

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Contaduría y Administración

Facultad de Ciencias Químicas e Ingeniería

Maestría en Tecnologías de la Información y la Comunicación



Desarrollo de mapa interactivo de difusión a la actividad agropecuaria y pesquera en Baja California enfocado al sector educativo.

TESIS QUE PARA OBTENER EL GRADO DE
MAESTRO EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

Presenta:

Juan José Raymundo Torres

Bajo la dirección de:

Dra. Margarita Ramírez Ramírez

M. Comité

Dra. Dora Luz Flores Gutiérrez

Codirección:

Dra. Ana Isabel Acosta Martínez

M. Comité

M.C.C. Esperanza Manrique Rojas.

Dedicatorias y agradecimientos

A mi madre.

Por haberme apoyado en todo momento, por sus consejos, sus valores y la motivación constante que me ha permitido ser una persona de bien, pero más que nada por su confianza incondicional.

A mis Maestros.

A la Dra. Margarita Ramírez por su gran apoyo y motivación para la culminación del postgrado y para la elaboración de esta tesis; a la Dra. Esperanza Manríquez, Dra. Ana Acosta y Dra. Dora luz Flores por sus comentarios y apoyo ofrecido durante la elaboración el presente trabajo de tesis.

A mis amigos.

Que con sus buenos deseos y ánimos incentivaban en mí el terminar una meta más en mi vida profesional.

Juan José Raymundo Torres

Resumen

El presente trabajo de tesis nace como una vertiente del proyecto nombrado “Producción y consumo del sector agropecuario de Baja California. Estrategias para su fortalecimiento y vinculación”, el cual es una iniciativa que el cuerpo académico: “Desarrollo y competitividad agroalimentaria en colaboración”, con la participación de integrantes del Cuerpo académico Sistemas de Información y gestión empresarial de la Facultad de Contaduría y Administración. Este proyecto participó en la 15va. Convocatoria interna de apoyo a trabajos de investigación, resultando seleccionada, cabe mencionar que esta investigación es un esfuerzo de La Universidad Autónoma de Baja California en conjunto con la Universidad de Murcia, España y con el Consejo Agrícola de Baja California, A.C.

A nivel internacional, sobre todo en los países desarrollados, la industria y las organizaciones gubernamentales y no gubernamentales colaboran para encontrar la forma de incrementar el consumo como parte de las estrategias de transformar el estilo de vida de la población y combatir la obesidad por la ingesta de alimentos de alto contenido calórico y las enfermedades crónicas que se relacionan (Avendaño, 2010).

México cuenta con una sociedad rural que goza de calidad de vida atractiva y con diversidad de oportunidades de desarrollo así como un sector agroalimentario y pesquero rentable y sustentable que ofrece alimentos accesibles, sanos y de calidad a sus habitantes (SAGARPA, 2010).

Es por todo esto que proponemos una estrategia que busque captar la atención de aquellos usuarios que estén en el nivel de educación básica, mediante interfaces multimedia amigables y atractivas visualmente. El sistema informará a los usuarios sobre los productos agropecuarios que la región ofrece, así como su valor nutricional. El sistema está pensado para desarrollarse basado en capas y así permitir una escalabilidad, esto es mostrar la información adecuada al usuario que ingrese al sistema. Como primera fase este proyecto contempla el desarrollo de una

primera capa multimedia donde se muestra información de los productos agropecuarios producidos en la región de una manera gráfica y amigable dirigida a los alumnos del nivel de educación básica.

Con dicho sistema se espera apoyar los esfuerzos por fortalecer al sector agropecuario de Baja California, al dar a conocer la actividad productiva y su potencial a la sociedad de consumo al vincularse con el sector educativo permitiendo que se forme una conciencia de consumo de productos regionales y de ahí, incentivar su demanda y nivel de producción y por ende mejorar la economía del sector y la región.

Actualmente la educación es asistida por los medios multimedia. La educación actual plantea al alumno como el centro de los procesos educativos (Guiza E., 2002).

El beneficio más importante de la multimedia es que permite enriquecer la experiencia del usuario-receptor, logrando una asimilación más fácil y rápida de la información presentada. Esto es bastante claro en las aplicaciones de tipo formativa o educacional (Gayeski, 1992).

Índice

Capítulo 1	11
Introducción.....	11
1.1 <i>Antecedentes</i>	13
1.1.1 El Sector Agropecuario en México.....	13
1.1.2 Consumo de frutas y hortalizas	13
1.1.3 Introducción a la multimedia	15
1.1.4 Multimedia como herramienta de apoyo al aprendizaje.....	17
Otro tema que es necesario conocer para ubicarnos en el contexto en el cual es desarrollado.....	20
el presente sistema es lo relativo al sistema de información geográfica del sector.....	20
agropecuario de B.C., que a continuación se detalla.....	20
1.1.5 Sistema de Información Geográfica del Sector Agropecuario de B.C.....	20
1.2 <i>Planteamiento del problema</i>	22
1.3 <i>Justificación</i>	23
1.4 <i>Objetivos</i>	24
1.4.1 Objetivo general	24
1.4.2 Objetivos específicos	24
Capítulo 2	25
Marco teórico.....	25
2.1 <i>Introducción a la multimedia</i>	25
2.1.1 Multimedia en los procesos de enseñanza-aprendizaje	26
2.1.2 ¿Qué se entiende hoy por multimedia?	26
2.1.3 Diseño de sistemas multimedia	27
2.2 <i>Internet como recurso educativo</i>	29
2.3 <i>Introducción a las versiones de HTML</i>	31
2.3.1 ¿Qué es y cómo nace Adobe Flash?	33
2.3.2 Contenido multimedia websin Flash.....	33
2.4 <i>Introducción a CSS</i>	35
2.4.1 Breve Historia de CSS.....	35
2.4.2 ¿Qué es CSS?	35
2.4.3 Soporte de CSS en los Navegadores	36
2.5 <i>Introducción a Javascript</i>	37
2.5.1 Breve Historia de Javascript.....	37
2.5.2 ¿Qué es JavaScript?.....	37
2.6 <i>HTML5 + CSS3 + JavaScript</i>	38
2.7 <i>Archivos SVG que es y su implementación en HTML5</i>	38
2.8 <i>Lenguaje Unificado de Modelado</i>	40

2.8.1	Diagrama de Casos de Uso	41
2.8.2	Diagramas de Secuencia.	41
2.8.3	Diagramas de Colaboración.....	42
2.8.4	Diagramas de Estado.....	42
2.8.5	Diagramas de Clases.	43
2.8.6	Diagramas de Objetos.	43
2.8.7	Diagramas de Componentes.	44
2.9	<i>Proceso Unificado Racional</i>	45
2.9.1	Concepción.....	45
2.9.2	Elaboración.....	45
2.9.3	Construcción	45
2.9.4	Transición.	46
2.10	<i>PHP& MySQL como herramienta de desarrollo</i>	48
2.10.1	Breve Historia de PHP	48
2.10.2	Lenguaje de programación interpretado PHP	49
Capítulo 3	50
Metodología	50
3.1	<i>Fase de planeación:</i>	50
3.2	<i>Fase de elaboración:</i>	51
3.3	<i>Fase de desarrollo:</i>	52
3.4	<i>Fase de Pruebas/Implementación:</i>	53
Capítulo 4	54
Desarrollo	54
4.1	<i>Planeación y Análisis</i>	54
4.1.1	Casos de Uso.....	59
4.1.2	Diagrama de Clases.	65
4.2	<i>Diseño y Desarrollo</i>	66
4.2.1	Descripción y desarrollo de SIMA.....	66
4.2.2	Diseño sistema Multimedia	73
4.2.3	Desarrollo de Mapa Multimedia SIMA.....	77
4.2.3.1	Desarrollo del mapa multimedia en capas.	78
4.2.3.2	Elementos gráficos y diseño del mapa multimedia.....	81
4.2.3.3	Contenedor de información de los productos.	89
Capítulo 5	92
Resultados	92
5.1	<i>Implementación del sistema de captura</i>	92

1.1	<i>Integración y pantallas de mapa multimedia</i>	94
Capítulo 6	98
Conclusiones	98
Bibliografía	101
Capítulo 7	104
Anexos	104
7.1	<i>Anexo 1 - Acerca de SIMA</i>	104
7.2	<i>Anexo 2 - Reglas de inocuidad</i>	105
7.3	<i>Anexo 3 - Tablas de descripción de casos de uso</i>	106
7.4	<i>Anexo 4 - Script de base de datos SIMA</i>	161
7.5	<i>Anexo 5 Código fuente sistema de captura</i>	175
Glosario de términos	234

Índice de Tablas

Tabla 1 Soporte de CSS en los navegadores.	36
Tabla 2 Ejemplo de ficha técnica	57
Tabla 3 Universo de productos agropecuarios y pesqueros más representativos.....	58
Tabla 4 Descripción de caso de uso “Inicio de Sesión”	60
Tabla 5 Descripción de caso de uso “Cerrar Sesión”	61
Tabla 6 Descripción de caso de uso “Menú Productos”	63
Tabla 7 Descripción de caso de uso “Menú Producción”	64
Tabla 8 Descripción de caso de uso “Menú Catálogos”	65
Tabla 9 Productos más representativos de B.C.	67
Tabla 10 Tabla de diseños finales de productos agropecuarios y pesqueros.	83

Índice de Figuras

Figura 1 Ejemplo de diagrama de caso de uso.	41
Figura 2 Ejemplo de diagrama de secuencia.	42
Figura 3 Ejemplo de diagrama de clases.	43
Figura 4 Fases del modelo RUP.	46
Figura 5 Modelo entidad relación del sistema de administración de Base de Datos.	59
Figura 6 Diagrama de caso de uso “Inicio Sesión”.	60
Figura 7 Diagrama de caso de uso “Cerrar Sesión”.	61
Figura 8 Diagrama de caso de uso “Menú”.	62
Figura 9 Diagrama de caso de uso “Menú Productos”.	62
Figura 10 Diagrama de caso de uso Menú Producción.	63
Figura 11 Diagrama de caso de uso “Menú Catálogos”.	64
Figura 12 Diagrama de clases para sistema de captura.	66
Figura 13 Pantalla de iniciar sesión en el sistema SIMA.	68
Figura 14 Menús del sistema SIMA.	68
Figura 15 Pantalla de administración de productos en el sistema SIMA.	69
Figura 16 Pantalla localidades del producto.	69
Figura 17 Pantalla Atributos del Producto.	70
Figura 18 Pantalla Nutrientes del Producto.	70
Figura 19 Pantalla de Catálogos del Producto.	71
Figura 20 Pantalla Nutrientes del Producto.	72
Figura 21 Pantalla de Ficha Técnica.	72
Figura 22 Pantalla de Ficha técnica con indicadores de producción.	73
Figura 23 Primer prototipo Mapa Interactivo.	74
Figura 24 Prototipo de interacción básica con el usuario.	75
Figura 25 Prototipo pantalla de bienvenida a localidad.	76
Figura 26 Prototipo pantalla de presentación del producto.	76
Figura 27 Prototipo pantalla detalle del producto.	77
Figura 28 Representativa de JavaScript.	78
Figura 29 Representativa de SVG.	78
Figura 30 Esquema de Capas para los códigos SVG.	79
Figura 31 Fondo Mapa SIMA.	79
Figura 32 Icono de “Acerca De”.	79
Figura 33 Icono de Inocuidad.	79
Figura 34 Gráficos estáticos.	80
Figura 35 Gráficos delimitadores (Municipios).	80
Figura 36 Gráficos con eventos (Valles).	80
Figura 37 Prototipo pantalla detalle del producto.	81
Figura 38 Gráficos Valle de Mexicali.	83
Figura 39 Gráficos Valle Punta Colonet.	84

Figura 40 Gráficos Valle Tecate.	84
Figura 41 Gráficos Valle de las Palmas.	85
Figura 42 Gráficos Valle de las Palmas.	85
Figura 43 Gráficos Valle de Maneadero.	86
Figura 44 Gráficos Valle de Maneadero.	86
Figura 45 Gráficos Valle de Rosarito.	87
Figura 46 Gráficos Valle de la Trinidad.	88
Figura 47 Gráficos Valle Marino.	88
Figura 48 Jerarquía de contenido en la información de los productos.	89
Figura 49 Representación archivo contenido Productos.	89
Figura 50 Gráficos Contenedor Productos y Detalle de Productos.	90
Figura 51 Gráficos Contenedor indicadores de nutrición de productos.	91
Figura 52 Infraestructura de la implementación en FCA.	92
Figura 53 Respaldo de BD SIMA.	93
Figura 54 Infraestructura de la implementación en FEyRI.	93
Figura 55 Pantalla Inicial Mapa Multimedia.	94
Figura 56 Pantalla “Acerca de SIMA”.	94
Figura 57 Pantalla “Reglas de Inocuidad”.	95
Figura 58 Pantalla selección de valles.	95
Figura 59 Pantalla listado de productos por valle.	96
Figura 60 Pantalla detalle del producto.	97
Figura 61 Pantalla indicadores de nutrición del producto.	97

Capítulo 1

Introducción

A pesar de que las frutas y hortalizas constituyen alimentos básicos, no es sino hasta hace poco que el análisis de su consumo se convierte en un factor central de las políticas económicas, sociales y sanitarias en México.

En México el consumo de frutas, hortalizas y pescados es apenas el 50% del consumo de otros países con menor producción o disponibilidad (Avendaño, 2010). Una posible razón de lo anterior es la falta de conocimiento por parte de los consumidores de la oferta regional, sus propiedades y precios, hasta la influencia en los hábitos de consumo derivados de la cercanía geográfica a los Estados Unidos de América (EUA).

A nivel internacional, sobre todo en los países desarrollados, la industria y las organizaciones gubernamentales y no gubernamentales colaboran para encontrar la forma de incrementar el consumo como parte de las estrategias de transformar el estilo de vida de la población y combatir la obesidad por la ingesta de alimentos de alto contenido calórico y las enfermedades crónicas que se relacionan (Avendaño, 2010).

Se propone un proyecto que busque generar un medio multimedia de difusión pública enfocado al sector educativo que sirva de estrategia para fortalecer la producción agropecuaria de Baja California, a través de la promoción de su consumo, bajo un concepto de divulgación de la actividad regional producción-consumo, a través de un sistema de información web.

Hoy en día la población se encuentra frente a la educación digital apoyada fuertemente en los medios electrónicos y cibernéticos. Los niños de hoy ya no utilizan tanto el papel y el lápiz sino cada vez más el mouse y la pantalla táctil. Para las tareas y entretenimiento ya no utilizan solamente el libro en papel sino toda una inmensa biblioteca virtual a la largo de

Internet, el cual les permite descargar y consultar información en varios formatos.

Este trabajo integra además de información que muestra la realidad en lo relativo al uso de sistemas de tipo multimedia y web, un marco teórico con la conceptualización base para el entendimiento de este trabajo y la descripción de los procesos realizados en el análisis, diseño e implementación del sistema desarrollado.

A continuación se describen los capítulos en los cuales está organizado el presente trabajo: El primer capítulo contempla una breve introducción del trabajo de tesis, así como el establecimiento de la problemática, justificación y los objetivos.

El segundo capítulo presenta un marco teórico para poner en contexto sobre el reto tecnológico, las tecnologías utilizadas actualmente para el desarrollo de los sistemas multimedia, así como las ventajas de utilizar los sistemas multimedia como herramienta de apoyo en la educación.

