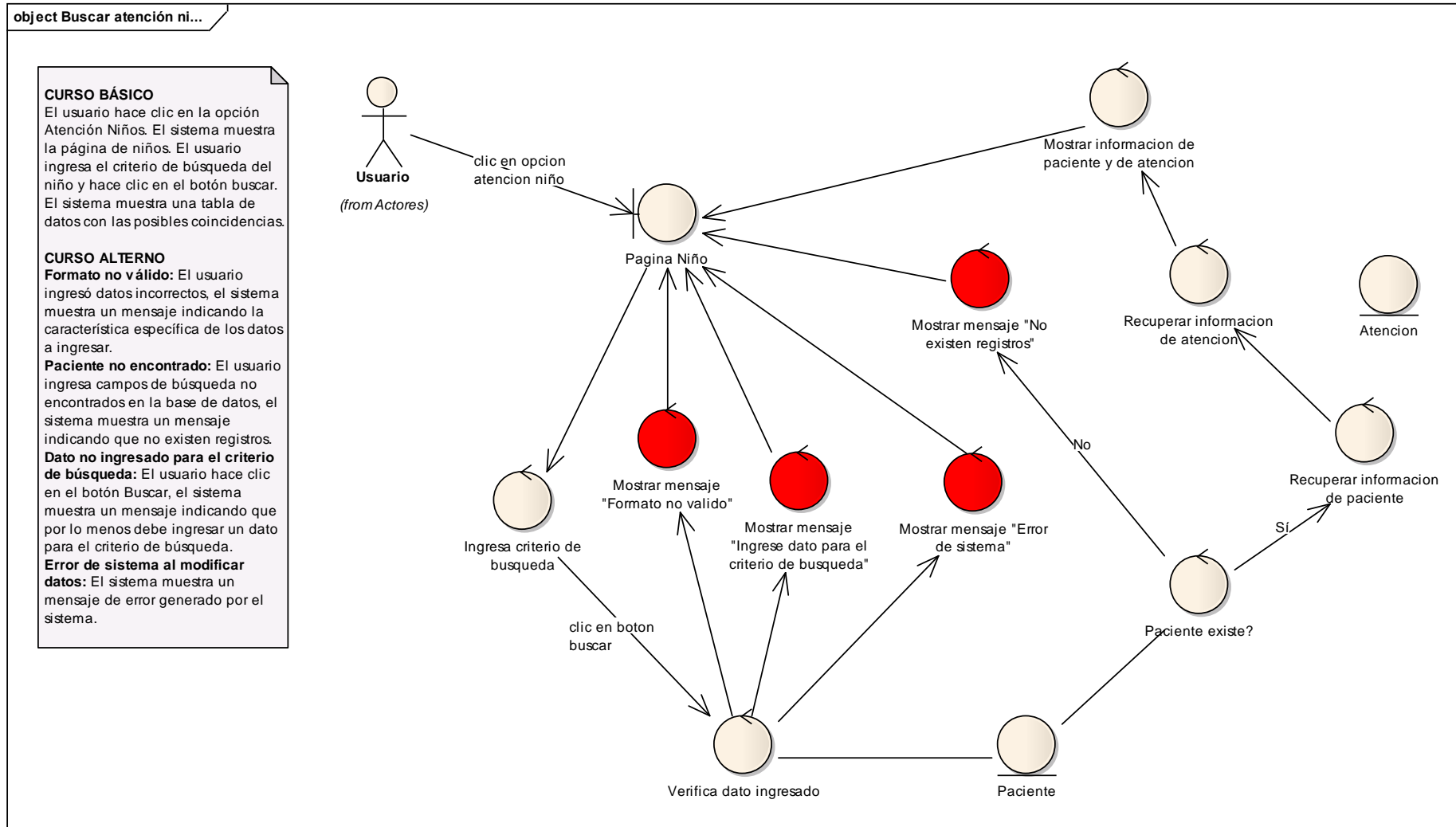
4.1.3.1.1. DIAGRAMA DE ROBUSTEZ: Registrar atención niño

Figura N° 4.13: Diagrama de robustez: registrar atención niño



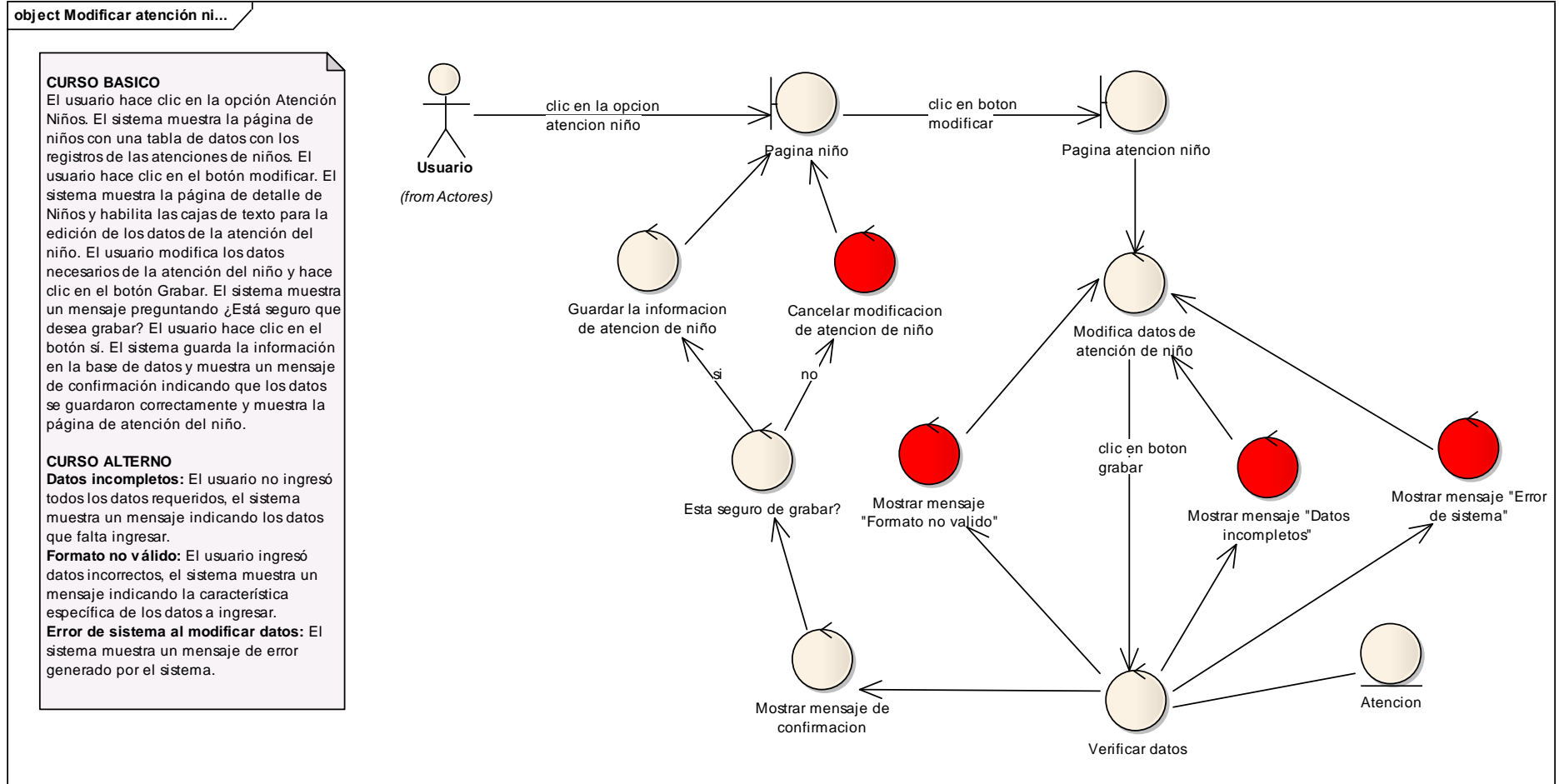
4.1.3.1.2. DIAGRAMA DE ROBUSTEZ: Buscar atención niño

Figura N° 4.14: Diagrama de robustez: buscar atención niño



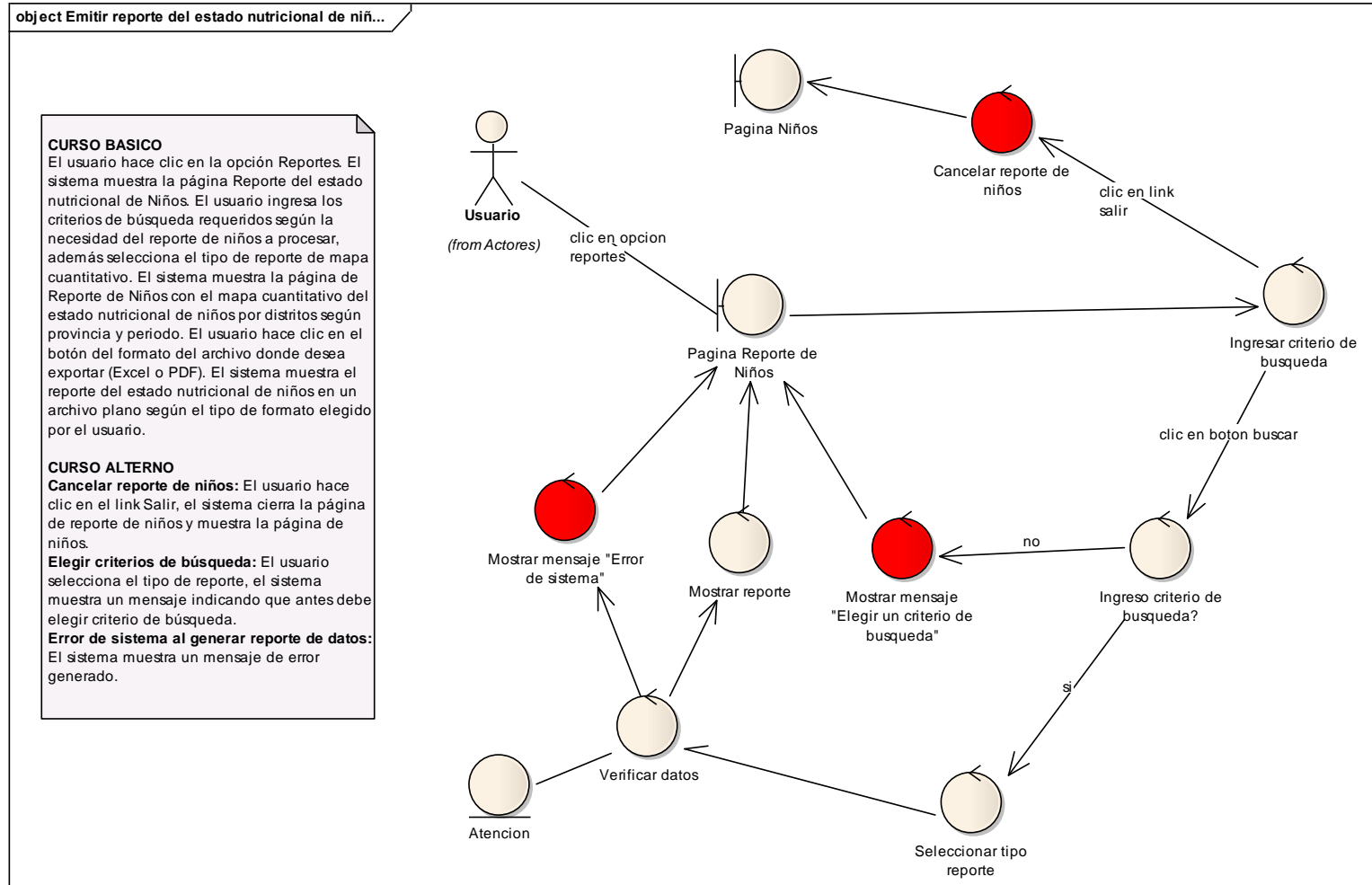
4.1.3.1.3. DIAGRAMA DE ROBUSTEZ: Modificar atención niño

Figura N° 4.15: Diagrama de robustez: modificar atención niño



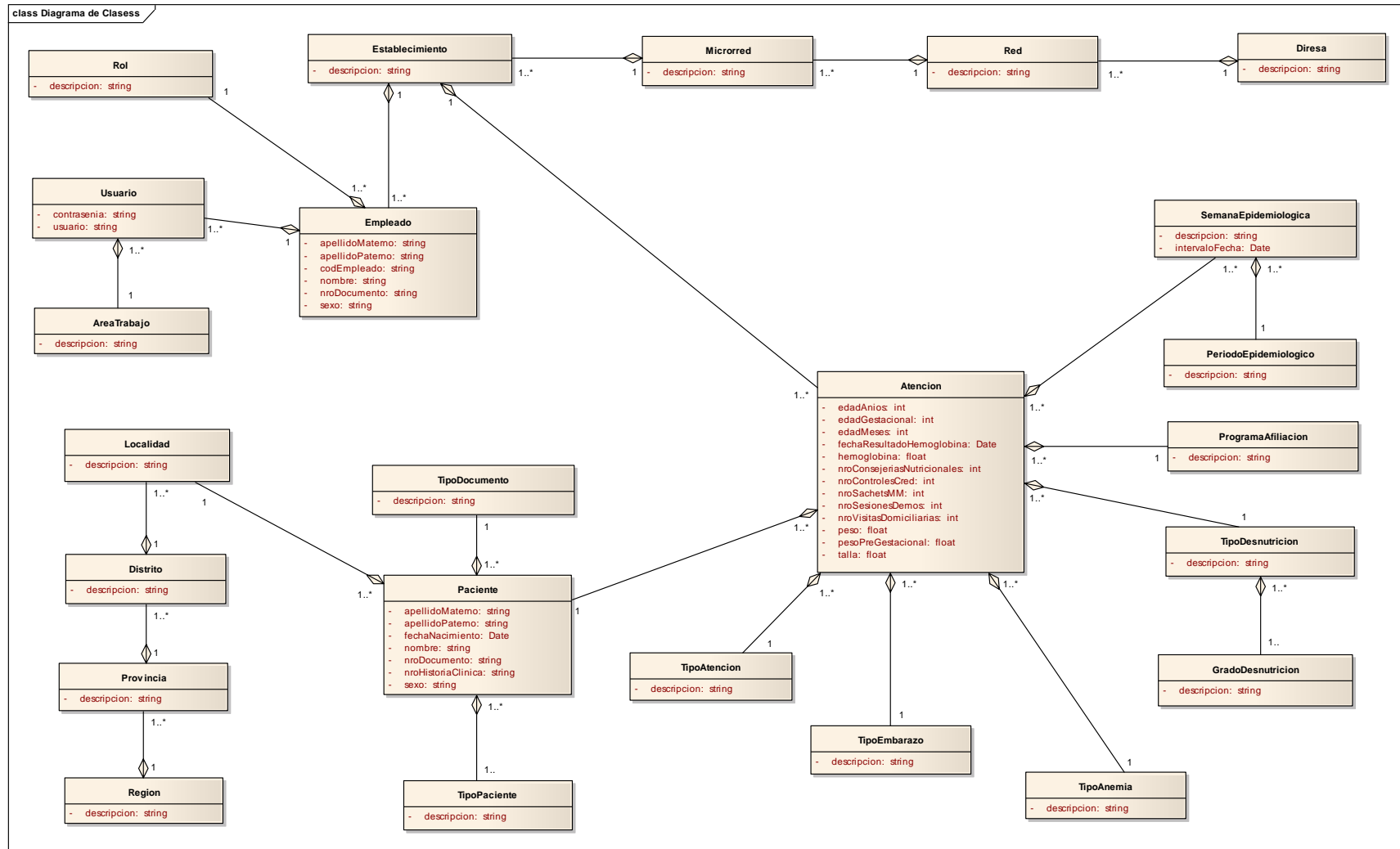
4.1.3.1.3. DIAGRAMA DE ROBUSTEZ: Emitir reporte del estado nutricional de niños

Figura N° 4.16: Diagrama de robustez: emitir reporte del estado nutricional de niños



4.1.3.2. MODELO DE DOMINIO ACTUALIZADO

Figura N° 4.17: Modelo de dominio actualizado para los casos de uso seleccionados



4.1.4. HITO ETAPA 2: REVISIÓN DEL DISEÑO PRELIMINAR (PDR)

No se encontraron cambios en la descripción de casos de uso, ni en los requisitos revisados.