En un tercer capítulo se establece la metodología que se seguirá durante el desarrollo de los objetivos establecidos en el trabajo de tesis, dicho desarrollo se presenta durante el capítulo cuarto. En el cuarto capítulo nos lleva de la mano del como fue desarrollado el sistema multimedia, pasando por el proceso de planeación, elaboración y desarrollo.

En los capítulos quinto y sexto se establecen los resultados del desarrollo, así como las conclusiones obtenidas por la experiencia de haber desarrollado el presente trabajo de tesis.

1.1 Antecedentes

1.1.1 El Sector Agropecuario en México

En México el organismo encargado de fomentar y organizar la producción agropecuaria en el país, es la Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación (**SAGARPA**), es una dependencia del poder ejecutivo federal, que busca promover el desarrollo integral del campo y de la pesca del país que permita el aprovechamiento sustentable de sus recursos, el crecimiento sostenido y equilibrado de las regiones, la generación de empleos atractivos que propicien el arraigo en el medio rural y el fortalecimiento de la productividad y competitividad de los productos para consolidar el posicionamiento y la conquista de nuevos mercados, atendiendo a los requerimientos y exigencias de los consumidores (SAGARPA, 2010).

El presente trabajo de tesis nace como una vertiente del proyecto de investigación nombrado *“Producción y consumo del sector agropecuario de Baja California. Estrategias para su fortalecimiento y vinculación”*, el cual es una iniciativa que el cuerpo académico: “Desarrollo y competitividad agroalimentaria” ha desarrollado, en colaboración con el cuerpo académico: “Sistemas de Información y Gestión empresarial”, ambos de la UABC. Dicho proyecto participó en la 15va. Convocatoria Interna de apoyo a proyectos de investigación de la UABC, obteniendo el apoyo, cabe mencionar que este proyecto es un esfuerzo de La Universidad Autónoma de Baja California en conjunto con la Universidad de Murcia, España y el Consejo Agrícola de Baja California, A.C.

1.1.2 Consumo de frutas y hortalizas

A nivel internacional, sobre todo en los países desarrollados, la industria y las organizaciones gubernamentales y no gubernamentales colaboran para encontrar la forma de incrementar el consumo como parte de las estrategias de transformar el estilo de vida de la población y combatir la obesidad por la ingesta de alimentos de alto contenido calórico y las enfermedades crónicas que se relacionan. Un desafío para los países menos desarrollados lo constituye el suministro de alimentos saludables a su población, donde

prevalezcan los productos de alta calidad, inocuos y de valor nutritivo.

Según estudios que analizan la ingesta de frutas y hortalizas, sus resultados señalan los factores psicosociales claves que inciden en su consumo, como el conocimiento de la ingesta recomendada, las enfermedades relacionadas con la dieta y la conciencia acerca de la salud (OMS, 2003). Además, se encuentran en la lista de determinantes el precio, la textura, el gusto, la apariencia, la calidad, las dificultades de almacenamiento, la falta de disponibilidad en los comercios, la comodidad, la seguridad y los hábitos de la infancia entre otros determinantes. El grado u orden de importancia de los determinantes anteriores depende del grupo de estudio o segmento del mercado analizado con características sociodemográficas definidas.

Según expertos en alimentación, nutrición y prevención de enfermedades crónicas de la FAO el gran desafío para la salud pública es incrementar la ingesta de frutas y hortalizas, ya que la mayoría de la población mundial sigue sin consumir cantidades suficientes de estos alimentos (FAO, 2003).

El Informe Mundial de la Salud, elaborado por la OMS, apunta expresamente: "Las frutas y las hortalizas son importantes componentes de una dieta saludable, que pueden prevenir enfermedades cardiovasculares y ciertos tipos de cáncer, principalmente del sistema digestivo. Su ingesta varía considerablemente entre países, en virtud de aspectos vinculados con la economía, la cultura y la producción agropecuaria".

5 al día México, es un programa liderado en México por la Fundación Campo, Educación y Salud A.C. y respaldan la iniciativa internacional de consumir 5 raciones de frutas y verduras al día, a fin de cubrir los 400 gramos recomendados por la OMS (Organización Mundial de la Salud).

Los objetivos en México de esta fundación civil son: el apoyar el fortalecimiento, promoción y mejoramiento de los hábitos alimenticios de las familias y de los grupos en estado de vulnerabilidad; incrementar el consumo de frutas y verduras para fomentar la

sana alimentación entre las familias mexicanas; procurar la participación del gobierno, iniciativa privada y organizaciones civiles en la implementación del programa 5xdía; y apoyar a los productores y comercializadores mexicanos. Este tipo de campañas, están implementándose en más de 40 países del mundo, hoy en día.

En materia de las preferencias del consumidor en frutas y hortalizas los consumidores hacen una elección encarando varias alternativas, cuando éstos eligen, responden a un conjunto de incentivos que resultan de sus percepciones acerca de sus beneficios y costos asociados a cada una de las alternativas disponibles. Las preferencias de los consumidores pueden ser determinadas por diferentes factores culturales, sociales e incluso geográficos.

Los factores que determinan las preferencias de un consumidor por un bien o servicio pueden ser externos como la cultura, que representa características tales como: los conocimientos, los valores, el arte, la moral, las leyes, las costumbres y los hábitos adquiridos en una sociedad; otro de los factores externos importantes es el nivel socioeconómico, el cual no está determinado únicamente por el nivel de ingreso, interviniendo en éste otras características, tales como: la educación, profesión, lugar de residencia, grupo social y amistades (Arroyo Cossío, 2011).

Otro tema que es necesario analizar en este trabajo, es el de multimedia, la cual es utilizada en el desarrollo del mismo, a continuación se presenta terminología considerada importante en este trabajo.

1.1.3 Introducción a la multimedia.

A pesar de que la enseñanza a distancia lleva más de 100 años, especialmente en Gran Bretaña y Canadá, en América Latina se comienzan a plantear las necesidades de distribuir educación asistida por medios tecnológicos, para llegar a aquellas comunidades alejadas de las grandes ciudades, para capacitar a cientos de personas y disminuir costos (Estrada, 2010).

En la actualidad existen incontables casos donde la educación es asistida por los medios multimedia y gracias a su éxito en el contexto de la autonomía educativa, ha permitido cuestionar qué tanto pueden sustituir la presencia de un profesor, y en general, un aula de clases. Sin embargo, es posible combinar medios (Multimedia), no para sustituir al profesor, sino para enriquecer la enseñanza y el aprendizaje. La educación actual plantea al alumno como el centro de los procesos educativos.

El profesor ha modificado su función para pasar a ser un facilitador, un asesor, un guía. En este contexto, la responsabilidad total del aprendizaje es del alumno. La multimedia cumple la función de hacer llegar al alumno, la información necesaria para su aprendizaje, presentándola de manera atractiva para el alumno y con el propósito de obtener un aprendizaje significativo. De alguna manera, las fuentes de información que antes atesoraba la escuela están ahora dispersas. El maestro ya no lo sabe todo, al contrario, puede que sobre ciertos temas sepan más los jóvenes, por ejemplo, sobre el manejo de nuevas tecnologías.

Hoy los niños cuentan con una variada cantidad de imágenes de televisión y este hecho ha mudado fundamentalmente la piel de nuestro cerebro. La televisión coloca imágenes, es decir percepciones del mundo en el cerebro de niños que aún no están escolarizados y ha transformado su experiencia cotidiana, porque cuando un niño tiene físicamente frente a él, un animal o un barco, conoce una imagen y no un objeto. Esto es, el recuerda una imagen que la televisión ha instalado en su cerebro y la evalúa, critica o ajusta a partir de la experiencia real de esa imagen, que es justamente lo contrario de lo que se hacía antes; cuando la primera percepción de los acontecimientos normalmente eran los hechos, después la imagen era el ayuda a la memoria, la representación o la conservación de algunas de las percepciones que se habían tenido en la realidad (Pérez, 2002).

El futuro de los materiales didácticos para la enseñanza tiene un carácter hipermedia, es decir, estos materiales tienen la posibilidad de presentar un contenido didáctico mediante la conexión de diversos textos con distintos signos sonoros, imágenes, etc. (Pérez, 2002).

De una civilización que ha sido dominada por el texto, hemos migrado a una civilización hipermedia con el surgimiento de una nueva escritura que puede utilizar las imágenes por primera vez en la historia. Con la computadora hemos descubierto la escritura de la imagen, de esta manera, la imagen será uno de los instrumentos más poderosos de escritura del futuro (Pérez, 2002).

Bajo este contexto tecnológico, ¿En dónde van a quedar los libros de texto? Por el bajo costo que cada día adquiere la tecnología, llegará el momento en que será igual dar a un niño un libro de texto, que darle una computadora para trabajar; porque estamos ante una sociedad que va a aligerar los costos de la tecnología digital.

1.1.4 Multimedia como herramienta de apoyo al aprendizaje

Aunque el término multimedia no es nuevo en el campo educativo, lo parece por haber adquirido ciertas connotaciones en el campo de los iniciados de las nuevas tecnologías de la información, que ha hecho que los profesionales de la educación tengan la sensación de encontrarse ante algo totalmente nuevo.

En la actualidad multimedia puede significar muchas cosas, esto depende del contexto en el que se analice y del tipo de especialista que lo defina. En un tiempo multimedia se refería por lo general a presentaciones de diapositivas con audio, también ha designado a aquellos materiales incluidos en kits o paquetes didácticos; etc. Sin embargo, multimedia de hoy suele significar la integración de dos o más medios de comunicación que pueden ser controlados o manipulados por el usuario vía ordenador (Bartolomé, 1994). En rigor, el término multimedia es redundante, ya que 'media' es en sí un plural, por ello hay autores que prefieren utilizar el término hipermedia en vez de multimedia (Salinas, 1996).

Multimedia se refiere normalmente a vídeo fijo o en movimiento, texto, gráficos, audio y animación controlada por una computadora. Pero esta integración no es sencilla. Es la combinación de hardware, software y tecnologías de almacenamiento incorporadas para

proporcionar un entorno multi-sensorial de información.

Por su propia naturaleza, el ámbito de los multimedia no se ha asentado todavía y se producen continuas discusiones y reflexiones. Estas pueden girar en torno a las diferentes tecnologías necesitadas para crear, almacenar y ofrecer las presentaciones.

Multimedia será efectivo en el contexto de la docencia en la medida en que comprometa activamente al estudiante en un proceso comunicativo en forma de diálogo. Los sistemas multimedia, aun en los sistemas más sencillos, incorporan y mejoran aquellas características didácticas que reúnen los medios que lo integran, especialmente el texto, el vídeo y el ordenador como medios didácticos. Mantiene la posibilidad de manipulación y el manejo sencillo de los aparatos, pero, sobre todo, desarrolla al máximo la posibilidad de retroalimentación inmediata (Salinas, 1996).

Desde el punto de vista cognoscitivo, está comprobado que el cerebro trabaja por asociación, por lo que es posible presentar la información de una forma asociativa con el uso de multimedia, esto lograría que multimedia se convierta en una interfaz para uso en la educación.

Otro aspecto importante a considerar con respecto a multimedia, ocurre cuando se tienen grupos muy grandes la única manera de llegar a los alumnos de una forma individual es utilizar software educativo de multimedia.

Según Gayeski se han hecho algunos estudios para conocer la manera en que la multimedia puede ser utilizada para cumplir con ciertas necesidades de estudiantes. Los estilos de aprendizaje de los estudiantes fueron clasificados de acuerdo a las cuatro dimensiones de los estilos de aprendizaje de Soloman's. Los cuales son:

- Procesamiento (activo/reflexivo)

Los estudiantes de este grupo preferían un ambiente de aprendizaje donde ellos

participarán activamente. Aquí el descubrimiento se podría lograr con el uso adecuado de multimedia. El uso de esta tecnología involucra al alumno activamente en el aprendizaje y expone a los estudiantes a conceptos de forma innovadora que los métodos tradicionales de enseñanza no lo permiten (Gayeski, 1992).

Cabe aquí señalar, que el tipo de aprendizaje que se ha logrado, es el aprendizaje significativo que requiere una actitud u orientación más activa con respecto al propio aprendizaje, en la que el alumno ha de tener más autonomía en la definición de sus objetivos, sus actividades y sus fines, para lograr con esto la adquisición de conceptos.

- Percepción (sensitivo/intuitivo)

La multimedia incluye software de simulación interactivo de sistemas reales con lo que se llenan las necesidades del grupo de aprendizaje sensitivo al cual le interesa tener experiencias con los datos reales.

- Captación (visual/verbal)

El software de Multimedia favorece a los aprendizajes visuales ya que se presentan imágenes, gráficas, animación, películas, esquemas y otros objetos visuales.

- Comprensión (secuencial/global)

Se ha comprobado que multimedia y el software educativo, pueden abarcar un campo amplio para llenar los espacios creados por la dicotomía de los estilos de aprendizaje y enseñanza (Gayeski, 1992).

Según Gayeski los estilos de aprendizaje de los alumnos basado en un estudio responden a los siguientes resultados:

- 67% de los estudiantes aprenden mejor en forma activa, a pesar de que la cátedra es típicamente pasiva.

- 57% de los estudiantes son sensitivos pero se les enseño de manera intuitiva.
- 69% de los estudiantes son visuales, aún así la cátedra es básicamente verbal.
- 28% de los estudiantes son de la dimensión global, y pocas veces enfocamos la cátedra en un ámbito general.

Otro tema que es necesario conocer para ubicarnos en el contexto en el cual es desarrollado el presente sistema es lo relativo al sistema de información geográfica del sector agropecuario de B.C., que a continuación se detalla.

1.1.5 Sistema de Información Geográfica del Sector Agropecuario de B.C.

OEIDRUS (Oficina Estatal de Información para el Desarrollo Rural Sustentable) tiene como objetivo proveer información estadística agropecuaria, forestal y pesca confiable a los productores, consumidores y público en general, integrándola y distribuyéndola en medios electrónicos e impresos, con esto se busca apoyar el fortalecimiento del Desarrollo Rural Sustentable del Estado (*OEIDRUS*, 2010).

OEIDRUS en su portal en internet cuenta con tablas de información estadística actual en materia agropecuaria, cuenta también con un sistema de información geográfico denominado *SigaBC* (Sistema de Información Geográfica del Sector Agropecuario de Baja California).

SigaBC, es un instrumento de consulta y análisis diseñado con el objeto de presentar datos geográficos digitales organizados en capas sobre distintos aspectos relacionados con la agricultura, ganadería, pesca, clima y recursos hidráulicos del estado de Baja California. Esta aplicación web fue diseñada en plataforma *MapServer CGI 5.6.3* y *P.Mapper 3.2.0*, y permite consultar información en un contexto espacial, desde diferentes niveles de acercamiento, sobreponer información de diferentes capas, incluir puntos de referencia, generar reportes, construir y descargar mapas a partir de requerimientos específicos del usuario (*OEIDRUS*, 2010).

El Sistema multimedia *SigaBC* actualmente cuenta con algunas capas correspondientes a actividades y/o acontecimientos de la región de Baja California:

- ***Daños por terremotos.***
- ***Información agrícola:*** Palma datilera en valle de Mexicali, superficie de Invernaderos y Malla-Sombra en el estado, centros de acopio de trigo en el Valle de Mexicali.
- ***Información pecuaria:*** Corrales de engorda y establos lecheros en el valle de Mexicali, rastros y apiarios en Baja California.
- ***Información de pesca:*** Localidades de arribo.
- ***Información hidro-agrícola:*** Red mayor, canales secundarios en el Valle de Mexicali, cuerpos de agua y ríos en el estado.
- ***Uso de Suelo y Vegetación:*** área agrícola, bosques, Matorral, área sin vegetación, otro tipo de vegetación, pastizal y topónimos.
- ***Zonas agropecuarias:*** Módulos de riego y parcelas en el valle de Mexicali, propiedad social, CADERS.
- ***Puntos de Inspección:*** Fitozoosanitarias.
- ***Información General:*** Delimitación estatal, zonas urbanas, vías de comunicación, curvas de Nivel.