4.1.5. DISEÑO DETALLADO

4.1.5.1. ARQUITECTURA TÉCNICA DEL FRAMEWORK .NET

Según Antonio Martín Sierra (2007), “El .NET Framework constituye el núcleo fundamental de .NET al proporcionar la plataforma para el desarrollo y ejecución de las aplicaciones” (p. 5)

El Framework de .Net es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables. Los principales componentes de este entorno son:

- Lenguajes de compilación (Esamblados).
- Biblioteca de clases de .Net
- CLR (CommonLanguageRuntime).

Figura N° 4.18: Arquitectura de .NET Framework



A. LENGUAJES DE COMPILACIÓN

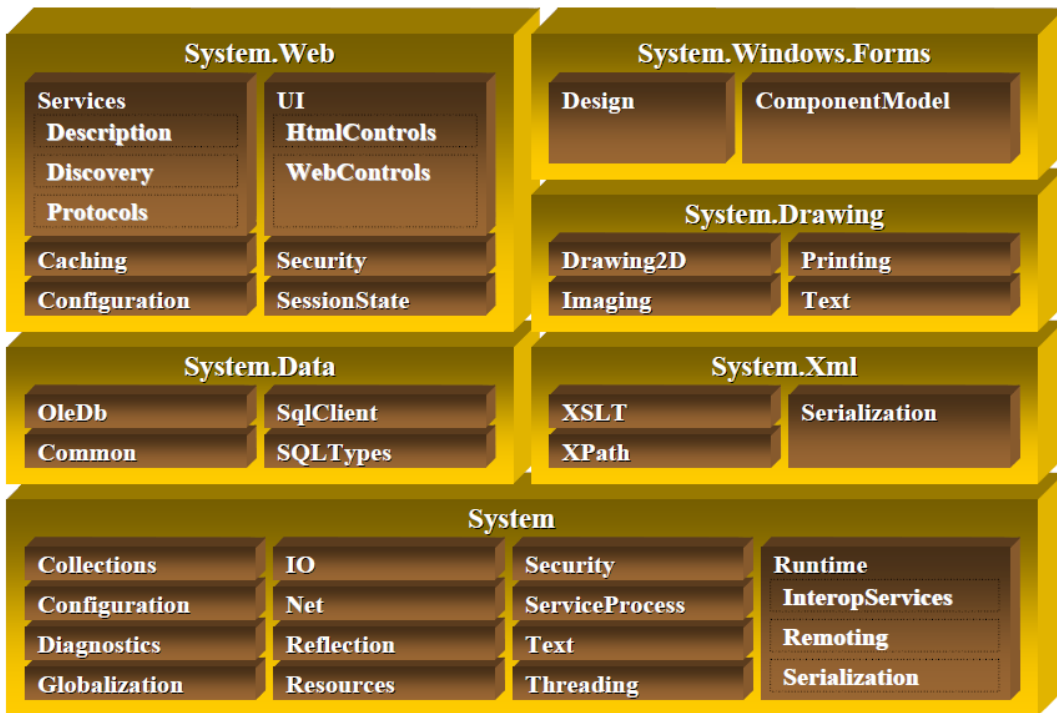
Uno de los mayores problemas de las aplicaciones actuales es que en muchos casos tienen que tratar con diferentes archivos binarios (DLLs), elementos de registro, conectividad abierta a bases de datos (ODBC), etc. Para solucionarlo el Framework de .Net maneja un nuevo concepto denominado ensamblado. Los ensamblados son ficheros con forma de EXE o DLL que contienen toda la funcionalidad de la aplicación de forma encapsulada. Por tanto la solución al problema puede ser tan fácil como copiar todos los ensamblados en el directorio de la aplicación. Con los ensamblados ya no es necesario registrar los componentes de la aplicación. Esto se debe a que los ensamblados almacenan dentro de sí mismos toda la información necesaria en lo que se denomina el manifiesto del ensamblado. El manifiesto recoge todos los métodos y propiedades en forma de meta-datos junto con otra información descriptiva, como permisos, dependencias. Para gestionar el uso que hacen las aplicaciones de los ensamblados .Net utiliza la llamada caché global de ensamblados (GAC, Global Assembly Cache). Así, .Net Framework puede albergar en el GAC los ensamblados que puedan ser usados por varias aplicaciones e incluso distintas versiones de un mismo ensamblado, algo que no era posible con el anterior modelo COM.

B. BIBLIOTECA DE CLASES DE .NET

Cuando se está programando una aplicación muchas veces se necesitan realizar acciones como manipulación de archivos, acceso a datos, conocer el estado del sistema, implementar seguridad, etc. El Framework organiza toda la funcionalidad del sistema operativo en un espacio de nombres jerárquico de forma que a la hora de programar resulta bastante sencillo encontrar lo que se necesita. Para ello, el Framework posee un sistema de tipos universal, denominado CommonTypeSystem (CTS). Este sistema permite que el programador pueda interactuar los tipos que se incluyen en el propio Framework (biblioteca de clases de .Net) con los creados por él mismo (clases). De esta forma se aprovechan las ventajas propias de la programación orientada a objetos, como la herencia de clases predefinidas para crear nuevas clases, o el polimorfismo de clases para modificar o ampliar funcionalidades de clases ya existentes. La biblioteca de clases de .Net Framework incluye, entre otros, tres componentes clave:

- ASP.NET para construir aplicaciones y servicios Web.
- Windows Forms para desarrollar interfaces de usuario.
- ADO.NET para conectar las aplicaciones a bases de datos.

Figura N° 4.19: Biblioteca de clases .Net Framework



La forma de organizar la biblioteca de clases de .Net dentro del código es a través de los espacios de nombres (namespaces), donde cada clase está organizada en espacios de nombres según su funcionalidad. Por ejemplo, para manejar ficheros se utiliza el espacio de nombres System.IO y si lo que se quiere es obtener información de una fuente de datos se utilizará el espacio de nombres System.Data.

La principal ventaja de los espacios de nombres de .Net es que de esta forma se tiene toda la biblioteca de clases de .Net centralizada bajo el mismo espacio de nombres (System). Además, desde cualquier lenguaje se usa la misma sintaxis de invocación, ya que a todos los lenguajes se aplica la misma biblioteca de clases.

C. ENSAMBLADOS

Uno de los mayores problemas de las aplicaciones actuales es que en muchos casos tienen que tratar con diferentes archivos binarios (DLLs), elementos de registro, conectividad abierta a bases de datos (ODBC). Para solucionarlo el Framework de .Net maneja un nuevo concepto denominado ensamblado. Los ensamblados son ficheros con forma de EXE o DLL que contienen toda la funcionalidad de la aplicación de forma encapsulada. Por tanto la solución al problema puede ser tan fácil como copiar todos los

ensamblados en el directorio de la aplicación.

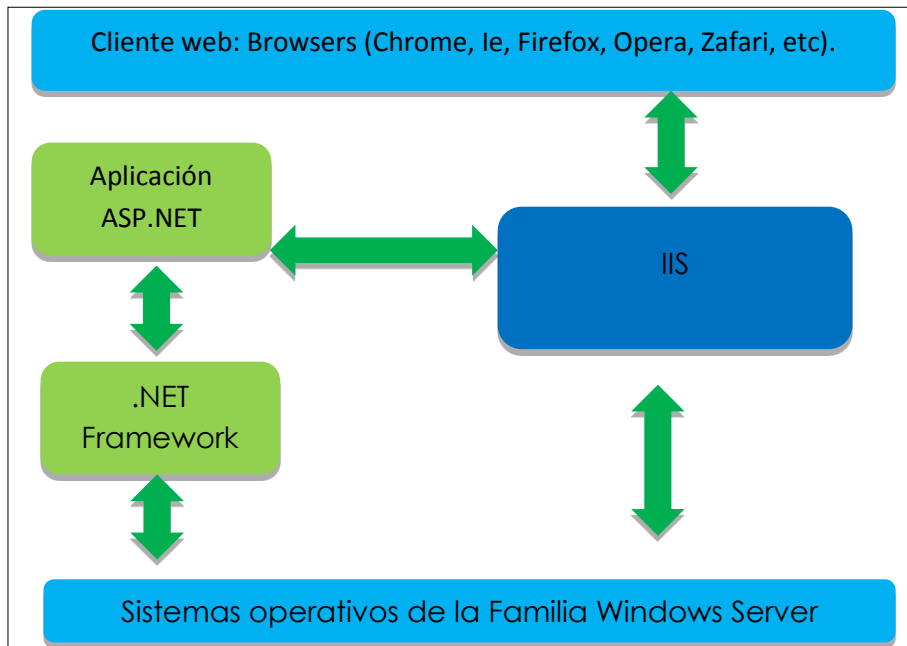
Con los ensamblados ya no es necesario registrar los componentes de la aplicación. Esto se debe a que los ensamblados almacenan dentro de sí mismos toda la información necesaria en lo que se denomina el manifiesto del ensamblado. El manifiesto recoge todos los métodos y propiedades en forma de meta-datos junto con otra información descriptiva, como permisos, dependencias. Para gestionar el uso que hacen las aplicaciones de los ensamblados .Net utiliza la llamada caché global de ensamblados (GAC, Global Assembly Cache). Así, .Net Framework puede albergar en el GAC los ensamblados que puedan ser usados por varias aplicaciones e incluso distintas versiones de un mismo ensamblado, algo que no era posible con el anterior modelo COM.

4.1.5.2. ARQUITECTURA DE LA APLICACIÓN ASP.NET

Como se muestra en la ilustración, todos los clientes Web se comunican con las aplicaciones ASP.NET a través de los Servicios de Internet Information Server (IIS). IIS descifra y, opcionalmente, realiza la autenticación de la petición. Si la opción Permitir anónimos está establecida en true, no se efectuará ninguna autenticación. IIS también busca el recurso solicitado (como una aplicación ASP.NET) y, si se autoriza al cliente, devuelve el recurso correspondiente.

Además de las características integradas de ASP.NET, una aplicación ASP.NET puede utilizar características de seguridad de bajo nivel de .NET Framework. Para obtener más información, vea el tema "Conceptos clave de seguridad" en la Ayuda de .NET Framework.

Figura N° 4.20: Arquitectura de ASP.NET



4.1.5.3. ARQUITECTURA TÉCNICA POR CAPAS DE LA APLICACIÓN WEB

Para el desarrollo de los casos de uso seleccionados se elige el patrón MVC, de tres capas, para las vistas se usaran los aspx v3.0, para los controladores y las clases se construirán en el lenguaje orientado a objetos C# v3.5, por su potencia y robustez y para la base de datos se construirá en MSSQL SERVER 2008 sp3, para la conectividad de la aplicación con el motor de base de datos se hará uso de ADO.NET mediante los DataSet y sus correspondientes DataTable.