Una vez analizados los antecedentes de temas involucrados en la presente investigación podemos plantear el problema de manera clara y concreta.

1.2 Planteamiento del problema

El bajo consumo de productos agropecuarios de la población mexicana contrasta con la oferta aparente de frutas y verduras, de acuerdo a esto, la demanda y la oferta se encuentran desvinculadas, esto ocasiona limitantes en el desarrollo del sector agropecuario. En México la demanda de frutas, hortalizas y pescados es apenas el 50% del consumo de otros países con menor producción o disponibilidad. Este consumo se encuentra por debajo de las recomendaciones de la FAO, por sus siglas en inglés, La Organización de las Naciones Unidas para la Agricultura y la Alimentación (FAO, 2003).

El consumo de pescado se encuentra en una situación precaria ya que el promedio nacional anual de 10 Kg. está por debajo del mundial de 16 Kg. Per cápita (FAO, 2003), y esto a pesar de que varias pesquerías de México y de Baja California gozan de reconocimiento internacional, como el atún, camarón, langosta entre otros.

Lo anterior es un reflejo de la falta de conocimiento por parte de los consumidores de la oferta regional, sus características, precios y puntos de venta, hasta la influencia en los hábitos de consumo derivados de la cercanía geográfica a los EUA.

Baja California cuenta con 880 Km. de litorales en el lado del Océano Pacífico y 675 Km. en el lado del Golfo de California, lo que representa el 13.41% del total de litorales del país, a pesar de la disponibilidad y del bajo precio de algunas especies marinas, la población mexicana consume más productos cárnicos (83%) que pescados y mariscos (22%) (PROFECO, 2010).

1.3 Justificación

A pesar de que en México se han desarrollado programas para fomentar la ingesta de frutas y hortalizas como *5 AL DÍA (México)*. No es suficiente para dejar el tema de la educación nutricional de lado, ejemplo de esto es que aunque en las escuelas públicas la extinción de alimentos “chatarra” es hoy en día una realidad, los niños aun tienen acceso a la venta externa, ya sea en alguna tienda de abarrotes, mercado o cualquier lugar fuera de la escuela. Se debe hacer conciencia en ellos que más que una imposición debe ser una decisión el comprar una fruta o bien cualquier producto con valor nutrimental adecuado.

Un desafío para los países como México lo constituye el suministro de alimentos saludables a su población, donde prevalezcan los productos de alta calidad, inocuos y de valor nutritivo. El balance de costo/beneficio de consumir productos agropecuarios en fresco influye negativamente en la decisión de consumo de estos productos, a pesar de que en promedio estos son de bajo precio en relación a la cantidad de nutrientes que proporcionan.

Por todo se propone una estrategia que busque captar la atención de aquellos usuarios que estén en el nivel de educación básica, mediante interfaces multimedia amigables y atractivas visualmente. Se pretende desarrollar un sistema que reúna las características antes mencionadas y que además informe a los usuarios sobre los productos agropecuarios y pesqueros que la región ofrece, así como su valor nutrimental.

Mediante la estrategia propuesta se espera lograr un sistema que permita la difusión y conocimiento de la oferta de productos agropecuarios y pesqueros en el estado de Baja California.

1.4 Objetivos

1.4.1 Objetivo general

Diseñar un sistema de información con un mapa interactivo que muestre la oferta de productos agropecuarios y pesqueros que se producen en el estado de Baja California, así como la información nutrimental e inocuidad de los mismos.

1.4.2 Objetivos específicos

- Diseñar y desarrollar una base de datos para el control y centralización de la información de los productos agropecuarios y pesqueros del estado de Baja California.
- Diseñar y desarrollar sistema básico de administración de la información general, de nutrición e indicadores económicos de los productos agropecuarios y pesqueros del estado de Baja California.
- Presentar información general y de nutrición de los productos agropecuarios y pesqueros que se producen en el estado de Baja California en un formato multimedia.

Capítulo 2

Marco teórico

2.1 Introducción a la multimedia

Multimedia es el término utilizado para referirse a cualquier objeto o sistema que utiliza múltiples medios de expresión (físicos o digitales) para presentar o comunicar información. De allí la expresión "multi-medios". Los medios pueden ser variados, desde texto e imágenes, hasta animación, sonido, video, etc. También se puede calificar como multimedia a los medios electrónicos (u otros medios) que permiten almacenar y presentar contenido multimedia. Multimedia es similar al empleo tradicional de medios mixtos en las artes plásticas, pero con un alcance más amplio.

A principios de los noventa era común hablar de adquirir una computadora multimedia, esto ya que no todos los equipos de cómputo cumplían con los requerimientos técnicos para interactuar con la información que en ese momento se adelantaba a su época (imágenes, video y audio). Hoy en día ya no usamos más el término *computadora multimedia* para referirnos a las computadoras personales, no tiene sentido, ya que todos los equipos fabricados actualmente incorporan todo lo necesario para permitir la multimedia; de hecho son requerimientos básicos para instalar los actuales sistemas operativos.

El beneficio más importante de la multimedia es que permite enriquecer la experiencia del usuario-receptor, logrando una asimilación más fácil y rápida de la información presentada. Esto es bastante claro en las aplicaciones de tipo formativa o educacional.

Actualmente en prácticamente todas las empresas y organizaciones importantes emplean el "*e-learning*" o "*computer based training CBT*" (instrucción asistida por computadora) para capacitar a sus empleados. No sólo se reducen costos, sino que además le permiten avanzar al alumno a su propio ritmo, repitiendo y enfatizando aquellas lecciones más difíciles. Esto permite también la educación a distancia, desde una computadora con acceso

a Internet; un ejemplo son las líneas aéreas, que capacitan a sus pilotos desde sus distintas bases alrededor del mundo.

2.1.1 Multimedia en los procesos de enseñanza-aprendizaje

El término multimedia resulta ya familiar y es frecuente leer cosas sobre las posibilidades que ofrece en los procesos de enseñanza-aprendizaje. Multimedia suele presentarse como el último avance que, propiciado por la evolución y expansión de los medios electrónicos viene a resolver algunos de los problemas que tiene planteada la enseñanza. Sin embargo, el término no resulta nuevo para las ciencias de la educación: el convencimiento de la importancia de la comunicación multi-sensorial en el proceso didáctico, el principio didáctico de la redundancia y la reflexión que ha acompañado a cada aparición de un nuevo medio, han hecho que si no el término, al menos el concepto sea usual en Tecnología Educativa (Salinas, 1996).

Desde hace algunos años, el acelerado desarrollo de las tecnologías de la información y de la comunicación ha hecho posible y accesible al ser humano nuevas formas de aprendizaje. Aunque las estructuras del mundo de la educación son aún resistentes y conservadoras, están a punto de agrietarse, y un nuevo paradigma, un modelo explota el potencial, los recursos y las posibilidades de las Nuevas Tecnologías de la Información y de la Comunicación (NTIC), de suerte que el campo de la tecnología educativa cobra mayor relieve día con día (Salinas, 1996).

2.1.2 ¿Qué se entiende hoy por multimedia?

En la actualidad multimedia puede significar muchas cosas, depende del contexto en que nos encontremos y del tipo de especialista que lo defina. En un tiempo multimedia se refería por lo general a presentaciones de diapositivas con audio, también ha designado a aquellos materiales incluidos en kits o paquetes didácticos; etc.

Multimedia se refiere normalmente a vídeo fijo o en movimiento, texto, gráficos, audio y animación controlados por un ordenador. Pero esta integración no es sencilla. Es la

combinación de hardware, software y tecnologías de almacenamiento incorporadas para proporcionar un entorno multi-sensorial de información.

Según Salinas por su propia naturaleza, el ámbito de los multimedia no se ha asentado todavía y se producen continuas discusiones y reflexiones. Estas pueden girar en torno a las diferentes tecnologías necesitadas para crear, almacenar y ofrecer las presentaciones.

La fuerte aceptación y desarrollos que experimenta actualmente la multimedia es fruto de los avances tecnológicos en:

- Software de desarrollo de aplicaciones multimedia. Fundamentalmente descubrimiento y desarrollo de los sistemas de hipertexto y de hipermedia, y la aparición de sistemas de autor interactivo, así como algoritmos de compresión.
- El hardware de desarrollo. Fundamentalmente ligado al tema del almacenamiento: la llegada de los discos ópticos con grandes capacidades de almacenamiento de grandes cantidades de datos ordenados, así como imágenes de video y audio, ha sido crítica para el desarrollo multimedia.
- Dispositivos periféricos multimedia. Amplían el rango de usuarios, al hacer más fácil la interacción entre usuario y ordenador.

La mayor parte de aportaciones en el terreno de los multimedia suele referirse, sin embargo, a los dos últimos aspectos (discusión sobre las tecnologías digitales, el almacenamiento, la velocidad de respuesta, las formas de presentación, etc.).

2.1.3 Diseño de sistemas multimedia

En un sistema multimedia como un material didáctico de carácter modular en el que lo fundamental son las conexiones y posibles combinaciones de los distintos medios. La información de estos viene integrada para poder ser utilizada en situaciones de aprendizaje diversas, de acuerdo con decisiones del usuario (decisiones en relación a si se hará el aprendizaje, al cómo, al cuanto, al dónde, etc.), e integra la suficiente orientación para lograr los objetivos marcados de acuerdo con estas decisiones, de forma que la secuenciación y presentación depende de estas decisiones o de las respuestas del usuario al material (Bustamante, 2010).

Como consecuencia de este carácter modular también la misma lección o unidad tiende a desaparecer en los materiales transformándose en módulos. Estos módulos pueden utilizarse tanto en secuencias lineales, como transversalmente o en espiral. En parte la modularidad y la flexibilidad de las presentaciones están condicionadas por la concepción de la comunicación, y son, también en parte, consecuencia de elecciones pedagógicas, particularmente en el énfasis que ponemos en el alumno (Bustamante, 2010).

Desde esta perspectiva, lo verdaderamente importante de los sistemas multimedia es que se adapten a los principios de diseño de medios interactivos, que integren un interface usuario-material adecuado a la situación de aprendizaje. Y ello se logra más que con la sofisticación tecnológica, con un cuidado diseño didáctico del material.

Es quizás en el terreno del diseño, donde pueden darse las aportaciones que logren aplicaciones de estos sistemas al aprendizaje cada día más efectivas. Aportaciones que pueden ser menos espectaculares que la duplicación de la capacidad de almacenamiento y la de reducir el lapso de tiempo de espera, pero que pueden ser más productivas en la aplicación a situaciones reales de los sistemas.

En un contexto donde los sistemas multimedia interactivos, la secuencia y selección de mensajes se determinan por la respuesta del usuario al material, por la intervención de éste en la secuenciación del aprendizaje, es fundamental abordar el tema del diseño de sistemas multimedia desde el concepto de medio interactivo o enseñanza interactiva.

Es en el momento del diseño del programa cuando se determina si va a ser interactivo o no, o el grado de interacción (interactividad) con el alumno que va a presentar, ya que es en esta fase donde se determina la estructura y secuencia del programa, el control del usuario sobre el mismo, la personalización o estandarización del contenido, etc.

Estas reflexiones sobre el diseño de medios interactivos, la concepción de sistemas multimedia que ya hemos presentado y las exigencias pedagógicas, junto a los avances en

las tecnologías de la información logrados en los últimos años nos han llevado a la búsqueda de modelos más adecuados de diseño de materiales para la enseñanza.

La existencia de tecnologías interactivas a distancia (video-tex, tv cable), la irrupción de los satélites de difusión directa y las experiencias en el diseño de medios interactivos nos ha conducido a un proyecto en el que diversos medios se integran en paquetes didácticos multimedia de "aprendizaje abierto".

Los materiales didácticos para este tipo de aprendizaje tienen que formar verdaderos paquetes didácticos integrados por audio, vídeo, diapositivas, textos y software. Estos materiales deben ser diseñados para un doble uso: tanto los estudiantes presenciales, como aquellos que no pueden estar físicamente presentes, conseguirán el acceso al aprendizaje a través de una variedad de medios y con la posibilidad de clases tutoriales y entrevistas personales (Bustamante, 2010)

La versatilidad de este tipo de materiales, conduce a diseñar programas educativos en los que se contempla el uso de las instalaciones presenciales, la explotación de los sistemas de cable, ya sea televisión, teléfono, la televisión convencional o el satélite de difusión directa para explotar documentos que integran textos, gráficos, vídeo, audio, etc. Sin importar si esto se distribuye en un sólo medio o en varios. Lo importante es que todos estos medios vayan perfectamente integrados y diseñados para integrarse en distintos sistemas multimedia.

2.2 Internet como recurso educativo

Internet puede considerarse una de las características más significativas de nuestro tiempo. Los cambios son de tal magnitud, alcance y profundidad que han alterado los comportamientos individuales y las relaciones sociales, esto hace posible que los ciudadanos interactúen. Los alumnos con muy poco esfuerzo se han familiarizado con el uso de Internet. Los docentes tienen que ser conscientes que, muy pronto los alumnos no habrán conocido un mundo sin computadoras conectadas a Internet, habrán crecido en la era digital, serán la generación de la Red. Internet además de ser un

poderoso instrumento para el proceso de la información, facilitar el aprendizaje mediante el uso de materiales didácticos interactivos; añade la posibilidad de acceder a todo tipo de información y sobre todo, abre un canal comunicativo casi permanente alumno-profesor, alumno - alumno, profesor - profesor y con todo el mundo (Pacheco).

Con ello se abre un nuevo paradigma de la enseñanza, sin barreras espacio-temporales para el acceso a la información y para la comunicación interpersonal, esto nos ofrece múltiples posibilidades de innovación educativa en el marco de la enseñanza más personalizada y de un aprendizaje cooperativo acorde con los planteamientos socio constructivistas.

Internet cambia la forma de trabajar, hace que aparezcan nuevas formas: a distinto tiempo (de forma asíncrona) y en distinto lugar (a distancia). Hace posible que los alumnos y profesores distantes se comuniquen, o que alumnos presenciales puedan ser tutelados a distancia, en casa. También induce cambios en la metodología, esto favorece formas más colaborativas de trabajo (Pacheco).

En definitiva Internet es en sí misma es un recurso específico para la educación que crea entornos propios de aprendizaje, docencia y trabajo para alumnos y profesores. Asimismo la red está llena de informaciones y de instrumentos que pueden ser utilizados de forma original y con carácter propio y real. La Red, y sus servicios, establecen valores añadidos a la educación: Tutoría telemática, educación a distancia y metodologías propias (basadas en la comunicación distante y en la cooperación).

Internet es vehículo de comunicación para alumnos y profesores. Los instrumentos telemáticos están destinados a impactar fuertemente en los entornos educativos, en función de dos de sus características: favorecen el trabajo cooperativo, los contactos interpersonales, y eliminan barreras del espacio y tiempo.

El espacio del aula se abre a otros horizontes, los alumnos comparten sensaciones y experiencias de forma directa, de primera mano, y además en texto escrito, en la lengua propia o en un idioma extranjero. Estas actividades pueden ser completadas con

otras: Intercambio de textos escritos, imágenes, gráficos o datos, dirigidos por el profesor y orientado a la consecución de objetivos curriculares (Pacheco).

Otro efecto que se consigue con esta experiencia es el de que el alumno penetre en actividades del mundo real. Acceda a instrumentos, y opere con recursos, propios de la vida adulta y profesional, de la misma forma como puedan hacerlo un grupo de investigadores o de técnicos.

El papel del profesor es similar al que realiza en otros tipos de trabajos cooperativos: facilita la información necesaria, acompaña a los alumnos en la actividad, facilitándola y guiándoles, con el valor añadido de poder comunicar con otro colega suyo en un lugar y en un medio distante pero con unas inquietudes y motivaciones comunes.

2.3 Introducción a las versiones de HTML.

HTML es el lenguaje de marcación de hipertexto que se diseñó en 1989 para crear páginas web. Desde su creación ha ido evolucionando de acuerdo con diversas necesidades que se orientan sobre todo a mejorar el procesamiento de la información, y así fue como aparecieron varias revisiones (HTML 2, HTML 3.2, HTML 4, HTML 4.01) que ampliaron y depuraron este lenguaje (Alvarez, 2010).

Xhtml se creó para sustituir la sintaxis en la que se basa HTML: el SGML (Standard Generalized Markup Language), cuya finalidad principal era facilitar la creación manual de documentos. HTML era un lenguaje poco riguroso y, como consecuencia, los documentos resultaban un tanto caóticos y no siempre cumplían con la sintaxis. Esto provocó que los navegadores se hicieran más complejos para poder asumir ambigüedades y equivocaciones, es decir, para poder representar los documentos aunque tuvieran errores.

HTML Versión5 es la quinta revisión importante del lenguaje básico de la World Wide Web. HTMLVersión5 especifica dos variantes de sintaxis para HTML:

- HTML (text/html), la variante conocida como HTML
- XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML).

HTML 5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Algunos de ellos son técnicamente similares a las etiquetas <div> y , pero tienen un significado semántico, como por ejemplo <nav> (bloque de navegación del sitio web) y <footer>. Otros elementos proporcionan nuevas funcionalidades a través de una interfaz estandarizada, como los elementos <audio> y <video>. Mejoras en el elemento <canvas>, capaz de interpretar en algunos navegadores elementos 3D.

En esta versión no sólo define cómo se deben analizar los documentos, sino también cómo se deben interpretar si no son válidos o si están mal formados. Actualmente, los navegadores corrigen los errores de sintaxis de distinta manera, del modo en que a los fabricantes les resulta más práctico. HTML5 trata de poner fin a esa necesidad de ingeniería inversa de los navegadores, que compiten por definir cómo se deben subsanar los errores. El desarrollo de este código es regulado por el Consorcio W3C.

Dicha tecnología propone, pues, que sean los navegadores, y no los creadores de contenido, quienes faciliten la entrada y la validación de datos que tienen un patrón regular o están sometidos a restricciones. Así, el dolor de cabeza que supone para los diseñadores de páginas web el verificar el formato de direcciones electrónicas, intervalos de valores, términos de búsqueda, colores, fechas y horas, entre otros tipos de datos, se descarga ahora sobre los navegadores (Castro, 2007).

Una de las novedades principales de HTML5 es la inclusión del *Document Object Model* (DOM) como fundamento del lenguaje. El DOM describe la estructura de un documento de acuerdo con el paradigma de la orientación a objetos. En otras palabras, define el conjunto de entidades que están presentes en un documento HTML y las acciones que pueden realizarse sobre ellas (Alvarez, 2010).

Hasta ahora, el DOM siempre se había tratado de forma separada; cada navegador lo implementaba según la particular interpretación que su fabricante hacía de él. En cambio, en html5, el DOM forma parte del estándar. Así se garantiza que los navegadores interpretarán adecuadamente la sintaxis de HTML y que al mismo tiempo implementarán las funciones del DOM que la sustentan (Alvarez, 2010).

2.3.1 ¿Qué es y cómo nace Adobe Flash?

Adobe Flash debe sus raíces a una pequeña compañía llamada *FutureSplash* que fue adquirida por Macromedia en 1997. Macromedia acortó el nombre del producto a Flash, y contrató a Jonathan Gay el creador del sistema predecesor de flash como vicepresidente de tecnología. Esta aplicación era una mezcla de un editor de gráficos y de un editor de películas. Macromedia lanzó ocho versiones de Flash, cada una más avanzada que la anterior.

En 2005, el programa estaba equipado para reproducir música y video en vivo, y fue ampliamente utilizado para una variedad de aplicaciones de animación y gráficos. A finales de 2005, Adobe Systems adquirió Macromedia, incluyendo el programa.

En 2007, Adobe lanzó una nueva versión de Flash conocida como CS3, que comenzó a ser parte del paquete Adobe Creative Suite.

2.3.2 Contenido multimedia web sin Flash

Para incrustar contenido multimedia, HTML ya contaba con el elemento *object*, pero la nueva versión del estándar hace una propuesta más semántica. En el ámbito del multimedia, html5 incorpora directivas nuevas que actúan como contenedores de vídeo, gráficos vectoriales y audio.

Se espera que con los elementos *video* y *audio* los navegadores tengan la capacidad de presentar el contenido de forma nativa, es decir, sin requerir componentes externos como

Flash. Al fin y al cabo, una tarea sencilla como representar contenido multimedia no tiene por qué quedar en manos de un entorno cerrado y propietario (Allaire, 2010).

Este avance ha suscitado el debate sobre si Html5 acabaría con *Flash*. Es cierto que la propuesta de Html5 puede sustituir funciones que hasta ahora sólo eran posibles con *Flash*. Para reproducir multimedia no hace falta ningún mecanismo complejo, con que es razonable pensar que html5 desplazará a *Flash* como contenedor de vídeo y audio, pero no hay argumentos sólidos para pensar que *Flash* vaya a desaparecer (Allaire, 2010).

Al fin y al cabo, *Flash* no sólo sirve para mostrar contenido multimedia, sino que también tiene capacidad para resolver muchas otras tareas. El objetivo de la propuesta de html5 es que la reproducción de vídeo y audio sea más eficiente, consuma menos recursos y se pueda gestionar con un código abierto y transparente, y sin necesidad de instalar componentes adicionales (Allaire, 2010).

Todo apunta a que los navegadores incorporarán estas capacidades y que muchas páginas web se modificarán para aprovecharlas. Pero de momento, aunque incluidas en el estándar, esas funcionalidades se encuentran en la primera etapa de su adopción y no permiten controlar el contenido multimedia en la misma medida que *Flash* (Allaire, 2010).

De hecho, el popular depósito de vídeos *YouTube* ha declarado recientemente que HTML5 no cubre todas sus necesidades. Según Huggers son varios los problemas html5 entre los que destacan que aun no cuenta con un formato de vídeo estándar, no sustenta el visionado a pantalla completa ni permite escoger la calidad de reproducción. Por lo tanto, aunque las novedades de HTML5 son un avance hacia los estándares abiertos, es razonable pensar que *Flash* continuará desempeñando un papel vital en la distribución de contenido multimedia (Huggers, 2010).

2.4 Introducción a CSS

2.4.1 Breve Historia de CSS

Las hojas de estilos aparecieron poco después que el lenguaje de etiquetas SGML, alrededor del año 1970. Desde la creación de SGML, se observó la necesidad de definir un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos electrónicos (Eguíluz, 2010).

El gran impulso de los lenguajes de hojas de estilos se produjo con el boom de Internet y el crecimiento exponencial del lenguaje HTML para la creación de documentos electrónicos. La guerra de navegadores y la falta de un estándar para la definición de los estilos dificultaban la creación de documentos con la misma apariencia en diferentes navegadores (Eguíluz, 2010).

El organismo W3C (*World Wide Web Consortium*), encargado de crear todos los estándares relacionados con la web, propuso la creación de un lenguaje de hojas de estilos específico para el lenguaje HTML y se presentaron nueve propuestas. Las dos propuestas que se tuvieron en cuenta fueron la CHSS (*Cascading HTML Style Sheets*) y la SSP (*Stream-based Style Sheet Proposal*) (Eguíluz, 2010).

2.4.2 ¿Qué es CSS?

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la

accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes (Eguíluz, 2010).

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc. (Eguíluz, 2010).

2.4.3 Soporte de CSS en los Navegadores

El trabajo del diseñador web siempre está limitado por las posibilidades de los navegadores que utilizan los usuarios para acceder a sus páginas. Por este motivo es imprescindible conocer el soporte de CSS en cada uno de los navegadores más utilizados del mercado. Internamente los navegadores están divididos en varios componentes. La parte del navegador que se encarga de interpretar el código HTML y CSS para mostrar las páginas se denomina motor. Desde el punto de vista del diseñador CSS, la versión de un motor es mucho más importante que la versión del propio navegador.

La siguiente tabla muestra el soporte de CSS 1, CSS 2.1 y CSS 3 de los cinco navegadores más utilizados por los usuarios:

Navegador	Motor	CSS 1	CSS 2.1	CSS 3
Internet Explorer	Trident	Completo desde la versión 6.0	Completo desde la versión 8.0	Prácticamente nulo
Firefox	Gecko	Completo	Casi completo	Selectores, pseudo-clases y algunas propiedades
Safari	WebKit	Completo	Casi completo	Todos los selectores, pseudo-clases y muchas propiedades
Opera	Presto	Completo	Casi completo	Todos los selectores, pseudo-clases y muchas propiedades
Google Chrome	WebKit	Completo	Casi completo	Todos los selectores, pseudo-clases y muchas propiedades

Tabla 1 Soporte de CSS en los navegadores.
Fuente elaboración propia

2.5 Introducción a Javascript

2.5.1 Breve Historia de Javascript

A principios de los años 90, la mayoría de usuarios que se conectaban a Internet lo hacían con módems a una velocidad máxima de 28.8 Kbps. En esa época, empezaban a desarrollarse las primeras aplicaciones web y por tanto, las páginas web comenzaban a incluir formularios complejos (Eguíluz, 2010).

Con unas aplicaciones web cada vez más complejas y una velocidad de navegación tan lenta, surgió la necesidad de un lenguaje de programación que se ejecutara en el navegador del usuario. De esta forma, si el usuario no rellenaba correctamente un formulario, no se le hacía esperar mucho tiempo hasta que el servidor volviera a mostrar el formulario indicando los errores existentes (Eguíluz, 2010).

Brendan Eich, un programador que trabajaba en Netscape, pensó que podría solucionar este problema adaptando otras tecnologías existentes (como *ScriptEase*) al navegador Netscape Navigator 2.0, que iba a lanzarse en 1995. Inicialmente, Eich denominó a su lenguaje *LiveScript*.

2.5.2 ¿Qué es JavaScript?

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems.

2.6 HTML5 + CSS3 + JavaScript

HTML5 se podría decir que es sólo una marca constituida por varias tecnologías como lo son CSS3, JavaScript, y HTML Versión 5.

Lo que realmente nos trae esta combinación de tecnologías son páginas web mucho más ricas, completas, interactivas e interesantes que las que se hacen hoy en día. Infinitas posibilidades con gráficos e imágenes vectoriales, fotos, vídeos y otros elementos.

Las grandes compañías de la Red quieren que HTML5 + CSS3 + JavaScript sea la tecnología predominante para el desarrollo de sitios web. Esto ya que esta tecnología frente a otras como Flash o Silverlight, es de código abierto y por tanto común a todo el mundo.

Google ha puesto en marcha el sitio *HTML5Rocks.com* con ejemplos y recursos para desarrolladores. Lo mismo ha hecho Apple, que ha abierto un completo espacio en su sitio web con información de todo tipo. Y Microsoft, aunque todavía no tiene una página con recursos, sí ha dejado ver multitud de ejemplos en la página donde están las primeras pruebas de su la nueva versión de Internet Explorer, la 9.

2.7 Archivos SVG que es y su implementación en HTML5

Html5 se presenta como un marco estable para el desarrollo de páginas web. Ha sido definido para durar muchos años gracias a diversos mecanismos que permiten extender el lenguaje con facilidad. La filosofía de html5 aboga además por los estándares abiertos, que son fundamentales para impulsar la innovación e introducir en la sociedad los beneficios de las nuevas tecnologías (Huggers, 2010).

Debido al uso extendido de determinados complementos, como *Flash*, las nuevas funciones multimedia de html5 no pueden aspirar a sustituirlos por completo. Pero facilitan otros

métodos para presentar contenido multimedia. La intención es no depender tanto de los complementos externos y propietarios.

Los Gráficos Vectoriales Escalables (del inglés Scalable Vector Graphics) o SVG es una especificación para describir gráficos vectoriales bidimensionales, tanto estáticos como animados (Trippe, 2002).

SVG se convirtió en una recomendación del W3C en septiembre de 2001, por lo que ya ha sido incluido de forma nativa en el navegador web del W3C Amaya. Las versiones 1.5 y posteriores de Mozilla Firefox soportan gráficos hechos con SVG, así como el navegador Opera que desde su versión 8 ha implementado SVG 1.1 *Tiny* en su núcleo. Navegadores como *Google Chrome*, *Safari* e *Internet Explorer 9* también son capaces de mostrar imágenes en formato SVG sin necesidad de complementos externos. Otros navegadores web, como versiones anteriores a la 9 de Internet Explorer, necesitan un conector o *plug-in*, para lo que se puede utilizar el Visualizador SVG de Adobe (Trippe, 2002).

El SVG permite tres tipos de objetos gráficos:

- Formas gráficas de vectores (p.e. caminos consistentes en rectas y curvas, y áreas limitadas por ellos)
- Imágenes de mapa de bits /digitales
- Texto

Los objetos gráficos pueden ser agrupados, transformados y compuestos en objetos previamente interpretados, y pueden recibir un estilo común. El texto puede estar en cualquier espacio de nombres XML admitido por la aplicación, lo que mejora la posibilidad de búsqueda y la accesibilidad de los gráficos SVG. El dibujado de los SVG puede ser dinámico e interactivo. a cualquier objeto SVG (Trippe, 2002).

Para lograr un diseño correcto de procesos en un sistema es importante contar con herramientas que apoyen de manera efectiva el análisis, diseño e implementación, en el

siguiente tema se exponen las características del Lenguaje unificado de modelado que fue de gran apoyo para este desarrollo.

2.8 Lenguaje Unificado de Modelado

UML Mejor conocido como Lenguaje Unificado de Modelado por sus siglas en ingles (Unified Modeling Language) prescribe un conjunto de notaciones y diagramas estándar paramodelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan, mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación (Booch & Rumbaugh, 2006).

UML es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos. Empezó como una consolidación del trabajo de Grade Booch, James Rumbaugh, e Ivar Jacobson, creadores de tres de las metodologías orientadas a objetos más populares (Popkin, 2005).

En 1996, el Object Management Group (OMG), un pilar estándar para la comunidad del diseño orientado a objetos, publicó una petición con propósito de una meta-modelo orientado a objetos desemántica y notación estándares. UML, en su versión 1.0, fue propuesto como una respuesta a esta petición en enero de 1997. Hubo otras cinco propuestas rivales. Durante el transcurso de 1997, los seis promotores de las propuestas, unieron su trabajo y presentaron al OMG un documento revisado de UML, llamado UML versión 1.1. Este documento fue aprobado por el OMG en Noviembre de 1997. El OMG llama a este documento OMG UML versión 1.1. El OMG está actualmente en proceso de mejorar una edición técnica de esta especificación, prevista su finalización para el 1 de abril de 1999 (Popkin, 2005).