Figura N° 4.21: Arquitectura técnica. Diagrama de componentes

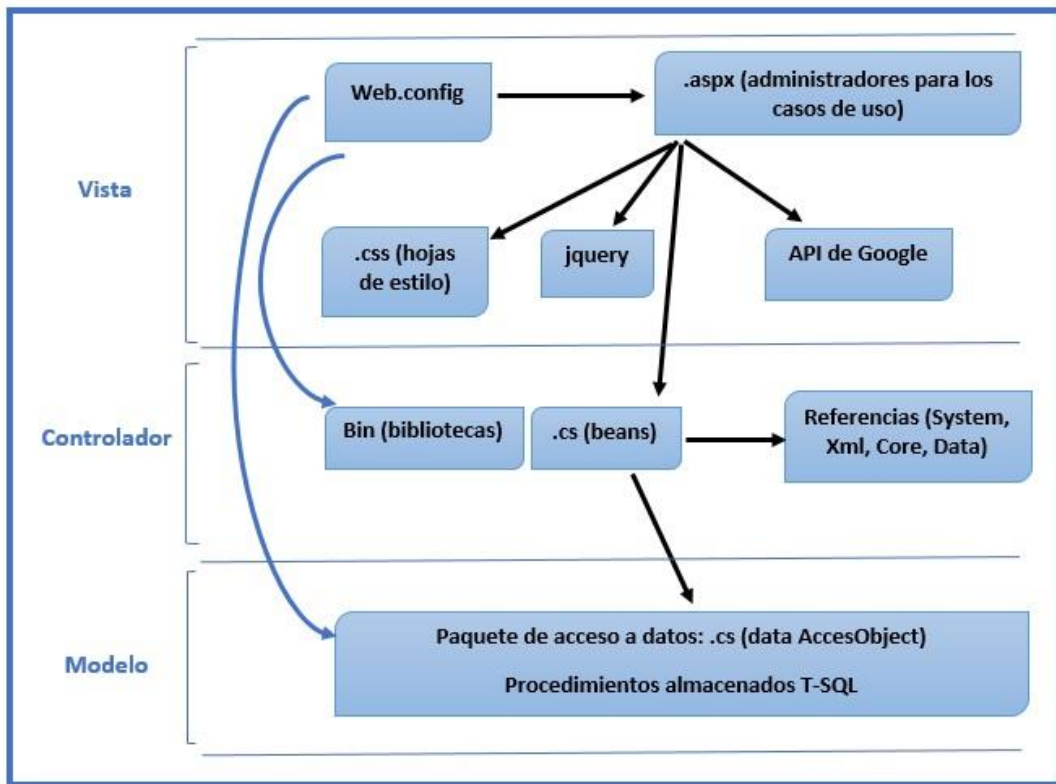
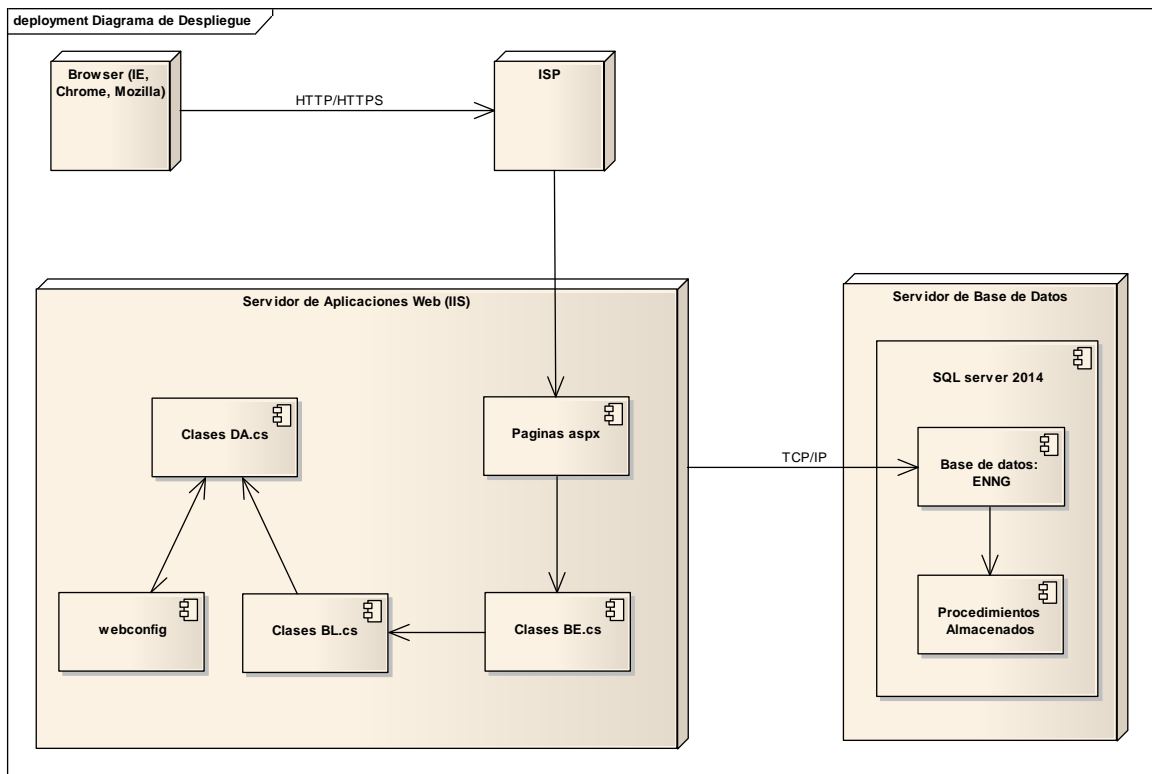
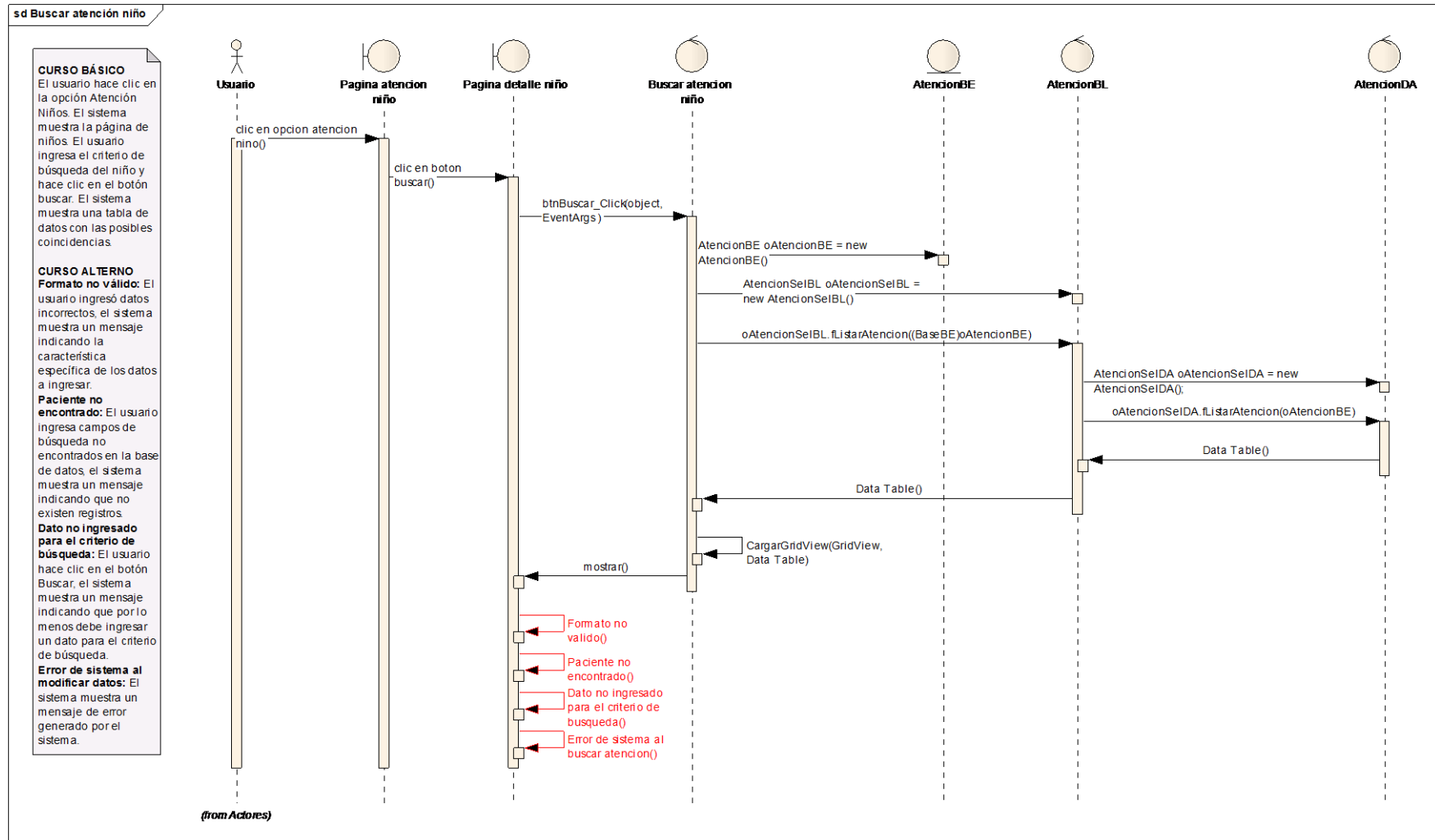


Figura N° 4.22: Arquitectura técnica. Diagrama de despliegue



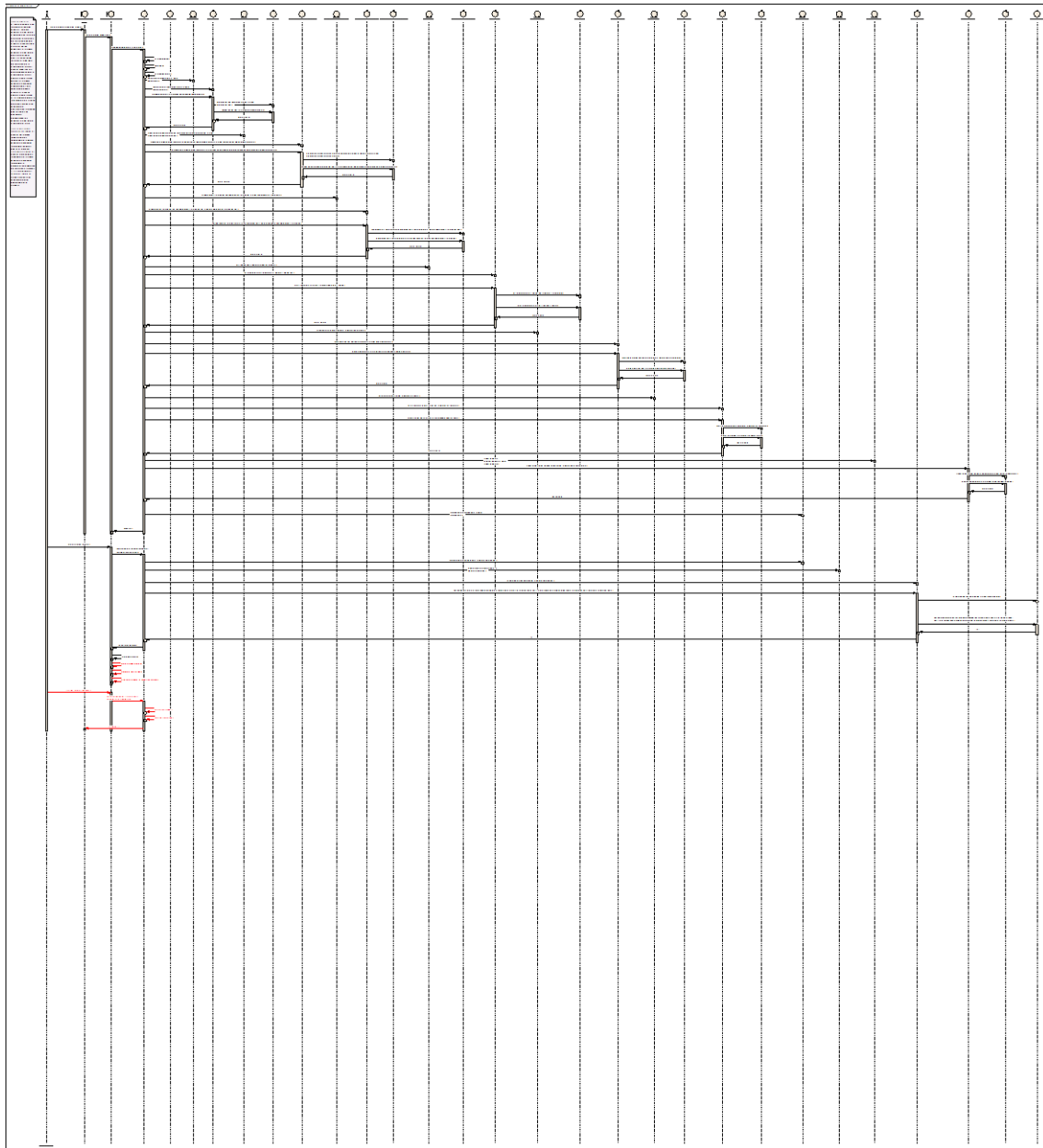
4.1.6.2. DIAGRAMA DE SECUENCIA: Buscar atención niño

Figura N° 4.24: Diagrama de secuencia: Buscar atención niño



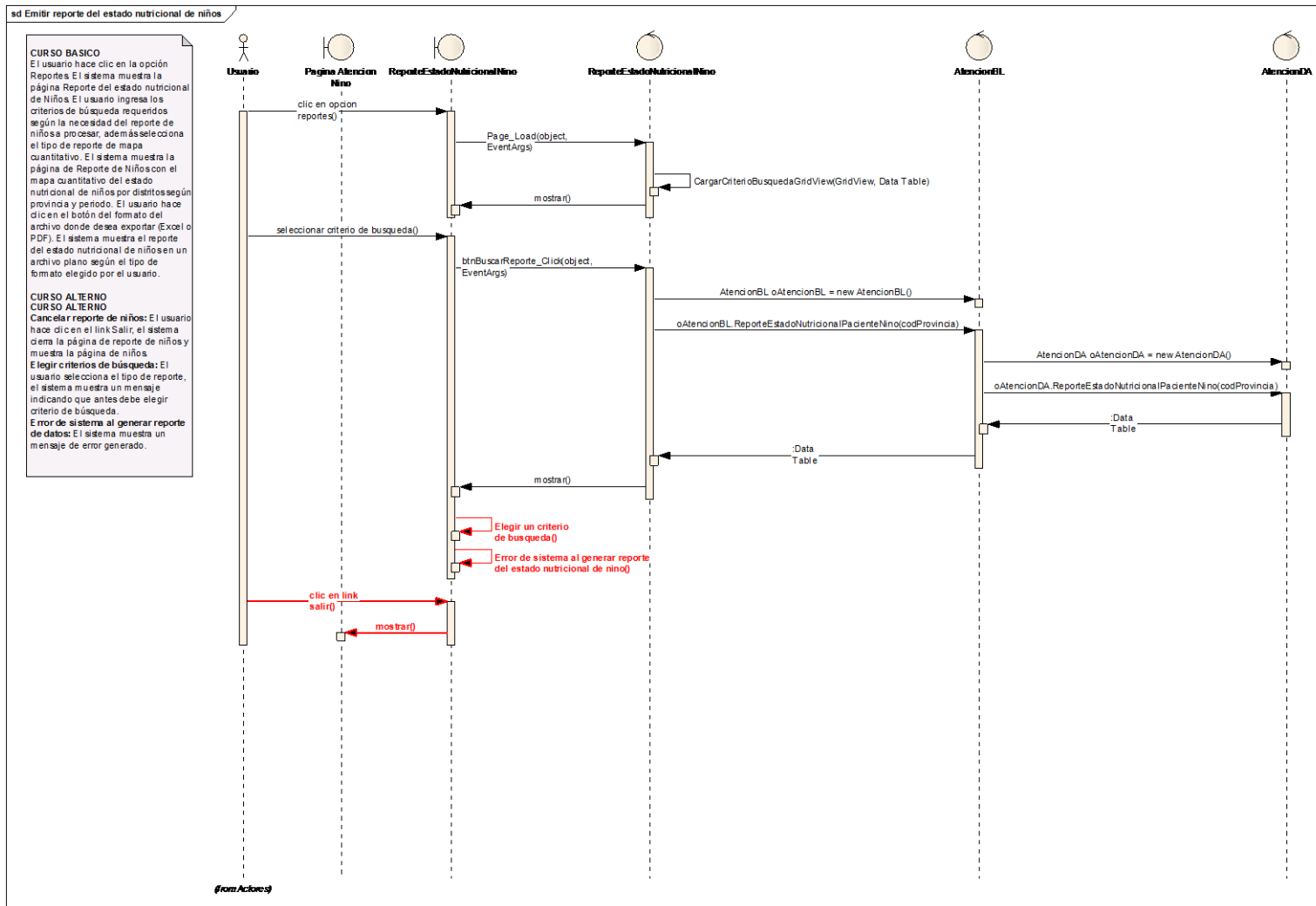
4.1.6.3. DIAGRAMA DE SECUENCIA: Modificar atención niño