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware y organizaciones del mundo real. UML en su primera versión (UML 1.0) ofrece nueve diagramas de los cuales se describen los 7 más importantes a continuación.

2.8.1 Diagrama de Casos de Uso

Un diagrama de casos de uso es una especie de diagrama de comportamiento. UML mejorado El Lenguaje de Modelado Unificado define una notación gráfica para representar casos de uso llamada modelo de casos de uso. UML no define estándares para que el formato escrito describa los casos de uso, y así mucha gente no entiende que esta notación gráfica define la naturaleza de un caso de uso; sin embargo una notación gráfica puede sólo dar una vista general simple de un caso de uso o un conjunto de casos de uso. Los diagramas de casos de uso son a menudo confundidos con los casos de uso. Mientras los dos conceptos están relacionados, los casos de uso son mucho más detallados que los diagramas de casos de uso (Booch & Rumbaugh, 2006).

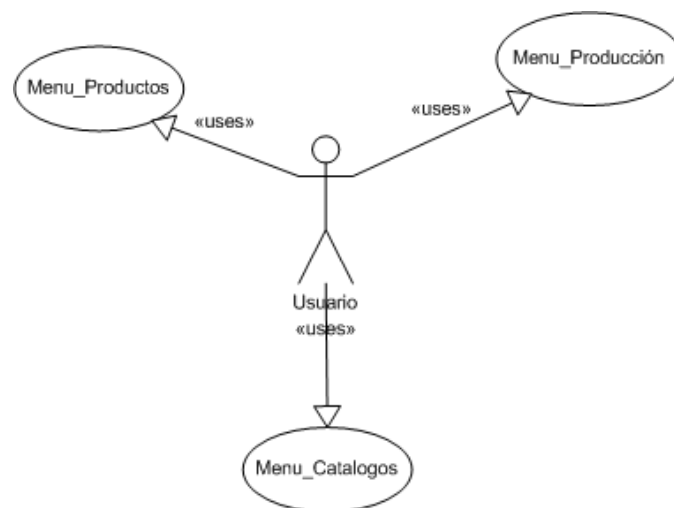


Figura 1 Ejemplo de diagrama de caso de uso.
Fuente elaboración propia

2.8.2 Diagramas de Secuencia.

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el diagrama de casos de uso permite el modelado de una vista de negocio del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y

clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos (Booch & Rumbaugh, 2006).

Típicamente se examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si se dispone de la descripción de cada caso de uso como una secuencia de varios pasos, entonces se puede "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos.

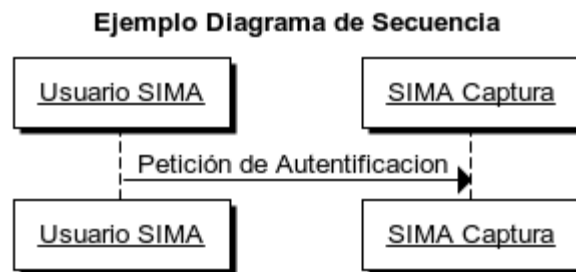


Figura 2 Ejemplo de diagrama de secuencia.
Fuente elaboración propia

2.8.3 Diagramas de Colaboración.

Es esencialmente un diagrama que muestra interacciones organizadas alrededor de los roles. A diferencia de los diagramas de secuencia, los diagramas de colaboración muestran explícitamente las relaciones de los roles. Por otra parte, un diagrama de colaboración no muestra el tiempo como una dimensión aparte, por lo que resulta necesario etiquetar con números de secuencia tanto la secuencia de mensajes como los hilos concurrentes.

2.8.4 Diagramas de Estado.

En UML, un diagrama de estados es un diagrama utilizado para identificar cada una de las rutas o caminos que puede tomar un flujo de información luego de ejecutarse cada proceso. Permite identificar bajo qué argumentos se ejecuta cada uno de los procesos y en qué momento podrían tener una variación. El diagrama de estados permite visualizar de una forma secuencial la ejecución de cada uno de los procesos.

2.8.5 Diagramas de Clases.

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro (Booch & Rumbaugh, 2006).

En la figura No. 3 se presenta un diagrama de clases generado por la información manejada en el sistema que se desarrolla.

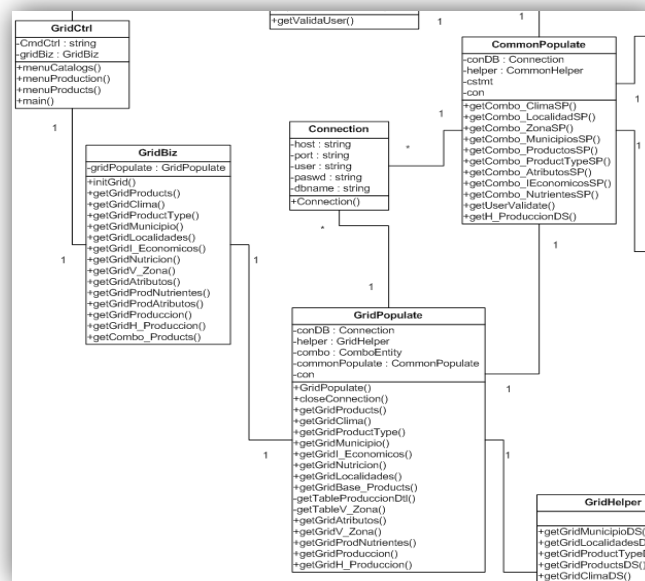


Figura 3 Ejemplo de diagrama de clases.

Fuente elaboración propia

2.8.6 Diagramas de Objetos.

Los diagramas de objetos son utilizados durante el proceso de Análisis y Diseño de los sistemas informáticos en la metodología UML. Se puede considerar un caso especial de un diagrama de clases en el que se muestran instancias específicas de clases (objetos) en un momento particular del sistema. Los diagramas de objetos utilizan un subconjunto de los elementos de un diagrama de clase. Los diagramas de objetos no muestran

la multiplicidad ni los roles, aunque su notación es similar a los diagramas de clase (Booch & Rumbaugh, 2006).

2.8.7 Diagramas de Componentes.

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema (Booch & Rumbaugh, 2006).

Debido a que estos son más parecidos a los diagramas de casos de usos estos son utilizados para modelar la vista estática y dinámica de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

El intercambio de información de diseño e ideas usando la notación UML sería hecho en los medios que siempre han sido populares: pizarras, cuadernos y trozos de papel por nombrar algunos. Pero UML se sirve mejor por una herramienta de modelado, la cual puede ser usada para capturar, guardar, rechazar, integrar automáticamente información, y diseño de documentación (Popkin, 2005).

Una característica que beneficia a los modeladores, UML también hace más fácil escoger una herramienta de modelado. Hace tiempo, el modelador primero tenía que seleccionar una notación de metodología, y después estaba limitado a seleccionar una herramienta que la soportara. Ahora con UML como estándar, la elección de notación ya se ha hecho para el modelador. Y con todas las herramientas de modelado soportando UML, el modelador puede seleccionar la herramienta basada en las áreas clave de funcionalidad soportadas que permiten resolver los problemas y documentar las soluciones (Popkin, 2005).

2.9 Proceso Unificado Racional

El Proceso Unificado Racional (*Rational Unified Process*) es un proceso moderno de desarrollo software que recoge elementos de todos los fundamentos de proceso comentados dentro del apartado de metodología, además de una serie de buenas prácticas que han de observarse para la obtención de un producto de calidad (Booch & Rumbaugh, 2006).

En RUP se puede ver la evolución del software en cuatro fases, al final de las cuales, y tras una serie de iteraciones, establece objetivos a alcanzar bien definidos.

2.9.1 Concepción.

El objetivo de esta fase es establecer los requisitos de negocio que cubrirá el sistema identificando todas las entidades que interactúan con el sistema (personas, sistemas, etc.) y hacer una valoración de la viabilidad del proyecto.

2.9.2 Elaboración.

Elaboración. El objetivo de esta fase es entender muy bien el problema desde el punto de vista del equipo de desarrollo. Lleva consigo la elaboración de la arquitectura marco del sistema y el diseño de la solución técnica, así como determinar el plan del proyecto e identificar los riesgos fundamentales del mismo. Al final de la fase se tiene definida la arquitectura, el modelo de requisitos del sistema empleando los diagramas de casos de uso especificados en lenguaje UML, el plan de desarrollo y los estándares de calidad que se han de seguir en el proyecto o las herramientas que se han de emplear durante el transcurso del mismo.

2.9.3 Construcción

En esta fase se profundiza en el diseño de los componentes y de manera iterativa se van añadiendo las funcionalidades al software a medida que se construyen y prueban, permitiendo a la vez que se puedan ir incorporando cambios.

Se podrán planificar entregas al final de cada iteración, momento en el que se recoge retroalimentación del usuario final y en el que se proponen cambios. Tras el análisis del impacto que suponen los mismos se decide si el mejor momento en que incorporar dichos cambios al sistema. Al final de la fase se tiene un sistema completamente operativo y la documentación para entregar a los usuarios.

2.9.4 Transición.

La fase final del RUP se ocupa del traslado del software desde los entornos de desarrollo a los entornos de producción, en los que el usuario final hará uso del sistema. Dependiendo del tipo de proyecto podrá requerir de entornos intermedios (preproducción o de aceptación por usuarios, etc.) para su correcta validación, antes de su pase a producción.

Cada fase se completa con la realización de varias iteraciones en las que se desarrollan una serie de actividades, que el modelo RUP clasifica en 9 disciplinas que tienen más o menos importancia en función de lo cerca que se esté o no de la finalización del proyecto.

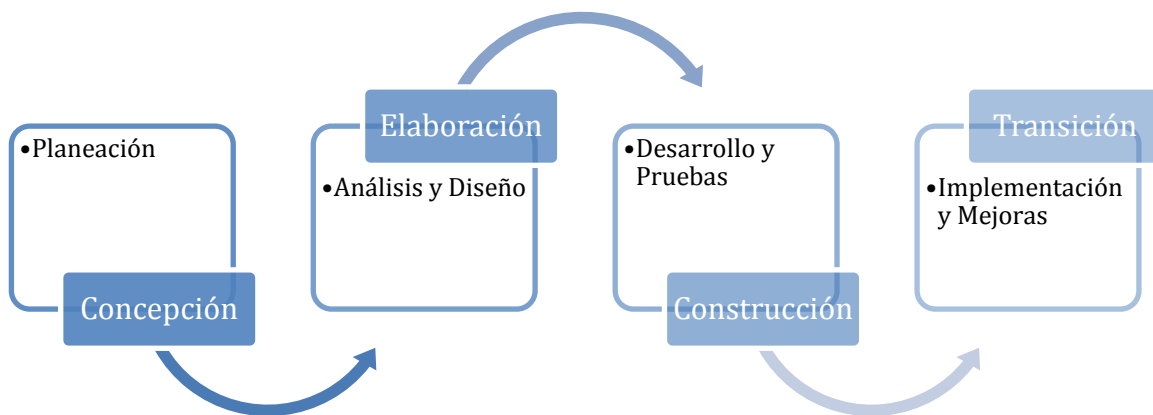


Figura 4 Fases del modelo RUP
Fuente elaboración propia

- **Modelado del negocio.** En este conjunto de actividades se persigue el entendimiento de las necesidades de negocio. Documentos de requisitos generales y de alto nivel, reglas del negocio, glosarios, etc. ayudan a definir lo que el producto software deba hacer.
- **Requisitos.** Traduce las necesidades del modelo de negocio a requisitos de sistemas automatizables y que con carácter más técnico (se emplean los casos de

uso UML), persiguen obtener un entendimiento más profundo del modelo de negocio por parte de los integrantes del equipo de desarrollo.

- **Análisis y diseño.** Estas actividades determinan, a partir de los requisitos la arquitectura del sistema más adecuada y el diseño detallado necesario previo a las actividades de implementación.
- **Implementación.** Actividades de codificación del software que de acuerdo al diseño, cumplen con los requisitos del sistema.
- **Pruebas.** Comprobaciones hechas a todos los elementos que se producen (documentos, diseños o código) para ver que cumplen con los requisitos y con los estándares de calidad definidos para el proyecto.
- **Despliegue.** Actividades que permiten tener el sistema instalado en los entornos en que finalmente va a ser explotado.
- **Gestión de configuración.** Gestión de los cambios y todos los elementos que intervienen en el proceso de construcción
- **Gestión del proyecto.** Actividades encaminadas a la gestión del desarrollo en cuanto a planes, recursos, seguimiento y control y gestión de riesgos.
- **Entorno.** Actividades que van encaminadas a dotar al proyecto de recursos hardware y software para facilitar la puesta en marcha y mantenimiento de los distintos entornos de desarrollo y pruebas o la propia puesta en producción del sistema.

RUP establece lo que denomina buenas prácticas como forma de trabajo adecuada para la consecución de objetivos que se pueden ir perfeccionando. Son las siguientes:

- Desarrollo iterativo que permita planificar desarrollos incrementales y entregas priorizando requisitos de modo que se entreguen antes las necesidades del usuario con mayor prioridad.
- Gestión de los requisitos: Documentar los requisitos y los cambios de los requisitos y analizar el impacto de cambios antes de aceptarlos.

- Emplear arquitecturas basadas en componentes para maximizar el aprovechamiento de desarrollos previos o componentes pre-construidos y abaratar los costes.
- Modelar visualmente el software empleando el estándar UML.
- Verificar la calidad de los productos del software asegurando que cumple los estándares de la compañía.
- Controlar los cambios del software.

Finalmente, cabe indicar que el modelo establece que el propio proceso es adaptable a cada caso de desarrollo. Por ejemplo, no todos los proyectos requieren del mismo nivel de documentación. El tamaño, la complejidad y el número de participantes entre otros aconsejan definir para el proyecto cómo adaptar el proceso. En la adaptación se definen qué artefactos hay que producir y en qué detalle, qué roles intervienen y las funciones que desempeñan dentro del proyecto, etc.

2.10 PHP& MySQL como herramienta de desarrollo

2.10.1 Breve Historia de PHP

PHP proviene de un programa anterior, llamado PHP/FI. PHP/FI fue creado por Rasmus Lerdorf en 1995, inicialmente como un simple conjunto de scripts de Perl para controlar quién accedía a su curriculum online. Llamó a ese conjunto de scripts *Personal Home Page Tools*. Según se requería, Rasmus fue añadiendo más funcionalidad escrita en el Lenguaje C que era capaz de comunicarse con bases de datos y permitía a los usuarios desarrollar sencillas aplicaciones Web dinámicas. Rasmus decidió volver libre el código fuente de PHP/FI para que cualquiera pudiese utilizarlo, así como arreglar errores y mejorar el código (Boronczyk, 2009).

PHP/FI, que se mantuvo para páginas personales y como intérprete de formularios, incluídas algunas de las funciones básicas de PHP tal y como lo conocemos hoy. Tenía variables como las de Perl, interpretación automática de variables de formulario y sintaxis

incrustada HTML. La sintaxis por sí misma era similar a la de Perl, aunque mucho más limitada, simple y algo inconsistente (Rasmus, 2011).

En 1997, PHP/FI 2.0, la segunda vez que se escribió la implementación en C, tuvo un seguimiento estimado de varios miles de usuarios en todo el mundo, aproximadamente 50.000 dominios tenían PHP/FI instalado, sumando alrededor del 1% de los dominios de Internet. Mientras había mucha gente contribuyendo con líneas de código a este proyecto, en su mayor parte era todavía el proyecto de una sola persona (Rasmus, 2011).

PHP/FI 2.0 no se liberó oficialmente hasta Noviembre de 1997, después de estar la mayoría del tiempo en versión beta. Después de poco tiempo de liberarse aparecieron las primeras versiones alpha de PHP 3.0 (Rasmus, 2011)

2.10.2 Lenguaje de programación interpretado PHP

PHP es un lenguaje de programación interpretado. PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools) (Boronczyk, 2009).

Un ejemplo del uso y potencia de este lenguaje son Wikipedia y la red Social Facebook, mismas que están desarrolladas en PHP. PHP Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Capítulo 3

Metodología

Como ya se mencionó este trabajo de investigación se realiza en conjunto con el grupo de investigación adjunto a la Facultad de Economía y Relaciones Internacionales, por lo que la parte de la investigación en lo que respecta al sector económico y determinantes de consumo, fué proporcionada por dicho grupo de investigación.

Para los procesos de desarrollo del sistema se utilizó el Proceso Racional Unificado (RUP), que es un proceso de desarrollo de software, como se menciona anteriormente, así como el Lenguaje Unificado de Modelado UML, ambos conforman la metodología estándar más utilizada para el análisis, implementación y documentación del sistema propuesto. La metodología RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización

A continuación se citan sus 4 fases y actividades asociadas a la misma:

3.1 Fase de planeación:

- a. Se determinó cronograma de actividades.
 - Se elaboró una gráfica de Gantt en la que se mostraron las actividades y/o tareas de forma cronológicamente en tiempo.
- b. Se realizaron entrevistas con el grupo de investigación del proyecto para determinar el universo de datos disponible para el sistema y el almacén de datos.
 - Se presentaron diagramas de Base de Datos en forma de tuplas relacionales, con la herramienta Microsoft Access para obtener retroalimentación del grupo investigativo.
- c. Determinación de requerimientos del sistema y la base de datos.
 - Debido a la magnitud de la información disponible por parte del grupo de investigación, fue necesario establecer un formato de presentación de la información, a este instrumento lo denominamos fichas técnicas del

producto. Las fichas técnicas, contienen la información general de producto, una tabla de los nutrientes que aporta dicho producto, y la serie histórica anual de los indicadores económicos más relevantes.

- d. Se realizaron análisis de los datos y requerimientos planteados durante las sesiones con el cuerpo académico.
- e. Se estableció un límite de productos según su tipo para poder realizar la representación de estos en el sistema multimedia.

3.2 Fase de elaboración:

- a. Realizar diagramas UML en base a los requerimientos obtenidos.
 - Para realizar los diagramas UML (Casos de uso, diagramas de flujo básico, y de colaboración) se utilizó la herramienta IBM Rational Rose.
- b. Basado en el análisis de universo diseñar modelos de E-R para la base de datos.
 - Se realizaron los modelos relacionales de la base de datos.
- c. Planear y/o realizar encuestas para obtener datos sobre el modo adecuado de llevar a cabo la interactividad del sistema multimedia.
 - Se realizaron sesiones de trabajo para establecer el modo adecuado y eficaz para la rápida captura de la información.
 - Se establecieron metas y estrategias para ir elaborando módulos del sistema y se fuera trabajando a la par en la captura de dicha información en ese módulo.
- d. Recopilar material hipermedia y multimedia para el desarrollo del sistema.
 - Se contemplaba el recibir apoyo para obtener material fotográfico y apoyo de personas del área de mercadotecnia para generar contenido didáctico.
 - Se establecieron acuerdos con la UABC, unidad valle de las Palmas a través de la Facultad de Diseño Gráfico para obtener contenido gráfico atractivo visualmente.

- e. Realizar modelos óptimos de interactividad con el usuario.

3.3 Fase de desarrollo:

- a. Creación de base de datos, basado en el modelo E-R elaborado.
 - Se generaron los scripts de base de datos localmente y se monto dicha base de datos en el servidor de base de datos MySQL que la FCA proporciona a alumnos de la Licenciatura en Informática.
- b. Desarrollo de sistema básico para la administración de la base de datos.
 - Debido a la amplia experiencia que existe en internet y la facilidad que implica el poder realizar una implementación de un sistema web en php, se opto por realizar el sistema de captura con el lenguaje de programación PHP 6.0.
- c. Captura en la base de datos del universo de datos propuesto para la primera etapa del proyecto.
 - Para poder realizar esta etapa se conto con el apoyo del grupo de investigación a través de 2 becarios para la captura de la información.
 - Se realizaron sesión de entrenamiento para el uso del sistema y para revisar avances de captura.
- d. Desarrollo del sistema de mapa interactivo e implementación de las técnicas de interacción obtenidas de la investigación.
 - La realización del mapa multimedia esta basada en un conjunto de vectores que conforman los contornos y delimitación de estado, municipios y valles.
 - Las capas establecidas en el mapa estarán basadas en la tecnología de archivos de vectores SVG.
 - Los elementos gráficos que están interactuando en el mapa multimedia deberán tener una resolución óptima para pantallas de 1024 x 768 px.
 - Se hizo uso de el lenguaje basado en scripts JavaScript para que los eventos de interacción en el mapa fueran intuitivos y volver la experiencia del usuario mucho más enriquecedora.

3.4 Fase de Pruebas/Implementación:

- a. Pruebas de los sistemas.
 - La implementación inicial se contempla para distribución del sistema en formato de Disco Compacto (CD) con un autoejecutable.
 - Para las pruebas de interacción del mapa multimedia se presento en AgroBaja, evento organizado por el municipio de Mexicali.
 - Para las pruebas del sistema de captura se realizaron prueba de caja negra, esto se presento el sistema al usuario y se le dejo interactuar con el y anotar las experiencias vividas.

- b. Implementación de los sistemas.
 - Se realizaron los scripts y los procesos necesarios para que el sistema de captura se migrara del servidor de alumnos de FCA hacia el servidor de la FEy RI. Durante la migración se pidió apoyo del encargado de laboratorio para obtener los accesos necesarios y dejar el sistema funcional.
 - El sistema multimedia se monto por primera vez en un servidor, y fue alojado también en el servidor de FEyRI.

- c. Coordinación para la elaboración de los manuales técnicos del sistema de captura.

Capítulo 4

Desarrollo

4.1 Planeación y Análisis

Como primer acercamiento al proyecto se realizaron una serie de entrevistas con todo el grupo de investigación, en estas reuniones se expusieron las inquietudes de alcance del proyecto, así como la factibilidad técnica del mismo, se planteo un proyecto ambicioso que se contempla para 2 o más etapas. Durante la primera etapa que es la que este trabajo de tesis contempla se planteo un sistema multimedia de difusión a los productos agropecuarios y pesqueros de la región de Baja California, enfocado al sector educativo, pero también la base de datos y sistema de captura que se utilizará en las siguientes etapas del proyecto.

El proyecto ya contaba con un amplio análisis de los productos agropecuarios y pesqueros más representativos de la región, dicho análisis arrojó los datos estadísticos por producto, por lo que se pidió complementar esa información con la tabla de nutrición y una descripción del producto. Todo esto se nos fue entregado en forma de fichas técnicas, misma que a continuación se muestra con un ejemplo representativo de la región.

Serie Histórica		
Producto		Uva
Grupo		Frutas
Nombre Científico		Vitis vinífera
Descripción Breve del Producto		Fruta. Baya de forma esférica que se agrupa en racimos, la cáscara es delgada y resistente, su color varía del verde limón al rojo solferino.

	Usos del Producto		Consumo humano: en fresco, deshidratadas o secas; solas, en ensaladas, postres o cubiertas con chocolate.
			Consumo industrial: en la elaboración de vinos y licores, jugos, jaleas y mermeladas. De la semilla se extrae aceite, el cual es comestible, además se utiliza en suplementos alimenticios.
	Distintivos del Producto		
	Varietades		Blancas (Riesling, Chardonnay, Chenin Blanc, Moscatel, Gewurztraminer, Sauvignon Blanc, Colombard y Palomino) y tintas (Cabernet Sauvignon, Anglianico, Cabernet Franc, Grenache, Barbera, Nebbiolo, Merlot, Pinoit Noir, Zinfandel, Tempranillo, Semillon y Syrah).
	Municipio		Ensenada, Mexicali, Tecate y Tijuana
	Valle		Valle de Guadalupe
	Microlocalización		
	Clima		
	Indicadores de Nutrición	Calorías Kcal.	kcal
Proteínas (g)		g	0,72
Grasa total (g)		g	0,16
Sodio (mg)		mg	2,00
Fibra(g)		g	0,80
Azucares (g)		g	15,48g

Potasio(mg)	mg	250,00
Magnesio(mg)	mg	10,00
Calcio(mg)	mg	17,00
Vitamina B6(mg)	mg	0,11
vitamina A(mg)	mg	1,00

Serie Histórica

Año	<i>Superficie cosechada (ha.)</i>	<i>Volumen producción (ton.)</i>	<i>Valor producción (miles de pesos, a precios corriente)</i>	<i>Superficie cosechada (ha.)</i>	<i>Volumen de producción (toneladas)</i>	<i>Valor de la producción (miles de pesos)</i>	<i>Volumen producción nacional (ton.)</i>	<i>Participación en volumen de producción a nivel nacional</i>
1984	63,397.00	539,223.00	19,575.37	7,059.00	37647	1,106.40	539223.00	6.98%
1985	61,704.00	592,750.00	33,503.40	6,838.00	63682	2,087.22	592750.00	10.74%
1986	73,014.00	666,669.00	71,322.15	6,486.00	23067	1,258.88	666669.00	3.46%
1987	54,605.00	516,169.00	148,892.57	6,665.00	44406	8,616.04	516169.00	8.60%
1988	51,943.00	552,668.00	318,678.36	6,544.00	46753	20,693.38	552668.00	8.46%
1989	50,419.00	502,470.00	504,558.59	6,475.00	32766	31,975.96	502470.00	6.52%
1990	47,189.00	428,896.00	558,814.57	6,081.00	29298	41,370.82	428896.00	6.83%
1991	45,831.00	529,579.00	1,002,588.62	5,986.00	39405	65,507.99	529579.00	7.44%
1992	42,527.00	522,041.00	754,853.88	5,749.00	43101	31,789.12	522041.00	8.26%
1993	42,670.00	466,596.00	693,288.77	5,427.00	45654	35,251.08	466596.00	9.78%
1994	41,854.00	536,924.00	627,824.27	5,236.00	51634	40,945.29	536924.00	9.62%
1995	42,836.00	475,857.00	1,047,271.01	5,099.00	32984	43,112.00	475857.00	6.93%
1996	41,724.00	408,275.10	1,132,310.25	5,257.00	31412.5	56,858.40	408275.10	7.69%
1997	39,443.00	473,337.00	1,829,239.01	5,498.00	52431	142,621.31	473337.00	11.08%
1998	39,135.00	478,047.00	2,382,942.73	5,379.00	49487	220,665.31	478047.00	10.35%
1999	40,158.00	482,805.35	2,062,988.49	5,049.00	42729.6	131,140.42	482805.35	8.85%

2000	39,154.03	371,795.66	1,889,068.67	5,023.03	33023.01	136,842.53	371795.66	8.88%
2001	37,341.50	435,685.74	2,524,362.31	4,962.50	35062.45	133,944.10	435685.74	8.05%
2002	32,904.32	363,000.19	3,090,203.36	3,571.07	25130.79	87,947.34	363000.19	6.92%
2003	30,685.00	331,250.05	3,584,272.82	3,789.50	28169.15	102,045.12	331250.05	8.50%
2004	32,970.75	305,278.78	2,399,607.42	3,307.50	22691.28	136,913.58	305278.78	7.43%
2005	30,013.80	331,897.61	3,030,654.27	3,342.90	19526.84	116,532.06	331897.61	5.88%
2006	29,324.30	244,073.43	2,647,986.55	3,007.40	17581.96	106,479.59	244073.43	7.20%
2007	29,267.90	356,257.85	4,518,464.13	2,687.00	17325.06	121,544.11	356257.85	4.86%
2008	26,547.25	266,089.10	3,459,430.42	2,918.20	16017.57	133,609.66	266089.10	6.02%
2009	~	~	~	2,933.77	20116.1	185,578.69	274,828.13	7.32%
TOTAL	1066657.85	11177634.86	40332701.97	130370.87	901101.31	2136436.387	11452462.99	7.80%

Tabla 2 Ejemplo de ficha técnica
Fuente elaboración propia

Como se puede observar la ficha técnica contiene un detalle de los productos y sus características más importantes para fines de los objetivos de esta tesis, así como los indicadores económicos de la región por año.

En la siguiente tabla se enlistan el universo de productos que el equipo de investigación de la FEyRI proporciono en forma de ficha técnica como la descrita en la tabla 2.

PRODUCTOS DE LA REGIÓN POR TIPO			
<i>Agrícola</i>		<i>Pecuario</i>	<i>Pesquero</i>
1. Algodón Hueso	2. Nopalitos	1. Leche Fresca	1. Anchoveta
3. Uva	4. Pepino	2. Carne Bovina	2. Pelo de Cochi
5. Fresa	6. Rabanito		3. Sargazo
7. Dátil	8. Tomatillo		4. Sargazo Rojo

9. Frambuesa	10. Zanahoria		5. Camarón
11. Cebolla	12. Cártamo		6. Tiburón
13. Cebollín	14. Maíz		7. Langosta Roja
15. Chile	16. Trigo		8. Erizo
17. Dátil	18. Sorgo		9. Pepino de Mar
19. Chile	20. Flores		10. Abulón
21. Cilantro	22. Alfalfa		11. Almeja
23. Esparrago	24. Sorgo Forrajero		12. Atún
25. Jitomate	26. Cebada		13. Macarela
27. Lechuga	28. Avena		14. Sardina Monterrey
29. Leek	30. Rye Grass		

Tabla 3 Universo de productos agropecuarios y pesqueros más representativos.
Fuente elaboración propia

Como se puede observar los productos de esta tabla están agrupados por tipo de producto, esto debido a que se contempla que la base de datos este desarrollada de tal forma que las características, atributos e indicadores económicos del producto pudieran variar para cada tipo de producto. También debido a que se requiere una serie histórica de indicadores económicos de los productos por año, será necesario tener un control de captura para subido masivo de estos valores o bien de actualización por año. Después del análisis realizado a las fichas técnicas proporcionadas, se determinó la necesidad de una base de datos relacional que contemplará el universo de productos locales de forma dinámica, esto es sin estar limitando a los productos a características o atributos fijos.

En la figura 5 podemos visualizar el modelo entidad relación, que fue resultado del análisis ya mencionado. Este modelo ER esta modela aplicando reglas de normalización y la tercera forma normal.