Figura N° 4.25: Diagrama de secuencia: Modificar atención niño



4.1.6.4. DIAGRAMA DE SECUENCIA: Emitir reporte del estado nutricional de niños

Figura N° 4.26: Diagrama de secuencia: Emitir reporte del estado nutricional de niños

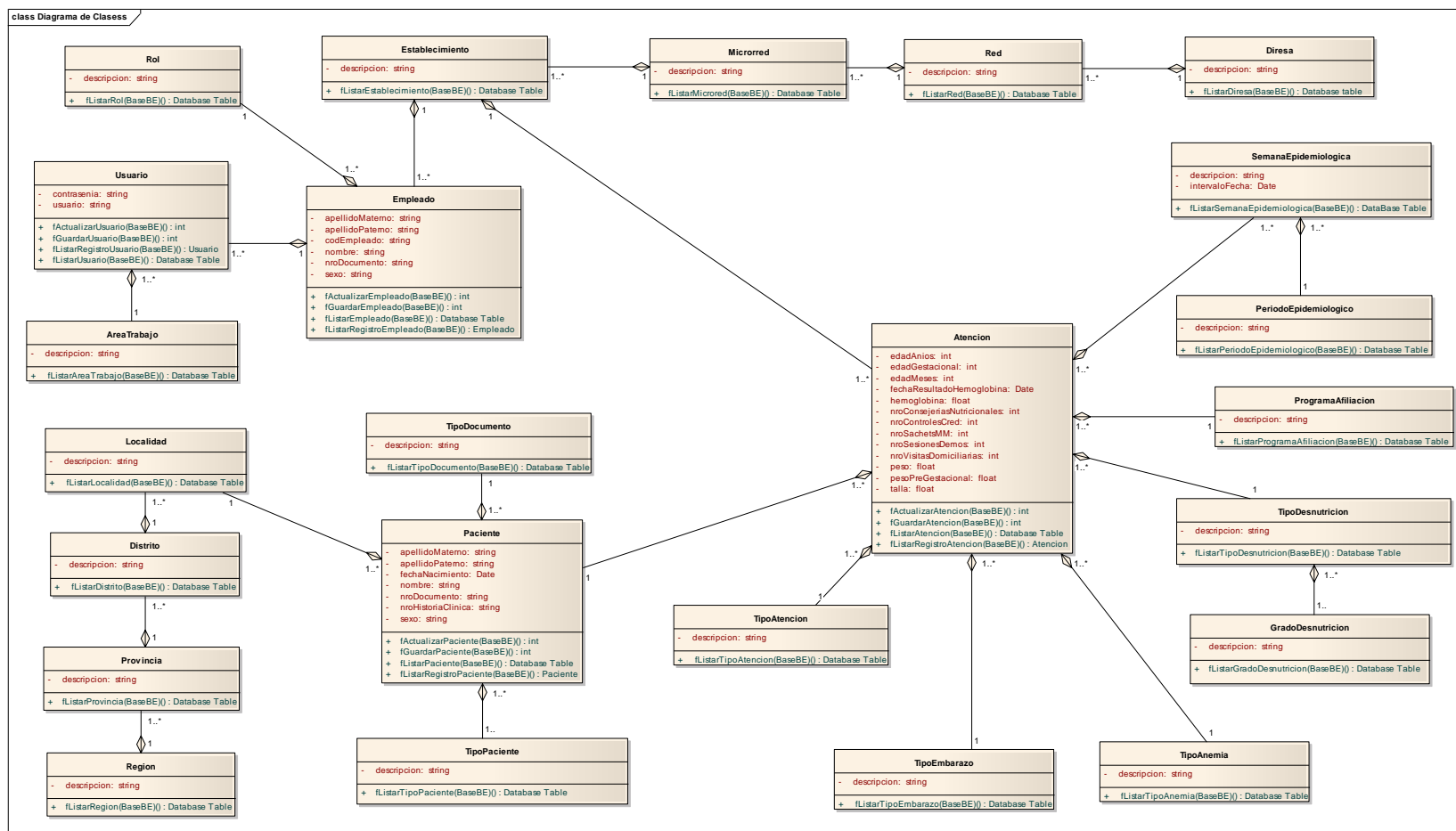


4.1.7. HITO ETAPA 3: REVISIÓN DEL DISEÑO DETALLADO

No se encontraron cambios en los diagramas de secuencia, y se pasa a diseñar el diagrama de clases.

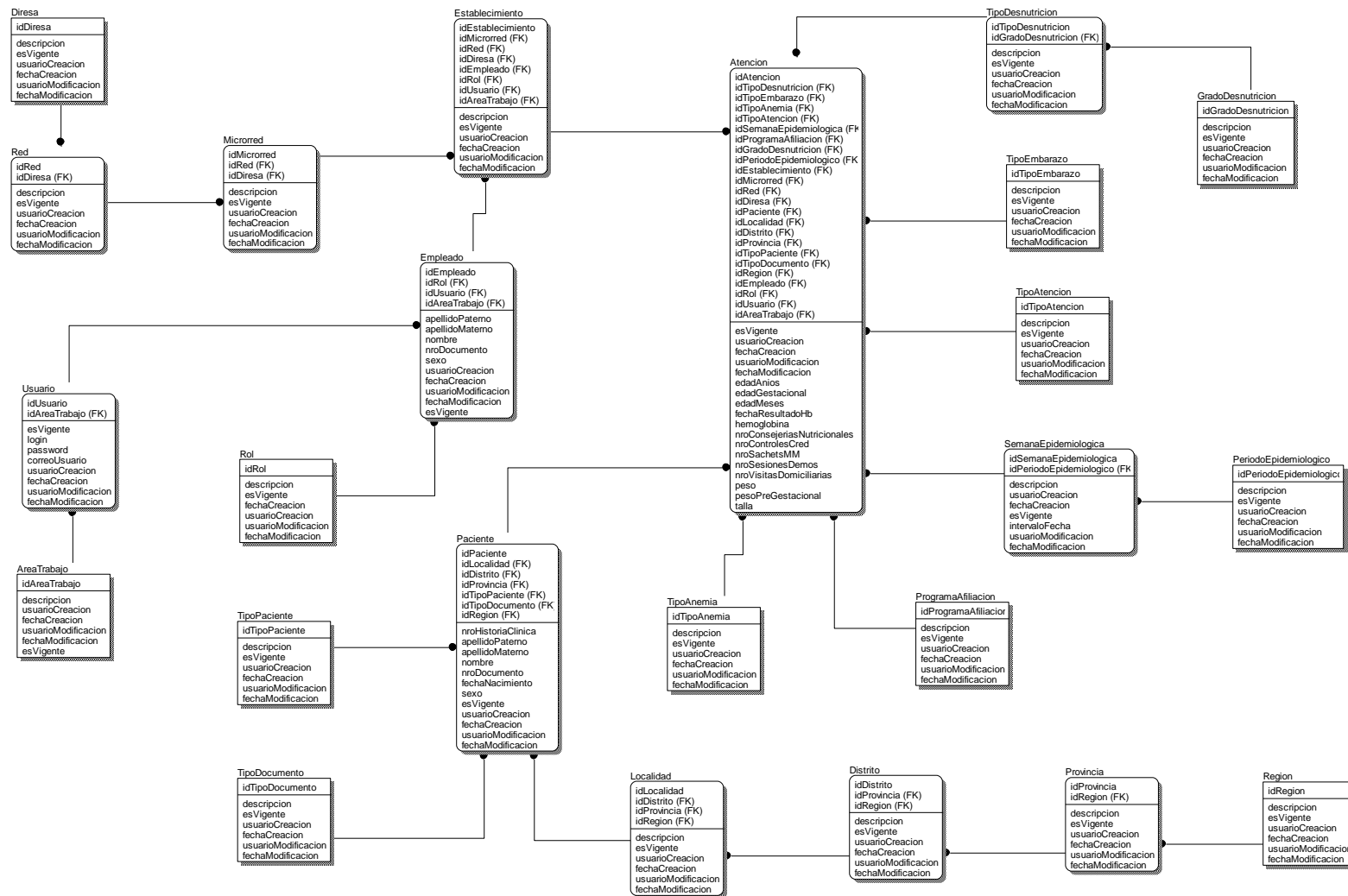
4.1.7.1. DIAGRAMA DE CLASES DE DISEÑO

Figura N° 4.27: Diagrama de clases para los casos de uso seleccionados



4.1.7.2. DIAGRAMA DEL MODELO LÓGICO DE LA BASE DE DATOS GENERAL

Figura N° 4.28: Diagrama Lógico de la Base de Datos para los casos de uso seleccionados



4.1.8. IMPLEMENTACIÓN

Para la implementación de los casos de uso se utilizó Visual Studio 2015 como el IDE de desarrollo, para las vistas se usó ASP.Net 4.0, implementados en el lenguaje de programación C#.

El Framework de desarrollo fue implementado con el Net Framework v4.0, para la implementación de la base de datos se utilizó el sistema modelador de datos Erwin Data Modeler r.7.3 y por el gestor de base de datos Microsoft SQL Server 2014, el modelado de software fue desarrollado con el software de modelador de sistemas Enterprise Architect v7.3 utilizando como herramienta de modelado del método Iconix, para las pruebas unitarias Visual Studio 2015 implementa una opción para poder generar estas pruebas denominada “Unit Test”, tomadas en cuenta por su versatilidad en las aplicaciones web.

Figura N° 4.29: Sobrecarga del constructor de la clase Atencion

```
public AtencionResponse GetAtencion(int idAtencion)
{
    var response = (
    from a in _contextConfiguracion.Query<CONTROL.atencion>()
    where a.ID_ATENCION == idAtencion
    select new AtencionResponse
    {
        IdAtencion = a.ID_ATENCION,
        FechaAtencion = a.FECHA_ATENCION,
        SemanaEpidemiologica = a.semana_epidemiologica.NRO_SEMANA,
        IdTipoDocumento = a.paciente.ID_TIPO_DOCUMENTO,
        NumeroDocumento = a.paciente.NUMERO_DOCUMENTO,
        Nombres = a.paciente.NOMBRES,
        PrimerApellido = a.paciente.PRIMER_APELLIDO,
        SegundoApellido = a.paciente.SEGUNDO_APELLIDO,
        HistoriaClinica = a.paciente.HISTORIA_CLINICA,
        Sexo = a.paciente.SEXO,
        FechaNacimiento = a.paciente.FECHA_NACIMIENTO,
        EdadMeses = a.EDAD_MESES,

        EdadAnios = a.EDAD_ANIOS,
        TallaCm = a.TALLA_CM,
        HemoglobinaGrDl = a.HEMOGLOBINA_GR_DL,
        EdadGestacionalSemanas = a.EDAD_GESTACIONAL_SEMANAS,
        PesoPreGestacional = a.PESO_PRE_GESTACIONAL,
        FechaResultadoHemoglobina = a.FECHA_RESULTADO_HEMOGLOBINA,
        PesoKg = a.PESO_KG,
        IdTipoEmbarazo = a.ID_TIPO_EMBARAZO,

        NroControles = a.NRO_CONTROLES,
        SachetsMm = a.SACHETS_MM,
        ConsejeriasNutricionales = a.CONSEJERIAS_NUTRICIONALES,
        SesionesDemostrativas = a.SESIONES_DEMOSTRATIVAS,
        VisitasDomiciliarias = a.VISITAS_DOMICILIARIAS,

        Crecer = a.atencion_programa_afiliacion.Where(x =>
        x.ID_PROGRAMA_AFILIACION == 1).Select(x =>
        x.ES_VIGENTE).FirstOrDefault(),
        Sis = a.atencion_programa_afiliacion.Where(x =>
        x.ID_PROGRAMA_AFILIACION == 2).Select(x =>
        x.ES_VIGENTE).FirstOrDefault(),
        Pin = a.atencion_programa_afiliacion.Where(x =>
        x.ID_PROGRAMA_AFILIACION == 3).Select(x =>
        x.ES_VIGENTE).FirstOrDefault(),
        Juntos = a.atencion_programa_afiliacion.Where(x =>
        x.ID_PROGRAMA_AFILIACION == 4).Select(x =>
        x.ES_VIGENTE).FirstOrDefault(),

        CodigoDepartamento = a.paciente.ID_REGION,
        CodigoProvincia = a.paciente.ID_PROVINCIA,
        CodigoDistrito = a.paciente.ID_DISTRITO,
        CodigoLocalidad = a.paciente.ID_LOCALIDAD,
        AltitudMsnm = a.paciente.ALTITUD_MSNM
    }
    )
}
```

Figura N° 4.30: Script de diseño para la implementación de la base de datos

```
/****** Object: Table [diresa].[atencion]    Script Date: 30/05/2017
19:46:43 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [diresa].[atencion](
    [ID_ATENCION] [int] IDENTITY(1,1) NOT NULL,
    [ID_PACIENTE] [int] NULL,
    [ID_ESTABLECIMIENTO] [int] NULL,
    [ID_TIPO_ATENCION] [int] NOT NULL,
    [ID_TIPO_ANEMIA] [int] NULL,
    [ID_GRADO_DESNUTRICION] [int] NULL,
    [ID_TIPO_DESNUTRICION] [int] NULL,
    [ID_TIPO_EMBARAZO] [int] NULL,
    [ID_SEMANA_EPIDEMIOLOGICA] [int] NULL,
    [ID_ESTADO_REGISTRO] [int] NULL,
    [FECHA_ATENCION] [date] NOT NULL,
    [EDAD_MESES] [int] NULL,
    [PESO_KG] [decimal](10, 2) NULL,
    [TALLA_CM] [decimal](10, 2) NULL,
    [HEMOGLOBINA_GR_DL] [decimal](10, 2) NULL,
    [FECHA_RESULTADO_HEMOGLOBINA] [date] NULL,
    [EDAD_ANIOS] [int] NULL,
    [EDAD_GESTACIONAL_SEMANAS] [int] NULL,
    [PESO_PRE_GESTACIONAL] [decimal](10, 2) NULL,
    [NRO_CONTROLES] [int] NULL,
    [SACHETS_MM] [int] NULL,
    [CONSEJERIAS_NUTRICIONALES] [int] NULL,
    [SESIONES_DEMOSTRATIVAS] [int] NULL,
    [VISITAS_DOMICILIARIAS] [int] NULL,
    [HBC] [decimal](10, 2) NULL,
    [P_T] [decimal](10, 2) NULL,
    [T_E] [decimal](10, 2) NULL,
    [P_E] [decimal](10, 2) NULL,
    [ES_VIGENTE] [bit] NOT NULL,
    [FECHA_CREACION] [datetime] NOT NULL,
    [USUARIO_CREACION] [nvarchar](20) NOT NULL,
    [FECHA_MODIFICACION] [datetime] NULL,
    [USUARIO_MODIFICACION] [nvarchar](20) NULL,
    CONSTRAINT [PK_atencion] PRIMARY KEY CLUSTERED
    (
        [ID_ATENCION] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
```

4.2. REVISIÓN DE CÓDIGO

4.2.1. PRUEBAS UNITARIAS

Se elaboraron las pruebas unitarias para las clases entidad requisitos indispensables para los casos de uso descritos, los cuales fueron ejecutados satisfactoriamente como se muestra a continuación.

Creación de la prueba unitaria de la clase Atención

```
[TestClass()]
public class DiresaAplicacionServicioTests: BaseUnitTest
{
    /// <summary>
    /// Se comprueba que no existe la atencion con ID = 1
    /// </summary>
    [TestMethod()]
    public void GetAtencionTest_Null()
    {
        var application = Container.Resolve<IDiresaAplicacionServicio>();

        int IdAtencion = 1;
        var response = application.GetAtencion(IdAtencion);

        Assert.IsNull(response);
    }

    /// <summary>
    /// Se comprueba que la atencion 536516 le corresponde a una gestante
    /// cuyo N° de DNI es: 28311096
    /// </summary>
    [TestMethod()]
    public void GetAtencionTest_DNI_28311096()//Gestante
    {
        var application = Container.Resolve<IDiresaAplicacionServicio>();

        int IdAtencion = 536516;
        var response = application.GetAtencion(IdAtencion);

        Assert.AreEqual(response.NumeroDocumento, "28311096");
    }

    /// <summary>
    /// Se comprueba que la atención 557463 le corresponde a un niño cuyo
    /// N° de DNI es: 77752400
    /// </summary>
    [TestMethod()]
    public void GetAtencionTest_DNI_77752400()//Niño
    {
        var application = Container.Resolve<IDiresaAplicacionServicio>();

        int IdAtencion = 557463;
        var response = application.GetAtencion(IdAtencion);
    }
}
```

```

        Assert.AreEqual(response.NumeroDocumento, "77752400");
    }

    /// <summary>
    /// Se comprueba que la gestante cuyo DNI es: 28311096 tiene una
    atención con ID 536516
    /// </summary>
    [TestMethod()]
    public void GetGestantesTest()
    {
        var application = Container.Resolve<IDiresaAplicacionServicio>();

        var request = new GestantePageRequest { NroDocumento = "28311096",
        Page = 1, PageSize = 10 };
        var response = application.GetGestantes(request);

        Assert.AreEqual(true, response.Any(x => x.IdAtencion == 536516));
    }

    /// <summary>
    /// Se comprueba que el niño cuyo DNI es: 77752400 tiene una atención
    con ID 557463
    /// </summary>
    [TestMethod()]
    public void GetNiniosTest()
    {
        var application = Container.Resolve<IDiresaAplicacionServicio>();

        var request = new NinioPageRequest { NroDocumento = "77752400", Page
        = 1, PageSize = 10 };
        var response = application.GetNinios(request);

        Assert.AreEqual(true, response.Any(x => x.IdAtencion == 557463));
    }

    /// <summary>
    /// Se comprueba que modifica los datos correctamente de la gestante
    /// </summary>
    [TestMethod()]
    public void PostAtencionTest_ID_ATENCION_536516()//Gestante
    {
        var application = Container.Resolve<IDiresaAplicacionServicio>();

        AtencionResponse request = application.GetAtencion(536516);
        var response = application.PostAtencion(Autenticado, request);

        Assert.AreEqual(true, response.Success);
    }

    /// <summary>
    /// Se comprueba que modifica los datos de los niños correctamente
    /// </summary>
    [TestMethod()]
    public void PostAtencionTest_ID_ATENCION_557463()//Niño
    {
        var application = Container.Resolve<IDiresaAplicacionServicio>();

        AtencionResponse request = application.GetAtencion(557463);

```

```

        var response = application.PostAtencion(Autenticado, request);

        Assert.AreEqual(true, response.Success);
    }

    /// <summary>
    /// Se comprueba que recupera los datos de los pacientes gestantes
    /// </summary>
    [TestMethod()]
    public void GetPacienteByDocTest_DNI_28311096()
    {
        var application = Container.Resolve<IDiresaAplicacionServicio>();

        var response = application.GetPacienteByDoc("28311096");

        Assert.AreEqual(response.NumeroDocumento, "28311096");
    }

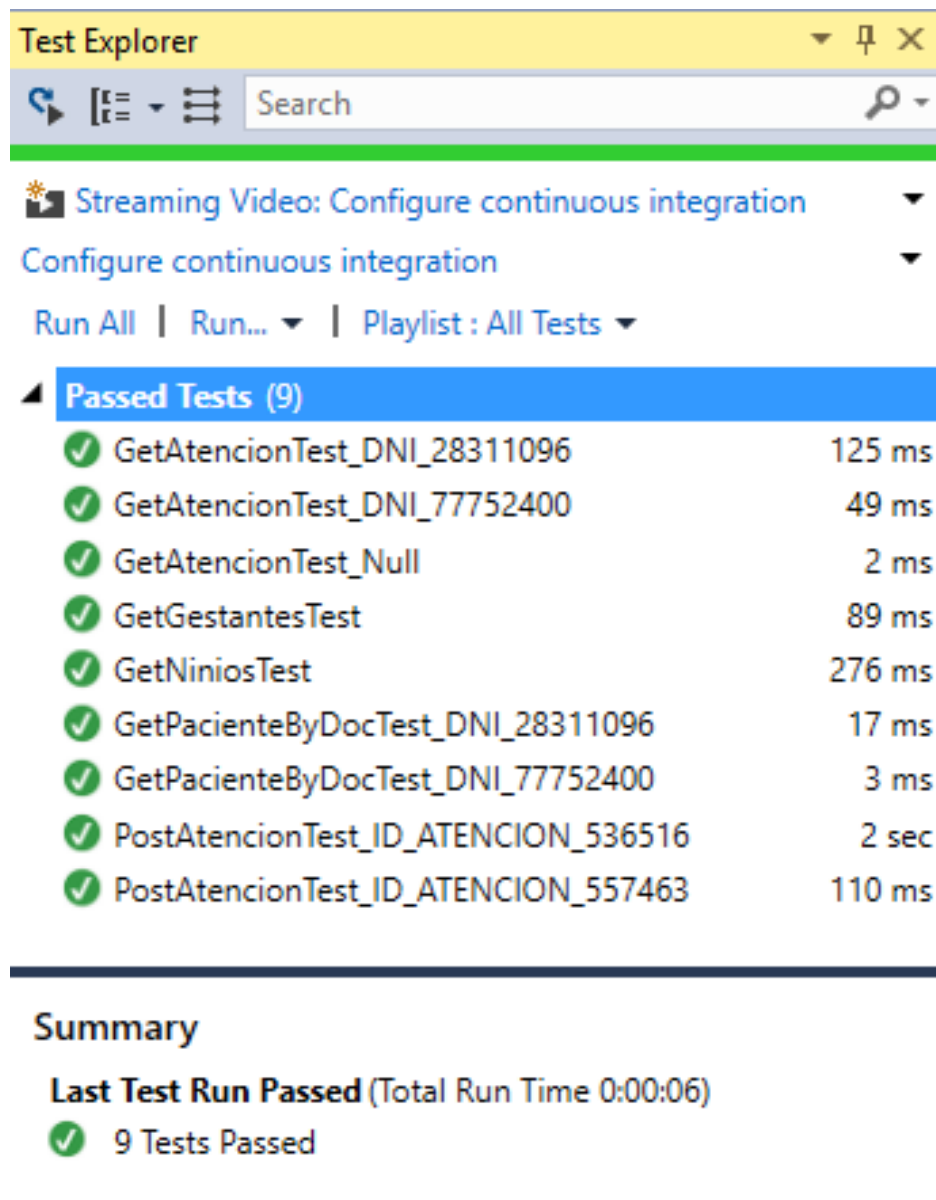
    /// <summary>
    /// Se comprueba que recupera los datos de los niños
    /// </summary>
    [TestMethod()]
    public void GetPacienteByDocTest_DNI_77752400()
    {
        var application = Container.Resolve<IDiresaAplicacionServicio>();

        var response = application.GetPacienteByDoc("77752400");

        Assert.AreEqual(response.NumeroDocumento, "77752400");
    }
}

```

Figura N° 4.31: Ejecución de las pruebas unitarias de los métodos de la clase Atencion.



4.2.2. PRUEBAS FUNCIONALES

Las pruebas funcionales tienen por objetivo probar que los sistemas desarrollados cumplan con las funciones específicas para los cuales han sido creados, a continuación se muestra las pruebas funcionales realizadas.

Tabla N ° 4.14: Pruebas funcionales atención niño

N°	PROPÓSITO	INICIALIZACIÓN	DESCRIPCIÓN DE DATOS DE ENTRADA	RESULTADO ESPERADO	EVALUACIÓN DE PRUEBA
1	Registrar datos de atención del niño	El usuario debe haberse autenticado e iniciado sesión previamente y estar en una sesión activa.	El usuario hace clic en el botón nuevo, el sistema muestra el formulario para poder ingresar los datos requeridos de la atención del niño, el usuario los registra haciendo clic en el botón Guardar.	El sistema muestra un mensaje de confirmación de registro indicando que los datos se guardaron correctamente.	SATISFACTORIA
2	Buscar atención de un niño	El usuario debe haberse autenticado e iniciado sesión previamente y estar en una sesión activa.	El usuario selecciona un tipo de búsqueda e ingresa el criterio de búsqueda requerido, seguidamente hace clic en el botón buscar.	El sistema muestra una grilla con los datos obtenidos por las coincidencias de búsqueda.	SATISFACTORIA
		El usuario debe haberse autenticado	El usuario selecciona un registro de la grilla y el sistema muestra la página de detalle niño	El sistema muestra un mensaje de	

3	Modificar atención de un niño	e iniciado sesión previamente y estar en una sesión activa.	habilitando el formulario para poder realizar modificaciones en los datos del registro seleccionado, el usuario hace clic en el botón guardar.	confirmación de actualización indicando que los datos se modificaron correctamente.	SATISFACTORIA
4	Emitir reporte del estado nutricional de niños	El usuario debe haberse autenticado e iniciado sesión previamente y estar en una sesión activa.	El usuario hace clic en la opción reporte, el sistema muestra la página de reporte de estado nutricional de niños, el usuario selecciona un tipo de búsqueda e ingresa el criterio de búsqueda requerido, seguidamente hace clic en el botón buscar, el sistema muestra una tabla con las coincidencias de la búsqueda, el usuario hace clic en la opción exportar y selecciona el formato del archivo a exportar (Excel o PDF).	El sistema muestra el reporte seleccionado por el usuario en un tipo de archivo plano.	SATISFACTORIA

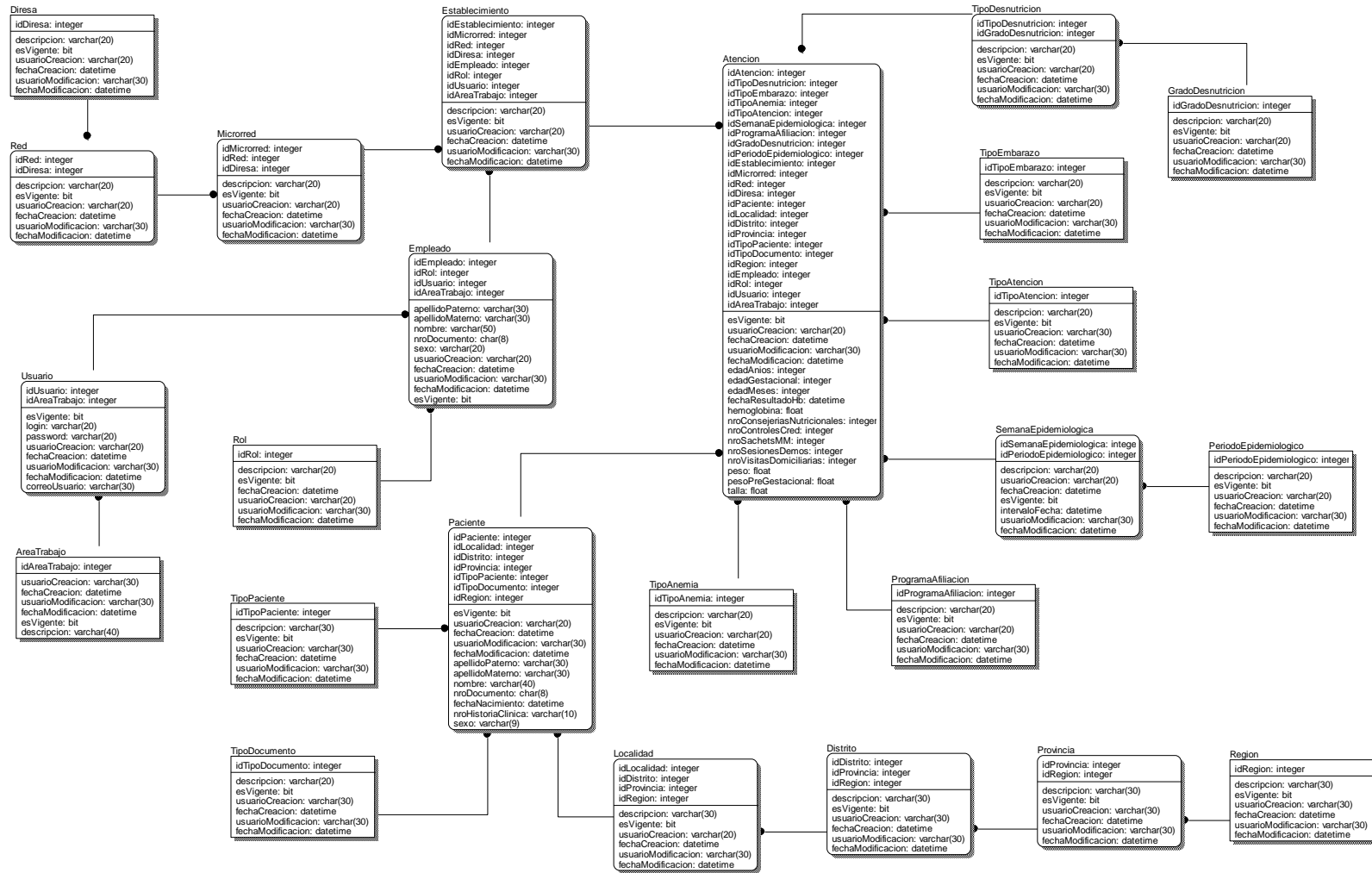
Tabla N ° 4.15: Pruebas funcionales atención gestante.