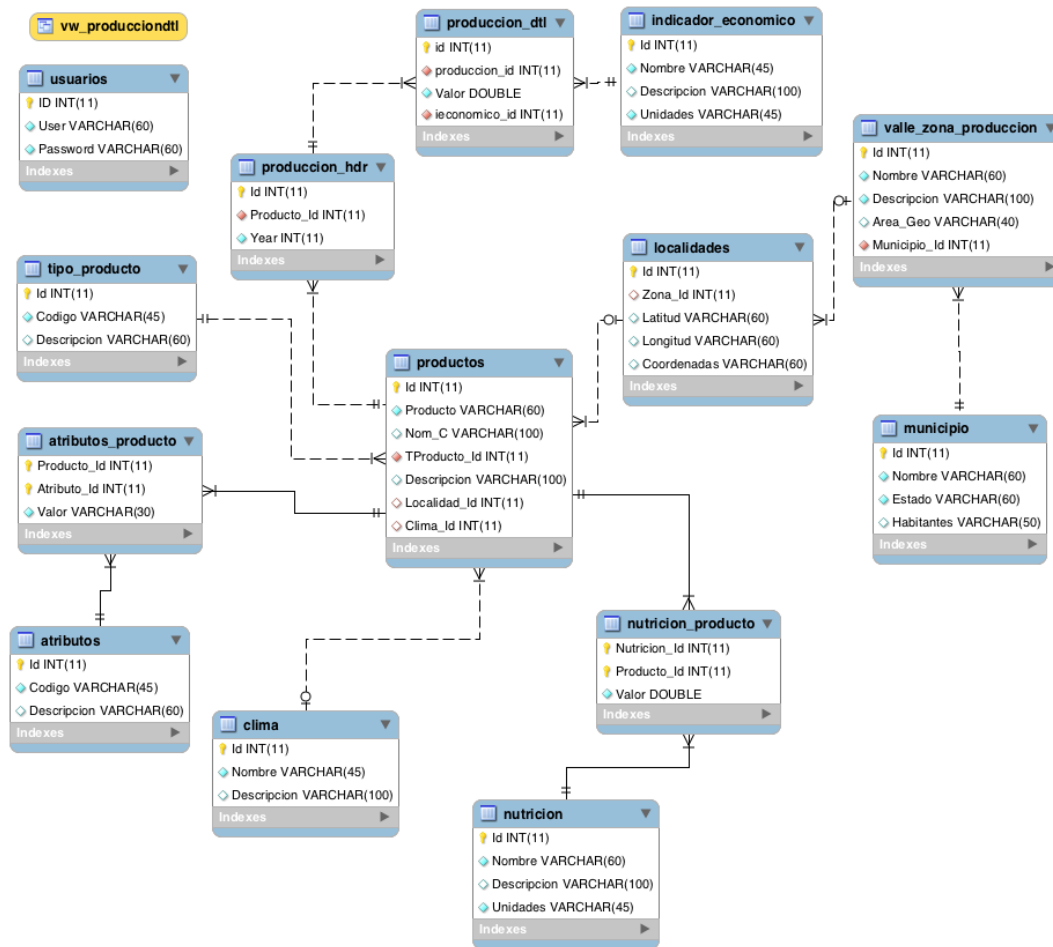


Figura 5 Modelo entidad relación del sistema de administración de Base de Datos.
Fuente elaboración propia

El motor de base de datos que se contempla utilizar para el proyecto es MySQL debido a su popularidad y fuerte integración con el ambiente de desarrollo web php.

4.1.1 Casos de Uso.

Como parte del análisis realizado se obtuvo la necesidad de un sistema de administración de la información que se alojara en la base de datos ya descrita por lo que se contemplo un sistema web de administración nombrado **SIMA Captura** (Sistema de Información Multimedia Agropecuario).

Los casos de uso que se identificaron para el sistema de captura se muestran a continuación, cabe mencionar que la tabla de descripción se encuentra como anexo a este trabajo de tesis, y solo para fin demostrativo se anexa la tabla del primer caso de uso.

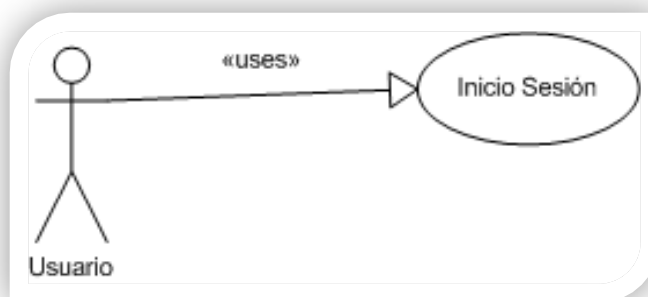


Figura 6 Diagrama de caso de uso “Inicio Sesión”.
Fuente elaboración propia

CASO DE USO		Inicio de Sesión	
CODIGO	UC01	Ref. Requerimiento	-
		Ref. Caso de Uso	-
Descripción general:			
Permite al usuario ingresar al sistema.			
Actores: Usuario			
Pre-Condiciones:			
<ul style="list-style-type: none"> • Carga del sitio exitosamente. • Se encuentre registrado el usuario en el sistema. 			
Pos-Condiciones:			
<ul style="list-style-type: none"> • Se mostrará el menú de opciones general al usuario. 			

Tabla 4 Descripción de caso de uso “Inicio de Sesión”.
Fuente elaboración propia

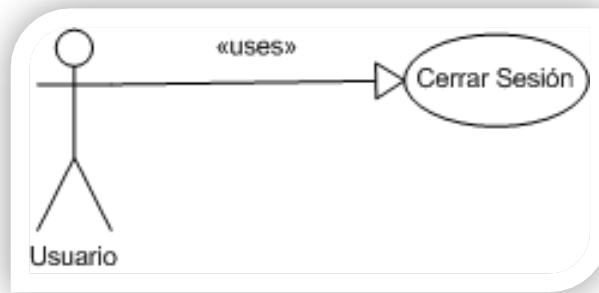


Figura 7 Diagrama de caso de uso “Cerrar Sesión”
Fuente elaboración propia

CASO DE USO		Cerrar Sesión	
CODIGO	UC02	Ref. Requerimiento	-
		Ref. Caso de Uso	-
Descripción general:			
Permite al usuario cerrar la sesión actual en el sistema.			
Actores: Usuario			
Pre-Condiciones:			
<ul style="list-style-type: none"> • Carga del sitio exitosamente. • Ingreso al sistema exitosamente. 			
Pos-Condiciones:			
<ul style="list-style-type: none"> • Se mostrará la pantalla de inicio de sesión. 			

Tabla 5 Descripción de caso de uso “Cerrar Sesión”.
Fuente elaboración propia

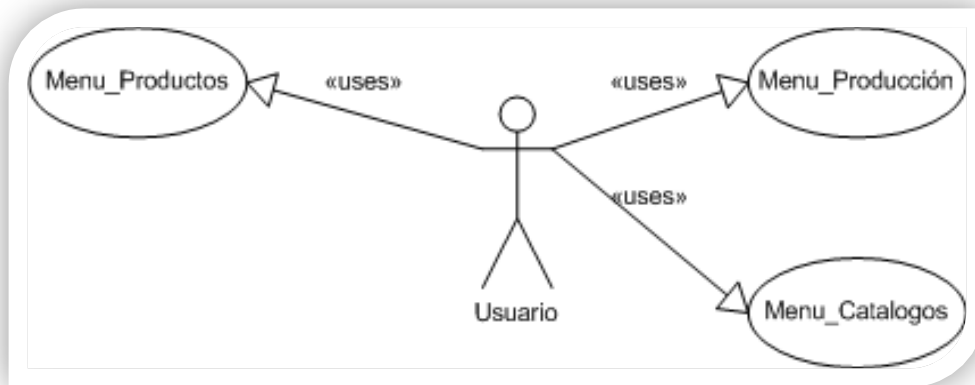


Figura 8 Diagrama de caso de uso “Menú”.
Fuente elaboración propia

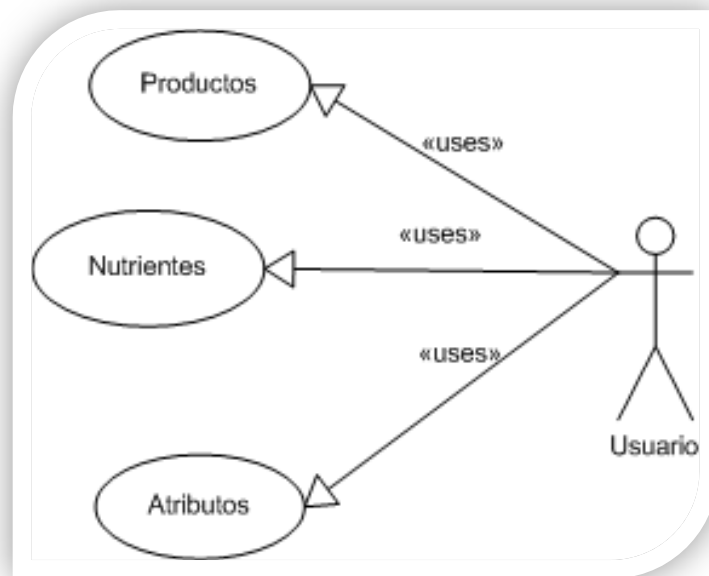


Figura 9 Diagrama de caso de uso “Menú Productos”
Fuente elaboración propia

CASO DE USO		Menú Productos	
CODIGO	UC03	Ref. Requerimiento	-
		Ref. Caso de Uso	-
Descripción general:			
Permite al usuario ingresar a las opciones del menú productos.			
Actores: Usuario			
Pre-Condiciones:			
<ul style="list-style-type: none"> • Ingreso al sistema exitosamente. 			
Pos-Condiciones:			
<ul style="list-style-type: none"> • Se mostrará las opciones del menú productos en el lado izquierdo de la pantalla. 			

Tabla 6 Descripción de caso de uso “Menú Productos”.
Fuente elaboración propia

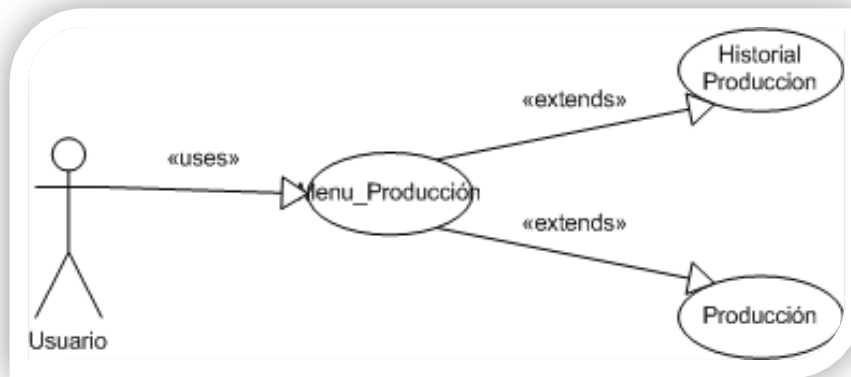


Figura 10 Diagrama de caso de uso Menú Producción
Fuente elaboración propia

CASO DE USO		Menú Producción	
CODIGO	UC04	Ref. Requerimiento	-
		Ref. Caso de Uso	-
Descripción general:			
Permite al usuario ingresar a las opciones del menú producción.			
Actores: Usuario			
Pre-Condiciones:			
<ul style="list-style-type: none"> • Ingreso al sistema exitosamente. 			
Pos-Condiciones:			
<ul style="list-style-type: none"> • Se mostrará las opciones del menú productos en el lado izquierdo de la pantalla. 			

Tabla 7 Descripción de caso de uso “Menú Producción”.
Fuente elaboración propia

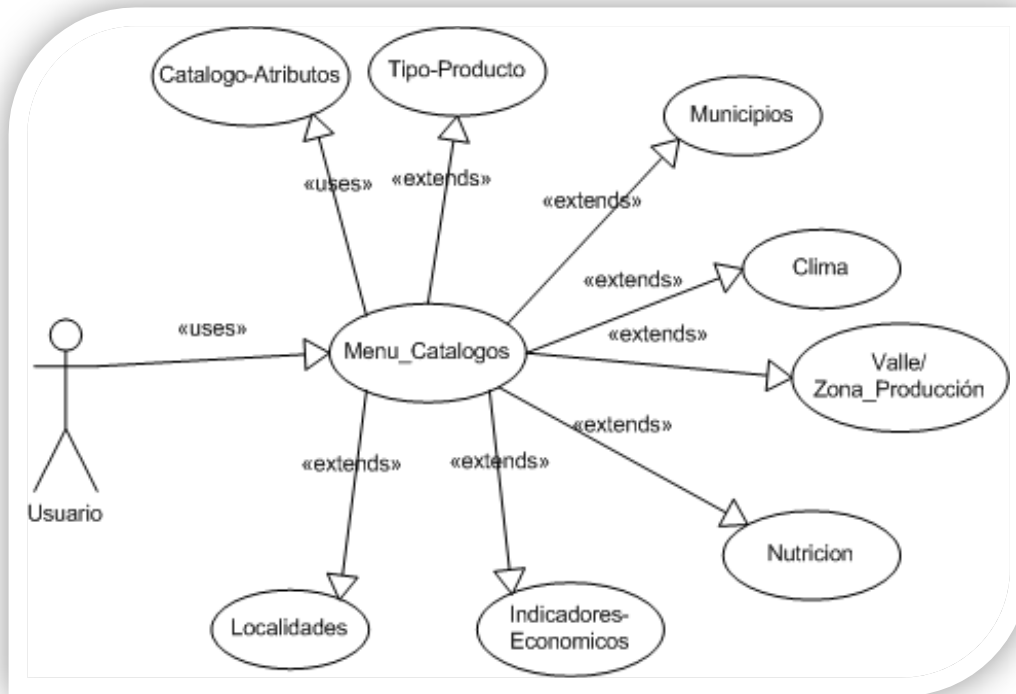


Figura 11 Diagrama de caso de uso “Menú Catálogos”.
Fuente elaboración propia

CASO DE USO		Menú Catálogos	
CODIGO	UC05	Ref. Requerimiento	-
		Ref. Caso de Uso	-
Descripción general:			
Permite al usuario ingresar a las opciones del menú catálogos.			
Actores: Usuario			
Pre-Condiciones:			
<ul style="list-style-type: none"> • Ingreso al sistema exitosamente. 			
Pos-Condiciones:			
<ul style="list-style-type: none"> • Se mostrara las opciones del menú productos en el lado izquierdo de la pantalla. 			

Tabla 8 Descripción de caso de uso “Menú Catálogos”.
Fuente elaboración propia

Para obtener el detalle y los casos de uso ver anexo X.

4.1.2 Diagrama de Clases.

Una vez identificado el flujo básico del proceso de captura para el sistema, se estableció un desarrollo de aplicación web basado en capas. Una aplicación web típica está compuesta por una capa de presentación (funcionalidad relacionada con la interfaz de usuario), una capa de negocios (funcionalidad relacionada con el procesamiento de reglas de negocios) y una capa de datos (funcionalidad relacionada con el acceso a datos).

Basado en los beneficios que ofrecen los *frameworks* de permitir a los desarrolladores tener un código más ordenado, limpio y fácil de actualizar, un código más seguro y robusto y mucho más eficiente. Optamos por desarrollar el sistema basado en una infraestructura de PHP basada en *grids* que funjan con la funcionalidad de alta, bajas, modificaciones y consultas.