N°	PROPÓSITO	INICIALIZACIÓN	DESCRIPCIÓN DE DATOS DE ENTRADA	RESULTADO ESPERADO	EVALUACIÓN DE PRUEBA
1	Registrar datos de atención de la gestante	El usuario debe haberse autenticado e iniciado sesión previamente y estar en una sesión activa.	El usuario hace clic en el botón nuevo, el sistema muestra el formulario para poder ingresar los datos requeridos de la atención de la gestante, el usuario los registra haciendo clic en el botón Guardar.	El sistema muestra un mensaje de confirmación de registro indicando que los datos se guardaron correctamente.	SATISFACTORIA
2	Buscar atención de una gestante	El usuario debe haberse autenticado e iniciado sesión previamente y estar en una sesión activa.	El usuario selecciona un tipo de búsqueda e ingresa el criterio de búsqueda requerido, seguidamente hace clic en el botón buscar.	El sistema muestra una grilla con los datos obtenidos por las coincidencias de búsqueda.	SATISFACTORIA
3		El usuario debe haberse autenticado e iniciado sesión previamente y estar en una sesión activa.	El usuario selecciona un registro de la grilla y el sistema muestra la página de detalle gestante habilitando el formulario para poder realizar modificaciones en los datos del registro	El sistema muestra un mensaje de confirmación de actualización indicando que los	SATISFACTORIA

	Modificar atención de una gestante		seleccionado, el usuario hace clic en el botón guardar.	datos se modificaron correctamente.	
4	Emitir reporte del estado nutricional de gestantes	El usuario debe haberse autenticado e iniciado sesión previamente y estar en una sesión activa.	El usuario hace clic en la opción reporte, el sistema muestra la página de reporte de estado nutricional de gestantes, el usuario selecciona un tipo de búsqueda e ingresa el criterio de búsqueda requerido, seguidamente hace clic en el botón buscar, el sistema muestra una tabla con las coincidencias de la búsqueda, el usuario hace clic en la opción exportar y selecciona el formato del archivo a exportar (Excel o PDF).	El sistema muestra el reporte seleccionado por el usuario en un tipo de archivo plano.	SATISFACTORIA

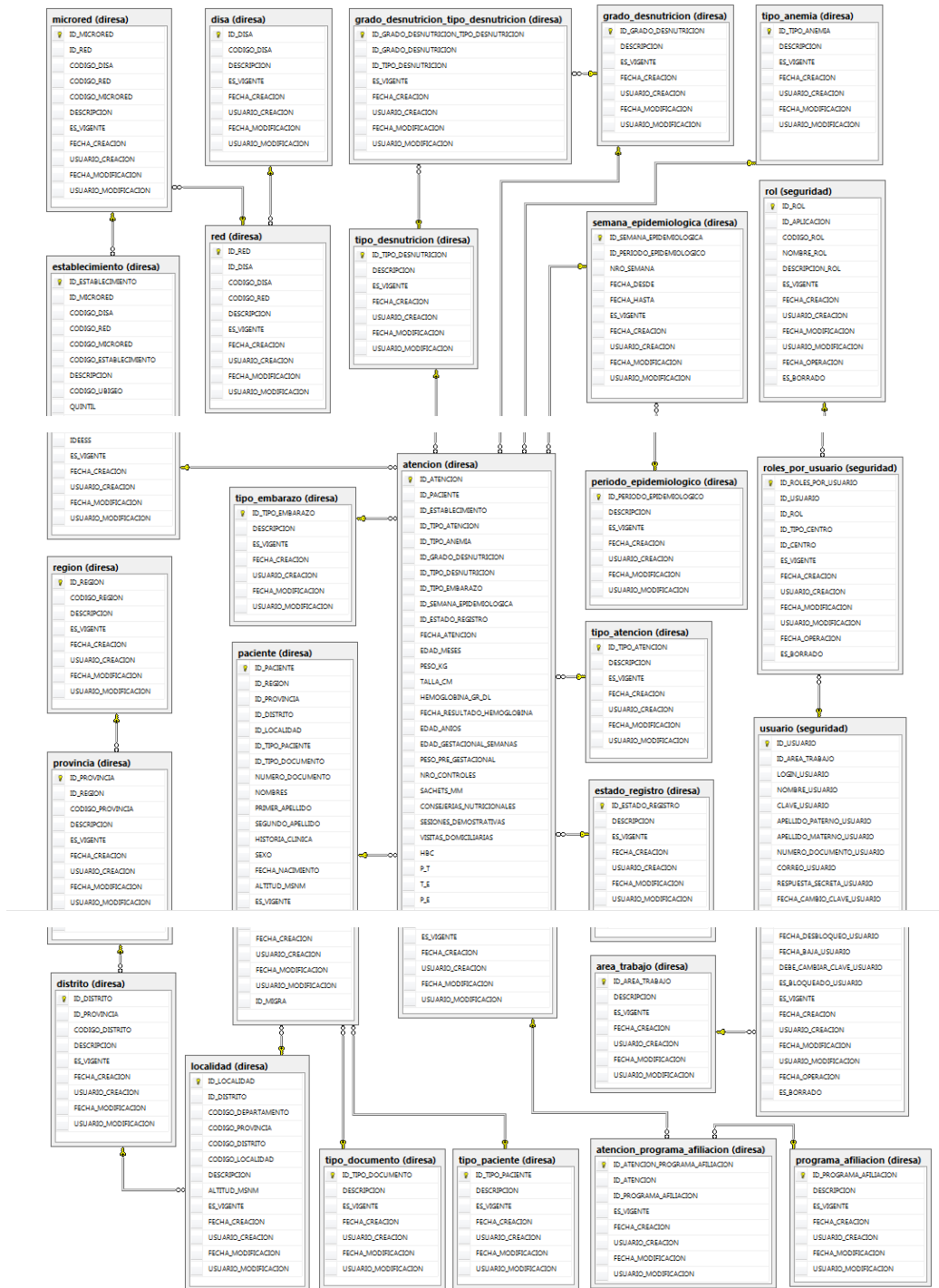
4.2.3. DIAGRAMA FÍSICO DE LA BASE DE DATOS

Figura N° 4.32: Diagrama Físico de la Base de Datos para los casos de uso seleccionados



4.2.4. DIAGRAMA DE LA BASE DE DATOS

Figura N° 4.33: Diagrama de Base de Datos para los casos de uso seleccionados



4.3. RESULTADOS DEL PROCESO DE DESARROLLO DEL SISTEMA

Se aplicó satisfactoriamente la metodología de desarrollo de software Iconix como metodología ágil y formal, para la implementación del sistema web de información georreferenciada del estado nutricional de niños y gestantes de la Dirección Regional de Salud de Ayacucho, siguiendo las actividades definidas por la metodología en cada etapa.

Se desarrolló satisfactoriamente el sistema web en base al lenguaje de programación orientada a objetos C#, además de la implementación de la base de datos de las atenciones de niños y gestantes para la correcta manipulación de los datos a ser procesados por los usuarios del sistema. Se realizó el análisis, diseño e implementación de los módulos de registro, búsqueda, modificación y eliminación de la atención de niños y gestantes, permitiendo de esta manera fortalecer la disponibilidad de información estadística del estado nutricional de niños y gestantes; con la representación de información en un mapa cuantitativo definiendo el mapa a través de divisiones territoriales como son distritos de la provincia de Huamanga.


Tabla N° 4.16: Tiempo estimado para la implementación de software

IMPLEMENTACIÓN DEL SISTEMA WEB DE INFORMACIÓN GEORREFERENCIADA DEL ESTADO NUTRICIONAL DE NIÑOS Y GESTANTES DE LA DIRECCIÓN REGIONAL DE AYACUCHO			
N°	TAREA	ARTEFACTO	TIEMPO ESTIMADO (DÍAS)
1	Identificar las necesidades y requisitos que debe cumplir el sistema.	Requisitos funcionales y no funcionales	4
2	Identificar el mundo real y los objetos de dominio del negocio.	Modelo de dominio	2
3	Definir los requisitos de comportamiento.	Casos de uso	3
4	Realizar análisis de robustez para eliminar ambigüedad de los casos		3

	de uso y determinar los defectos del modelo de dominio.	Diagrama de robustez	
5	Asignar comportamiento a los objetos.	Diagrama de secuencia	3
6	Finalizar el modelo estático.	Diagrama de clases	3
7	Generar código.	Código fuente	30
8	Realizar pruebas de aceptación.	Pruebas funcionales y unitarias	5

Los resultados de la implementación del software fueron los artefactos de la metodología de Iconix: requisitos funcionales para comprender las necesidades para el desarrollo del software en función a la atención del niño y gestante, el diagrama de clases para asignar los atributos sobre la atención, paciente, tipo de paciente, y operaciones de las clases de éstas, la base de datos física para registrar los datos; además de automatizar el proceso de recolección de datos, obtener datos en tiempo real y el reporte del estado nutricional de niños y gestantes en áreas menores por división territorial de acuerdo a los distritos que conforman la provincia de Huamanga.

Figura N° 4.34: Interfaz página niño



Atención
Atención

Usuario: Pilar Gabriela Serrano Quispe

Niños

Criterios de Búsqueda

Nro documento:

+ Nuevo niño
☰

Listado de Resultados

Nro	Fecha de atención	Provincia	Distrito	Documento	Sexo	Edad (meses)	Peso (Kg)	Talla (cm)	Hemoglobina (gr/dl)	Acción
1	31/12/2016	HUAMANGA	VINCHOS	79364840	M	14	9.4	72.2	0	✍️ 🗑️
2	31/12/2016	HUAMANGA	VINCHOS	79837667	M	4	5	57.2	0	✍️ 🗑️
3	31/12/2016	CANGALLO	PARAS	81381354	M	10	8.8	71.2	0	✍️ 🗑️
4	31/12/2016	CANGALLO	PARAS	79870622	M	3	6.5	61.2	0	✍️ 🗑️
5	31/12/2016	CANGALLO	PARAS	78778778	M	27	9.9	83.7	0	✍️ 🗑️
6	31/12/2016	HUAMANGA	OCROS	79500880	M	11	7.6	70	0	✍️ 🗑️
7	31/12/2016	HUAMANGA	OCROS	80855000	M	14	9	74.9	0	✍️ 🗑️
8	31/12/2016	HUAMANGA	OCROS	79828673	F	4	6	60.3	0	✍️ 🗑️
9	31/12/2016	VICTOR FAJARDO	VILCANCHOS	80738814	M	15	9.7	76.5	0	✍️ 🗑️
10	31/12/2016	VICTOR FAJARDO	VILCANCHOS	63160879	F	30	10.2	86	0	✍️ 🗑️

Mostrar:
Total: 98705 filas

<<
<
1
2
3
4
5
>
>>

Figura N° 4.35: Interfaz página detalle niño


The interface is titled "Niños" and includes a header with a logo, a user profile "Usuario: Pilar Gabriela Serrano Quispe", and a search icon. The main content is organized into several sections:

- Datos del niño**:
 - Registro de menor de 5 años (Referencia)**: Includes "Fecha de atención" (calendar), "Semana epidemiológica" (dropdown), "Sexo" (radio buttons for Masculino and Femenino), "Documento" (radio buttons for DNI and CUI), "Fecha de nacimiento" (calendar), "Historia clínica" (text input), "Número de documento" (text input with search icon), "Edad (meses)" (text input), "Nombres" (text input), "Primer apellido" (text input), and "Segundo apellido" (text input).
- Datos antropométricos y de laboratorio**: Includes "Peso (Kg)" (text input), "Talla (cm)" (text input), "Hemoglobina (gr/dl)" (text input), and "Fecha de resultado de hemoglobina" (calendar).
- CRED - desde la consulta anterior, número recibidos de...**: Includes "N° de controles" (text input), "Sachets MM" (text input), "Consejerías nutricionales" (text input), "Sesiones demostrativas" (text input), and "Visitas domiciliarias" (text input).
- Afiliación**: Includes checkboxes for "CRECER", "SIS", "PIN", and "JUNTOS".
- Lugar de residencia**: Includes dropdowns for "Departamento", "Provincia", "Distrito", and "Localidad", and a text input for "Altitud (msnm)".

At the bottom, there are "Cancelar" and "Grabar" buttons.

© LINSCH - DIRESA

Figura N° 4.36: Interfaz página gestante



Atención
Usuario: Pilar Gabriela Serrano Quispe

Gestantes

Criterios de Búsqueda

Nro documento:

+ Nuevo gestante
?

Listado de Resultados

Nro	Fecha de atención	Provincia	Distrito	Documento	Edad (meses)	Edad gestacional (semanas)	Peso (Kg)	Talla (cm)	Hemoglobina (gr/dl)	Acción
1	30/12/2016	HUAMANGA	SAN JUAN BAUTISTA	76037155	22	21	55.5	146	13	
2	30/12/2016	HUAMANGA	SAN JUAN BAUTISTA	73867103	19	13	48.2	158	13	
3	30/12/2016	HUAMANGA	SAN JUAN BAUTISTA	70573284	17	16	51	161.7	0	
4	30/12/2016	HUAMANGA	SAN JUAN BAUTISTA	44037811	30	32	66.5	148	11.8	
5	30/12/2016	HUAMANGA	SAN JUAN BAUTISTA	46094804	26	19	59.7	151	0	
6	30/12/2016	HUAMANGA	JESUS NAZARENO	43484636	33	22	68.5	151	0	
7	30/12/2016	HUAMANGA	JESUS NAZARENO	74347656	16	28	53.2	155	10.7	
8	30/12/2016	HUAMANGA	JESUS NAZARENO	62119633	19	24	59	155	10	
9	30/12/2016	HUAMANGA	JESUS NAZARENO	80045125	39	34	59.8	152	11.2	
10	30/12/2016	HUAMANGA	CARMEN ALTO	41906393	38	14	53.2	151	13.8	

Mostrar: 10 Filas Total: 20945 filas

<< < 1 2 3 4 5 > >>

© UNSCH - DIRESA

Figura N° 4.37: Interfaz página detalle gestante

Gestantes

Atención

Usuario: Pilar Gabriela Serrano Quispe

+ Nuevo gestante

Buscar Limpiar

Criterios de Búsqueda

Nro documento:

Listado de Resultados

Nro	Fecha de atención	Provincia	Distrito	Documento	Edad (meses)	Edad gestacional (semanas)	Peso (Kg)	Talla (cm)	Hemoglobina (gr/dl)	Acción
1	30/12/2016	HUAMANGA	VINCHOS	70825842	18	30	60	151	13.8	
2	30/12/2016	HUAMANGA	VINCHOS	71534842	19	39	60.5	163	0	
3	30/12/2016	HUAMANGA	AYACUCHO	46240396	27	26	66.1	150	0	
4	30/12/2016	HUAMANGA	SAN JUAN BAUTISTA	43108624	31	24	57.1	157	13	
5	30/12/2016	HUAMANGA	SAN JUAN BAUTISTA	70683314	18	14	39.8	154	0	
6	30/12/2016	HUAMANGA	SAN JUAN BAUTISTA	76037155	22	21	55.5	146	13	
7	30/12/2016	HUAMANGA	SAN JUAN BAUTISTA	73867103	19	13	48.2	158	13	
8	30/12/2016	HUAMANGA	SAN JUAN BAUTISTA	70573284	17	16	51	161.7	0	
9	30/12/2016	HUAMANGA	SAN JUAN BAUTISTA	44037811	30	32	66.5	148	11.8	
10	30/12/2016	HUAMANGA	SAN JUAN BAUTISTA	46094804	26	19	59.7	151	0	

Mostrar: 10 Filas Total: 20945 filas

<< < 1 2 3 4 5 > >>

© UNSCH - DIRESA

Figura N° 4.38: Interfaz reporte del estado nutricional de niños y gestantes

Reportes

Seleccione el reporte

Reporte:

Reporte

Geográfico

Mapa Satélite

VINCHOS

Proporción de anemia					
	Total con anemia	Leve	Moderada	Severa	Atenidos
Niños menores de 5 años	82.952193	6.016877	3.130733	73.804574	8177
Gestantes	61.762841	6.911858	3.868104	50.962879	1577

Proporción de desnutrición de niños					
Total niños	Aguda	Crónica	Global	Obesidad	Sobrepeso
8177	1.161795	33.863275	4.989605	0.904977	3.326403

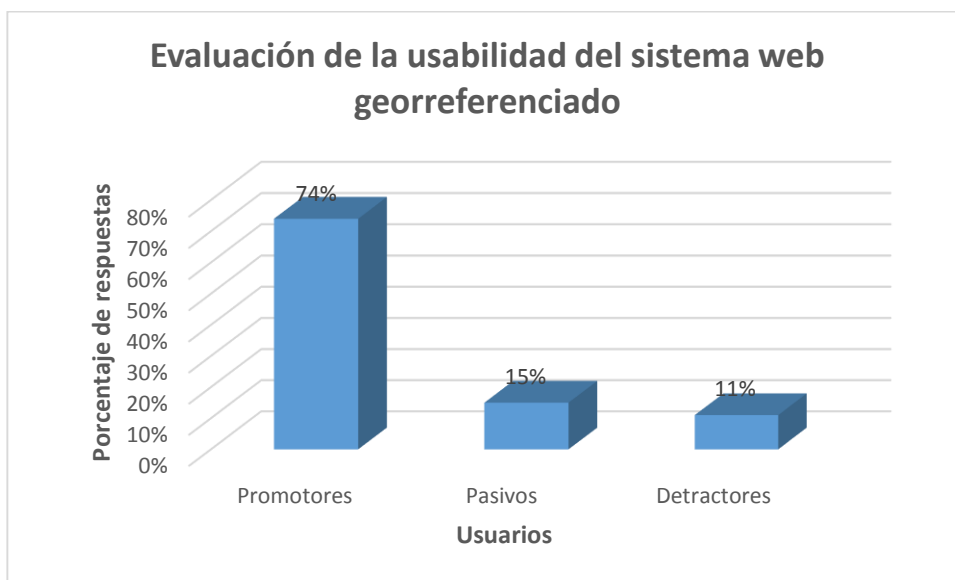
Proporción de estado nutricional de gestantes			
Total gestantes	Sobrepeso	Normal	Deficit
1577	10.209258	78.01078	10.779962

© UNSCH - DIRESA

4.4. RESULTADOS DE LA EVALUACIÓN DE USABILIDAD DEL SISTEMA WEB GEORREFERENCIADO

Al realizar la evaluación de usabilidad del sistema web georreferenciado se obtuvo la siguiente figura.

Figura N° 4.39: Medición de la usabilidad mediante NPS



NPS=Promotores – Detractores

NPS=74% - 15%

NPS= 59%

Se observa que el índice de promoción neto o net promoter score (NPS) es 59%, lo que significa que se puede afirmar que el “sistema web de información georreferenciada del estado nutricional de niños y gestantes de la Dirección Regional de Salud de Ayacucho” es usable, es decir entendible, de fácil navegación y de contenidos pertinentes para el usuario.

4.5. DISCUSIÓN DE RESULTADOS

La necesidad de información geográfica en áreas de salud es muy importante, tanto en la planificación como en la evaluación de la situación de salud. En caso del estado nutricional de niños y gestantes, los factores de riesgo de vulnerabilidad a la desnutrición se verifican siempre a través del territorio, es decir que para su conocimiento y gestión hay que considerar las variables geográficas y su interconexión espacial.

Frente a la cartografía convencional, y en lo que interesa directamente al sector salud, los Sistemas de Información Geográfica (SIG) ofrecen ventajas estratégicas. Permiten, una fácil actualización de la información contenida en cada capa o cobertura, frente al carácter estático de la información contenida en un mapa en papel. Este es un aspecto fundamental, ya que facilita el análisis espacial de la situación de salud, la investigación operacional y la vigilancia con vistas a la prevención y al control de problemas relacionados con el estado nutricional de niños y gestantes; facilitando sensiblemente la toma de decisiones.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- a. La implantación de herramientas basadas en software fue muy importante para llevar a cabo el desarrollo de la presente investigación. La madurez actual de las herramientas de Sistemas de Información Geográfica en este modelo de mercado, permite obtener aplicaciones profesionales de gran calidad. Como se ha demostrado con el desarrollo de la investigación es posible disponer y proveer la mayor parte de funcionalidades para el almacenamiento, procesamiento, representación, publicación y análisis e integración de la información geográfica.
- b. Se logró realizar la capacitación a cerca del manejo de la información georreferenciada para que el usuario responsable pueda monitorear y obtener los reportes del estado nutricional de niños y gestantes para mejorar los planes de control asistencial del estado nutricional de niños y gestantes haciendo mejor su formación y control de los mismos
- c. Al contar con información georreferenciada en tiempo real del estado nutricional de los niños y gestantes se podrá realizar un análisis de la situación actual y se procederá a realizar mejoras en los planes de organización, asimismo se podrá definir el recurso necesario para contrarrestar la situación y se tengan estándares de salud que permitirán brindar una mejor calidad de vida.
- d. Los mapas cuantitativos nos permiten ubicarnos espacialmente, lo cual ayuda a los directivos a tomar mejores decisiones en cuanto se refiere a la priorización y focalización de poblaciones más vulnerables para la distribución de recursos financieros en aras del beneficio de la población afectada.

5.2. RECOMENDACIONES

- a. La oficina de Estadística, Informática y Telecomunicaciones de la Dirección Regional de Salud de Ayacucho debe capacitar al personal administrativo para el manejo y administración del sistema web para la gestión de información del estado nutricional de niños y gestantes.

- b. Gestionar la adquisición de servidores dedicados tanto para los sistemas como para la base de datos, con la finalidad de garantizar el normal desenvolvimiento del sistema, priorizando performance y calidad para su funcionamiento.

- c. La oficina de Estadística, Informática y Telecomunicaciones de la Dirección Regional de Salud de Ayacucho debe fomentar el uso de sistemas de información para la automatización de procesos de las diferentes áreas, con la finalidad de mejorar la productividad y la eficiencia de tareas específicas ya existentes.

BIBLIOGRAFÍA

1. Alvarez, M. (2008). *Introducimos para los más profanos las bases sobre las que se asienta la Programación Orientada a Objetos*. Recuperado de <http://www.desarrolloweb.com/articulos/499.php>
2. Allamaraju et al. (2002). *Profesional programación Java Server con J2EE edición 1.3*. (1ra Edición). Anaya Multimedia.
3. Amaya, J. (2010). *Sistema de Información Gerencial: Hardware, software, redes, Internet, diseño*. Bogotá, Colombia: Ecoe Ediciones.
4. ATLAS SIG HYPARION (s.f.). *Mapas*. Recuperado de <http://www.hyparion.com/web/diccionario/dics/cartografia/mapa.htm#tematico>
5. Baltazar A. (2007). *Obesidad y Cirujía*. Recuperado de http://books.google.com.pe/books?id=3XpacoK8H1sC&pg=PA15&dq=definicion+obesidad&hl=es&ei=JdY4Tu24E4Ps0gGh_PzQAw&sa=X&oi=book_result&ct=result&resnum=1&ved=0CCwQ6AEwAA#v=onepage&q&f=false
6. Bauer, F. (2002). *Software engineering, Information processing*. Amsterdam, Holanda: North Holland.
7. Bartle, P (2009). *Información para la gestión*. Recuperado de <http://www.scn.org/mpfc/modules/mon-miss.htm>
8. Berners-Lee, T. (2010). *Long Live the Web. Scientific American*. Recuperado de http://www.cs.virginia.edu/~robins/Long_Live_the_Web.pdf
9. Berzal, F., Cortijo, F. y Cubero, J. (2003). *Desarrollo Profesional de Aplicaciones Web con ASP.NET*. (1ra Edición). Recuperado de <http://elvex.ugr.es/decsai/csharp/pdf/web/web-book-a4.pdf>
10. Booch, G. (1986). *Object Oriented Development. Trans. of Soft. Eng.* Vol. SE-12. Num. 2. Recuperado de <http://cs.txstate.edu/~rp31/papersSP/BoochOOD003.pdf>
11. Booch, G. (1995). *Análisis y diseño orientado a objetos con aplicaciones*. Madrid, España: Diaz de Santos/Addison-Wesley.
12. Bosque, J. (2000). *Sistemas de Información Geográfica*. Madrid, España: Rialp.
13. Burstein, T. (2002). *Sistemas de información geográfica y su aplicación en la salud pública*, *Revista Peruana de Medicina Experimental y Salud Pública*, vol. 19, núm. 3, p. 107.

14. Capote, Carrillo, Ruiz, Rodríguez, Miranda. (2008). *Introducción a los sistemas de base de datos*. Madrid, España: Paraninfo.
15. Ceballos, F. (2006). *Lenguaje de Programación C#*. España, Madrid: Ra-Ma.
16. Centro Nacional de Alimentación y Nutrición. (s.f.). Recuperado de <http://www.portal.ins.gob.pe/es/cenan>
17. Cerezo, Y., Peñalba, O., Caballero, R. (2006). *Iniciación a la programación en C#: un enfoque práctico*. Madrid, España: Delta publicaciones.
18. Chiavenato, I. (2006). *Introducción a la Teoría General de la Administración*. (7ma. Ed.) Cuajimalpa, México: McGraw-Hill Interamericana.
19. Comas, D. y Ruiz, E. (1993). *Fundamentos de los Sistemas de Información Geográfica*. Barcelona, España: Ariel
20. Cota, A. (1994). *Ingeniería de Software. Soluciones Avanzadas*. Recuperado de <http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html#IngSoft>
21. Curto, J. (2006). *Information Management*. Recuperado de <http://informationmanagement.wordpress.com/category/gestion/gestion-de-la-informacion/>
22. ESRI (2002). *¿Qué es ArcGis? – SIG de ESRI*. Recuperado de <http://www.esri.com>
23. Fernández, V. (2006). *Desarrollo de sistemas de información: una metodología basada en el Modelado* (4ª Ed.). Barcelona, España: UPC.
24. Florez, R. (2005). *Algoritmos, Estructuras de datos Programación Orientado a Objetos* (4ª Ed.). Bogotá, Colombia: Ecoe Ediciones.
25. IEEE (1993). *Standards Collection: Software Engineering*. Recuperado de https://standards.ieee.org/findstds/standard/software_and_systems_engineering_all.html
26. Instituto Nacional de Estadística e Informática (2009a). *Mapa de Desnutrición Crónica en Niños Menores de 5 años a nivel Provincial y Distrital, 2007*. Lima, Perú: Talleres de la Oficina Técnica de Administración del INEI.
27. INEI (2009b). *Encuesta Demográfica de Salud Familiar*. Lima, Perú: Dirección Técnica de Demografía e Indicadores Sociales.
28. Instituto Nacional de Salud (2011). *MedlinePlus*. Recuperado de <http://www.nlm.nih.gov/medlineplus/spanish/ency/article/000404.htm>
29. Instituto Nacional de Salud. (s.f.). Recuperado de <http://www.portal.ins.gob.pe/es/acerca-de-ins/ins-c1/acerca-del-ins/identidad>

30. Joyanes, L., Zahonero, I. (2002). *Programación en Java 2 Algoritmos, Estructura de Datos y Programación Orientada a Objetos* (1ra. Edición). Madrid, España: McGraw Hill/ Interamericana de España.
31. Joyanes, L. (2003). *Fundamentos de programación Algoritmos, Estructura de datos y Objetos* (3ra Edición). Madrid, España: McGraw Hill/ Interamericana de España.
32. Kendall, K. y Kendall, J. (2011). *Análisis y diseño de sistemas (8va edición)*. Mexico: Pearson.
33. Lagos, M. (2010). *Aplicación web para apoyar la cadena productiva de la palta en la ciudad de Huanta*. Tesis de pregrado, Universidad Nacional de San Cristóbal de Huamanga, Ayacucho, Perú.
34. Longhi, F. (2012). *Sistemas de Información Geográfica y desnutrición infantil en el Norte Grande Argentino*. Revista Latinoamericana de Población, Enero-Junio, 77-101.
35. Majem, Ll. y Aranceta, J. (2006). *Nutrición y salud pública: métodos, bases científicas y aplicaciones*. Barcelona, España. Grafiques 92 S.A.
36. Martín, A. (2007). *Desarrollo de Aplicaciones WEB con ASP.NET 2.0* (1ra Edición). España, Madrid: Ra-Ma.
37. Microsoft (2016). *Introducción a .NET Framework*. Recuperado de [https://msdn.microsoft.com/es-es/library/hh425099\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/hh425099(v=vs.110).aspx)
38. Ministerio de Salud (2001). *Lineamientos para la conformación de Redes de Salud* [Resolución Ministerial N° 122-2001-SA/DM] Lima, Perú: Ministerio de Salud.
39. Ministerio de Salud (s.f.). *Direcciones de salud*. Recuperado de <https://www.minsa.gob.pe/portalminsa/directorioinstitucional/nododis.asp?nodo=0201>
40. Ministerio de Salud (s.f.). *Reglamento de organizaciones y funciones*. Recuperado de ftp://ftp2.minsa.gob.pe/descargas/dgps/documentos/reglamento_de_organizacion_y_funciones.pdf
41. Mori, M. (s.f.). *Introducción a los mapas cuantitativos*. Recuperado de <http://es.slideshare.net/pasteurt/002-presentacionmapastematicos>
42. Ochoa, L. (2003). *Sistemas de información geográfica, ventajas y desventajas de su utilización en Guatemala*. Tesis de maestría no publicada. Universidad de San Carlos de Guatemala. Guatemala.
43. Oficina de Epidemiología de la DIRESA (2008). *Análisis de la Situación de Salud de Ayacucho – 2007*. Ayacucho, Perú: DSG e Vargas

44. Oficina Nacional de Gobierno Electrónico e Informática (2014). *Sistema Georreferenciado Sayhuite*. Lima, Perú: Presidencia del Consejo de Ministros.
45. Ortega R. y Requejo A. (2000). *Nutriguía: manual de nutrición clínica en atención primaria*. Recuperado de http://books.google.com.pe/books?id=x2Cll3457MsC&pg=PA117&dq=definicion+sobre+peso&hl=es&ei=tNc4ToKrCurk0QHJrK3SAw&sa=X&oi=book_result&ct=result&resnum=1&ved=0CCgQ6AEwAA#v=onepage&q&f=false
46. Porras, E. (2011). *La Metodología Ágil y Formal ICONIX para el Desarrollo de Software: Teoría y Práctica* (1° Ed.). Ayacucho, Perú: Ami Ayacucho.
47. Programa Especial para la Seguridad Alimentaria (2002). *Conceptos Seguridad Alimentaria*. Guatemala, C.A.
48. Pastor, J. (s,f.). **Concepto de Sistema de Información en la organización**. Recuperado de https://books.google.es/books?id=Ki_YBjGVlH4C&printsec=frontcover&dq=concepto+de+sistema+de+informacion+en+la&hl=es&ei=KreDTri1Ocb2gAfv1sQy&sa=X&oi=book_result&ct=book-thumbnail#v=onepage&q&f=false
49. Pressman, R (2010). *Ingeniería Del Software: Un Enfoque Práctico* (7ma Ed.). Aravaca, Madrid: McGraw-Hill.
50. Real Academia Española (2001). *Diccionario de la lengua española* (22.ªed.). Recuperado de <http://www.rae.es/>
51. Ricardo, K (2009). *Bases de datos*. México, D, F: McGraw-Hill.
52. Rodríguez C. (2007). *Anemia y Cáncer*. Recuperado de http://books.google.com.pe/books?id=XeMuX4gn604C&printsec=frontcover&dq=anemia&hl=es&ei=FtQ4TsnhDer00gGHtc2hBA&sa=X&oi=book_result&ct=result&resnum=1&ved=0CCgQ6AEwAA#v=onepage&q&f=false
53. Rosenberg, D., Stephenns, M., Collins-Cope, M. (2005). *Agile Development with ICONIX Process—People, Process, and Pragmatism*. New York, USA: Apress.
54. Rosenberg, D. y Stephens, M. (2007). *Use Case Driven Object Modeling with UML: Theory and Practice* (1° Ed.). New York: USA: Apress.
55. Segura, L., Montes, C., Hilario, M., Asenjo, P. y Baltazar, G. (2002) *Pobreza y desnutrición infantil*. Perú, Lima: Editorial Supergráfica EIRL.

56. SICORI (2001). **Definición SIG.** Recuperado de http://hades.inegi.gob.mx/sitio_inegi/informatica/espanol/tidap2002/tidap2001/tidap01/SIG
57. Sommerville, I. (2011). **Ingeniería de Software** (9na Ed.). Naucalpan de Juárez, Estado de México: Pearson Educación.
58. Tamayo, A. (2000). **Sistemas de Información** (1ª Ed.). Editorial McGraw Hill.
59. Tanenbaum, A. (2003). **Redes de Computadoras** (4ª Ed.). México: DF. Editorial Pearson
60. Usandivaras, J. (s.f). **Plataforma de eLearning sobre Geoinformación.** Recuperado de <http://pdi.topografia.upm.es/mab/tematica/htmls/inicial.html>
61. Weitzenfeld, A. (2002). **Ingeniería de Software Orientado a Objetos - Teoría y Práctica con UML y Java.** (1ra. Edición). México: Itam.
62. Winblad, A., King, D (2003). **Software Orientado a Objetos** (7ma Ed.). Madrid, España: Addison-Wesley.
63. Whitten, J., Bentley, L. y Dittman K. (2004). **System analysis & design methods.** Boston, EE.UU: McGraw-Hill Irwin.
64. FAO (2002). **Nutrición Humana en el Mundo en Desarrollo.**
65. OPS (1995). **Manual de Crecimiento y Desarrollo del Niño.** Sexta Edición

ANEXO A: Planificación del proceso Iconix para el proyecto

Tabla A.1: Planificación del proceso Iconix

FASES	FECHA	
	INICIAL	FECHA FINAL
ANÁLISIS DE REQUERIMIENTOS	2016/09/10	2016/10/02
Modelo del dominio	2016/09/11	2016/09/12
Modelo de Casos de Uso	2016/09/13	2016/09/22
Revisión de requerimientos	2016/09/23	2016/10/02
ANÁLISIS Y DISEÑO PRELIMINAR	2016/10/03	2016/10/25
Análisis de Robustez	2016/10/03	2016/10/10
Actualización del Modelo del Dominio	2016/10/11	2016/10/18
Revisión del diseño preliminar	2016/10/19	2016/10/25
DISEÑO DETALLADO	2017/10/26	2014/11/25
Diagrama de secuencia	2017/10/26	2017/11/17
Finalización del Modelo de clases	2017/11/18	2017/11/23
Revisión del Diseño Crítico	2017/11/24	2017/11/25
IMPLEMENTACIÓN Y PRUEBAS	2016/11/26	2017/03/03
Codificación de los casos de uso	2017/11/26	2017/03/06
Pruebas Unitarias	2017/03/07	2017/03/09

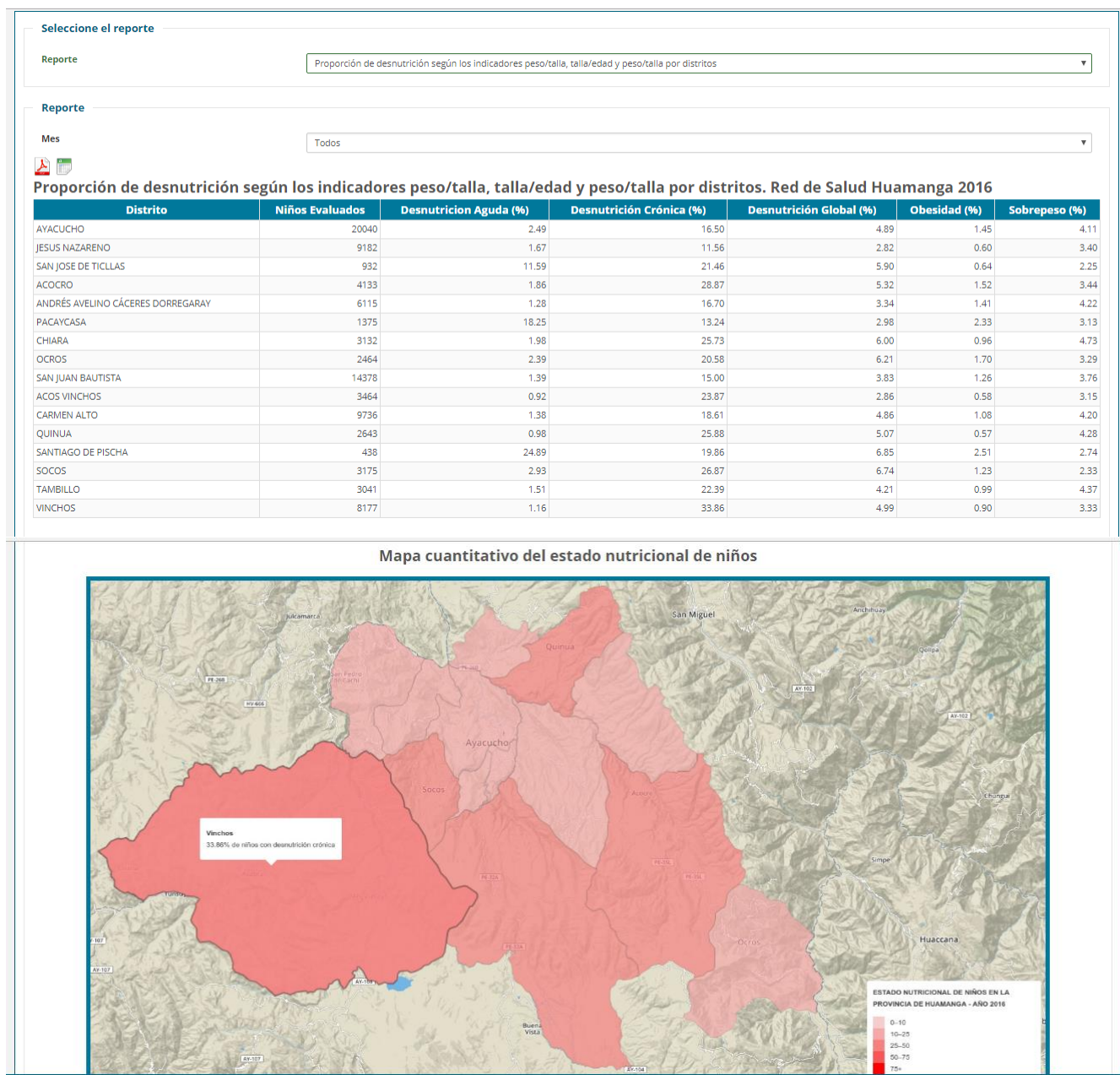
ANEXO B: Calendario epidemiológico – 2016

Figura B.1: Calendario epidemiológico – 2016



ANEXO C: Reporte obtenido con el “Sistema Web de Información Georreferenciada del Estado Nutricional de Niños y Gestantes”

Figura C.1: Reporte del estado nutricional de niños por distritos. Periodo: Enero- Diciembre 2016.



**ANEXO D: Reporte obtenido con la Aplicación de escritorio Sistema Información del Estado Nutricional de Niños y Gestantes –
propio de la DIRESA**

Figura D.1: Reporte obtenido con la Aplicación de escritorio SIEN – propio de la DIRESA - Periodo: Enero- Diciembre 2016

REPORTE MENSUALIZADO DEL ESTADO NUTRICIONAL DEL NIÑO MENOR DE 5 AÑOS POR DISTRITO								
DIRESA :	AYACUCHO							
RED :	HUAMANGA							
MENSUALIZADO :	ENERO - DICIEMBRE					AÑO:	2016	
ESTADO NUTRICIONAL DE NIÑOS								
PROVINCIA :	HUAMANGA							
Distrito	Nro de Evaluad	Crónico %	Global %	Agudo %	Sobre Peso %	Obesidad %		
ACOCRO	4133	28,9	5,3	1,9	3,4	0,9		
ACOS VINCHOS	3464	23,9	2,9	0,9	3,1	0,6		
ANDRES A VELINO CÁCERES DORREGARAY	6115	16,7	3,3	1,3	4,2	1,4		
AYACUCHO	20040	16,5	4,9	2,5	4,1	1,5		
CARMEN ALTO	9736	18,6	4,9	1,4	4,2	1,1		
CHIARA	3132	25,7	6,0	2,0	4,7	1,0		
JESUS NAZARENO	9182	11,6	2,8	1,7	3,4	0,6		
OCROS	2464	20,6	6,2	2,4	3,3	1,7		
PACAYCASA	1375	13,2	3,0	18,3	3,1	2,3		
QUINUA	2643	25,9	5,1	1,0	4,3	0,6		
SAN JOSE DE TICLLAS	932	21,5	5,9	11,6	2,3	0,6		
SAN JUAN BAUTISTA	14378	15,0	3,8	1,4	3,8	1,3		
SANTIAGO DE PISCHA	438	19,9	6,8	24,9	2,7	2,5		
SOCOS	3175	26,9	6,7	2,9	2,3	1,2		
TAMBILLO	3041	22,4	4,2	1,5	4,4	1,0		
VINCHOS	8177	33,9	5,0	1,2	3,3	0,9		
TOTAL	92425							
						Referencia:	OSM	
SIEN : SISTEMA INFORMATICO DEL ESTADO NUTRICIONAL								Page -1 of 1

ANEXO E: Tecnologías con la que cuenta cada Microrred de Salud

Figura E.1: Tecnologías con la que cuenta cada Microrred de Salud

RED DE SALUD DE HUAMANGA					
N°	MICRORRED DE SALUD	ACCESO A INTERNET	EQUIPO DE COMPUTO PROPIO		PROCESADOR
1	BELEN	SI	SI		INTEL® i3 - i5
2	CARMEN ALTO	SI	SI		INTEL® i3 - i5
3	CHIARA	SI	SI		INTEL® i3 - i5
4	CHONTACA	SI	SI		INTEL® i3 - i5
5	LICENCIADOS	SI	SI		INTEL® i3 - i5
6	MUYURINA	SI	SI		INTEL® i3 - i5
7	OCROS	SI	SI		INTEL® i3 - i5
8	PARAS	SI	SI		INTEL® i3 - i5
9	PUTACCA	SI	SI		INTEL® i3 - i5
10	QUINUA	SI	SI		INTEL® i3 - i5
11	SAN JUAN BAUTISTA	SI	SI		INTEL® i3 - i5
12	SANTA ELENA	SI	SI		INTEL® i3 - i5
13	SOCOS	SI	SI		INTEL® i3 - i5
14	TOTOS	SI	SI		INTEL® i3 - i5
15	VINCHOS	SI	SI		INTEL® i3 - i5