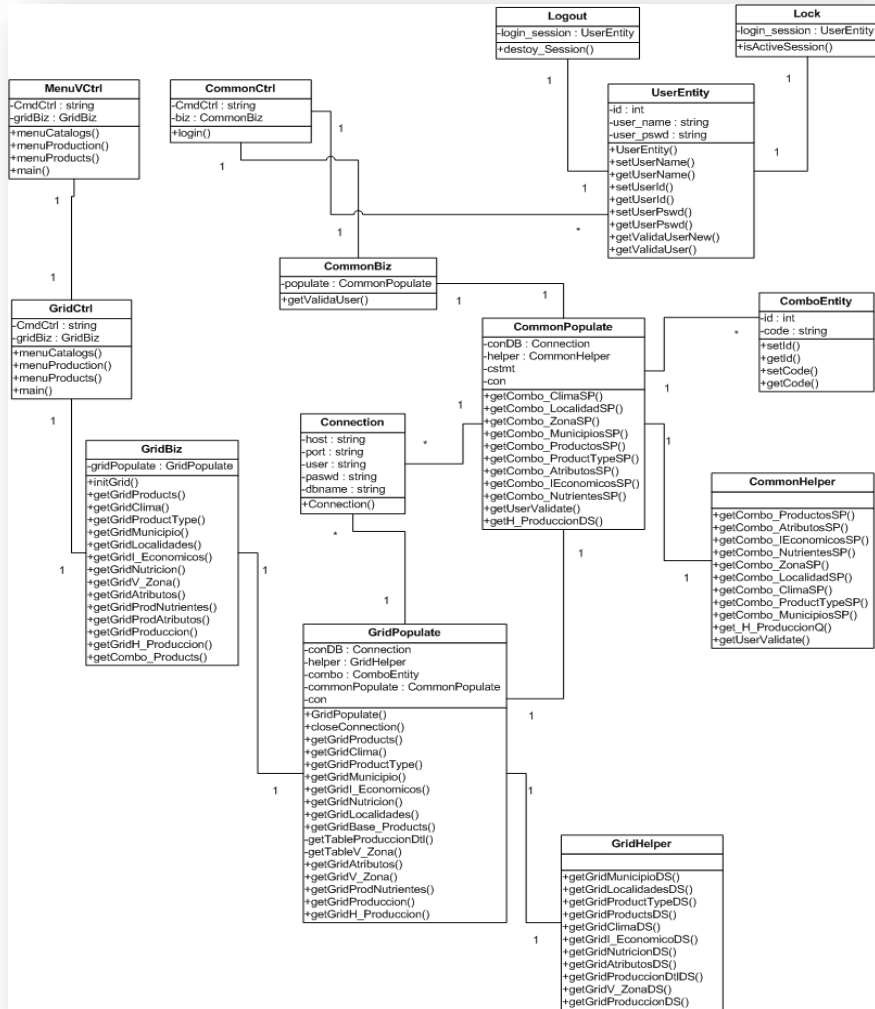


Figura 12 Diagrama de clases para sistema de captura
Fuente elaboración propia

4.2 Diseño y Desarrollo

4.2.1 Descripción y desarrollo de SIMA

El producto final de esta tesis es el denominado proyecto SIMA (Sistema de Información Multimedia Agropecuaria) Este proyecto está dividido en 2 fases, una fase es el sistema de captura y otro es el mapa multimedia.

Para poder lograr los objetivos del proyecto será necesario desarrollar el sistema de captura, como primera fase, esto ya que las actividades de captura son las que tomarán más tiempo según el calendario de planeación del proyecto.

A continuación en la tabla 9 se muestra la lista de los productos más representativos de la región.

PRODUCTOS MÁS REPRESENTATIVOS.	
1. Uva	2. Algodón
3. Fresa	4. Esparrago
5. Frambuesa	6. Jitomate
7. Carne Bovina	8. Chile
9. Pepino	10. Cebolla
11. Zanahoria	12. Cebollín
13. Leche Fresca	14. Tiburón (Cazón)
15. Trigo	16. Langosta Roja
17. Flores	18. Almeja
19. Lechuga	20. Atún
21. Alfalfa	22. Sardina
23. Huevo	

Tabla 9 Productos más representativos de B.C.
Fuente elaboración propia

Las siguientes figuras muestran las pantallas más representativas del sistema de captura del proyecto SIMA, y con el cual el usuario final realizara la captura de la información correspondiente a los productos agrícolas, pecuarios y pesqueros



Figura 13 Pantalla de iniciar sesión en el sistema SIMA
Fuente elaboración propia

En la *figura 14* se muestran los menús con los que el sistema SIMA cuenta siendo los estos ***Productos, Producción, Catálogos*** y el ultimo la opción de ***salir del sistema***.



Figura 14 Menús del sistema SIMA
Fuente elaboración propia

Al iniciar sesión como se muestra en la *figura 13* al sistema el usuario visualiza la primera opción del menú **Productos** seleccionada, siendo esta la pantalla de administración de productos.

La opción producto del menú **Productos** esta seguida por **Localidades**, **Atributos** y **Nutrientes** estas dos pantallas corresponden a la figura 16, 17 y 18 respectivamente, y son las pantallas que nos permitirán agregar la información correspondiente a la localidad(es), Nutrientes del producto así como todas las características del mismo.

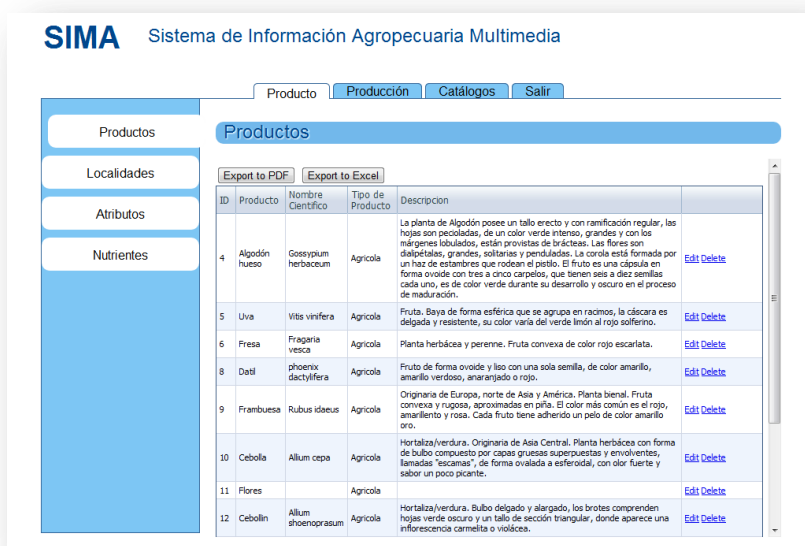


Figura 15 Pantalla de administración de productos en el sistema SIMA
Fuente elaboración propia

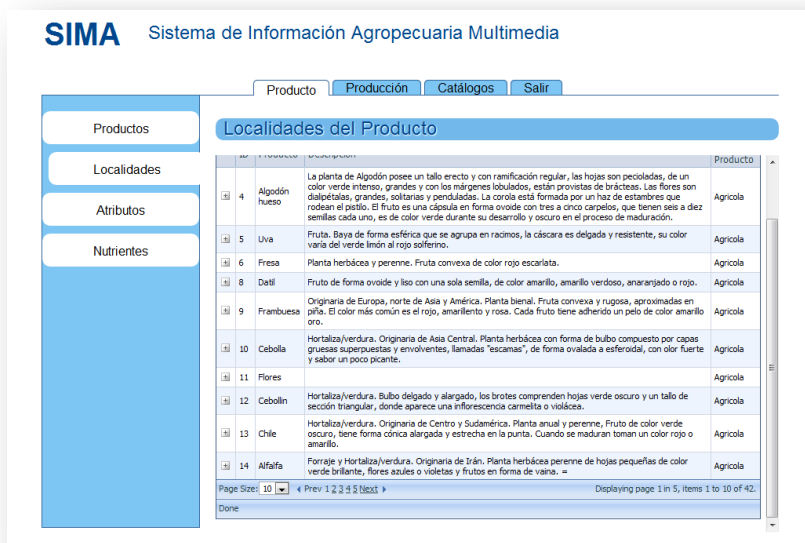


Figura 16 Pantalla localidades del producto
Fuente elaboración propia

Producto Producción Catálogos Salir

Atributos del Producto

Export to PDF Export to Excel

Productos: All

ID	Producto	Descripción	Tipo de Producto
4	Algodón hueso	La planta de Algodón posee un tallo erecto y con ramificación regular, las hojas son pecioladas, de un color verde intenso, grandes y con los márgenes lobulados, están provistas de brácteas. Las flores son dialpétalas, grandes, solitarias y penduladas. La corola está formada por un haz de estambres que rodean el pistilo. El fruto es una cápsula en forma ovoide con tres a cinco carpelos, que tienen seis a diez semillas cada uno, es de color verde durante su desarrollo y oscuro en el proceso de maduración.	Agrícola
5	Uva	Fruta. Baya de forma esférica que se agrupa en racimos, la cáscara es delgada y resistente, su color varía del verde limón al rojo sofferino.	Agrícola
6	Fresa	Planta herbácea y perenne. Fruta convexa de color rojo escarlata.	Agrícola
8	Dátil	Fruto de forma ovoide y liso con una sola semilla, de color amarillo, amarillo verdoso, anaranjado o rojo.	Agrícola
9	Frambuesa	Originaria de Europa, norte de Asia y América. Planta bienal. Fruta convexa y rugosa, aproximadas en piña. El color más común es el rojo, amarillento y rosa. Cada fruto tiene adhiriendo un pelo de color amarillo oro.	Agrícola
10	Cebolla	Hortaliza/verdura. Originaria de Asia Central. Planta herbácea con forma de bulbo compuesto por capas gruesas superpuestas y envolventes, llamadas "escamas", de forma ovalada a esferoidal, con olor fuerte y sabor un poco picante.	Agrícola
11	Flores		Agrícola
12	Cebollín	Hortaliza/verdura. Bulbo delgado y alargado, los brotes comprenden hojas verde oscuro y un tallo de sección triangular, donde aparece una inflorescencia carmelita o violácea.	Agrícola
13	Chile	Hortaliza/verdura. Originaria de Centro y Sudamérica. Planta anual y perenne. Fruto de color verde oscuro, tiene forma cónica alargada y estrecha en la punta. Cuando se maduran toman un color rojo o amarillo.	Agrícola
14	Alfalfa	Forraje y Hortaliza/verdura. Originaria de Irán. Planta herbácea perenne de hojas pequeñas de color	Agrícola

Figura 17 Pantalla Atributos del Producto
Fuente elaboración propia

Producto Producción Catálogos Salir

Nutrientes del Producto

Export to PDF Export to Excel

Productos: All

ID	Producto	Descripción	Tipo de Producto
4	Algodón hueso	La planta de Algodón posee un tallo erecto y con ramificación regular, las hojas son pecioladas, de un color verde intenso, grandes y con los márgenes lobulados, están provistas de brácteas. Las flores son dialpétalas, grandes, solitarias y penduladas. La corola está formada por un haz de estambres que rodean el pistilo. El fruto es una cápsula en forma ovoide con tres a cinco carpelos, que tienen seis a diez semillas cada uno, es de color verde durante su desarrollo y oscuro en el proceso de maduración.	Agrícola
5	Uva	Fruta. Baya de forma esférica que se agrupa en racimos, la cáscara es delgada y resistente, su color varía del verde limón al rojo sofferino.	Agrícola
6	Fresa	Planta herbácea y perenne. Fruta convexa de color rojo escarlata.	Agrícola
8	Dátil	Fruto de forma ovoide y liso con una sola semilla, de color amarillo, amarillo verdoso, anaranjado o rojo.	Agrícola
9	Frambuesa	Originaria de Europa, norte de Asia y América. Planta bienal. Fruta convexa y rugosa, aproximadas en piña. El color más común es el rojo, amarillento y rosa. Cada fruto tiene adhiriendo un pelo de color amarillo oro.	Agrícola
10	Cebolla	Hortaliza/verdura. Originaria de Asia Central. Planta herbácea con forma de bulbo compuesto por capas gruesas superpuestas y envolventes, llamadas "escamas", de forma ovalada a esferoidal, con olor fuerte y sabor un poco picante.	Agrícola
11	Flores		Agrícola
12	Cebollín	Hortaliza/verdura. Bulbo delgado y alargado, los brotes comprenden hojas verde oscuro y un tallo de sección triangular, donde aparece una inflorescencia carmelita o violácea.	Agrícola
13	Chile	Hortaliza/verdura. Originaria de Centro y Sudamérica. Planta anual y perenne. Fruto de color verde oscuro, tiene forma cónica alargada y estrecha en la punta. Cuando se maduran toman un color rojo o amarillo.	Agrícola
14	Alfalfa	Forraje y Hortaliza/verdura. Originaria de Irán. Planta herbácea perenne de hojas pequeñas de color	Agrícola

Figura 18 Pantalla Nutrientes del Producto
Fuente elaboración propia

En lo que respecta al menú de catálogos (figura 18), se agrupo a todas las pantallas y/o opciones que corresponden al producto de manera indirecta y que tienden a ser repetitivos con respecto a cada producto, como es localidades, indicadores económicos, tipo de clima, tipo de producto, Etc.



Figura 19 Pantalla de Catálogos del Producto
Fuente elaboración propia

Por último y no menos importante está el menú de producción en el cual el usuario podrá capturar la serie histórica de indicadores económicos por producto y año.



Figura 20 Pantalla Nutrientes del Producto
Fuente elaboración propia

También en este mismo menú se encuentra la opción de Ficha Técnica que es donde se muestra el resumen de información del producto en forma de ficha técnica con la finalidad de realizar impresión a modo reporte o simplemente ver de manera resumida la información de algún producto específico.



Figura 21 Pantalla de Ficha Técnica.
Fuente elaboración propia

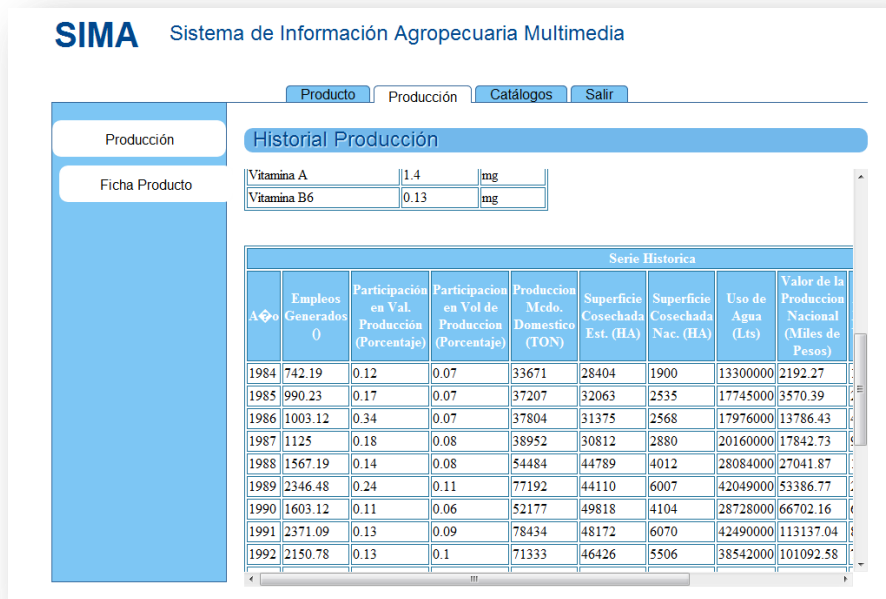


Figura 22 Pantalla de Ficha técnica con indicadores de producción.
Fuente elaboración propia

Como anexo a este trabajo de tesis, encontraremos un manual de usuario final, que fue entregado a los usuarios del sistema de captura.

4.2.2 Diseño sistema Multimedia

El sistema multimedia es el segundo producto final del proyecto por lo que es la parte culminante del proyecto. El sistema multimedia agropecuario y pesquero arroja un reto a nivel creativo y de diseño, esto es como proponer a un estudiante del nivel educativo básico un producto agropecuario de forma agradable y que despierte interés en él. La primera imagen en la que se pensó al pensar en un mapa interactivo del sistema multimedia fue en el juego por excelencia de la red social *Facebook*, ***FarmVille***, este sistema desarrollado en interfaz web es un juego que arroja miles de dólares a sus creadores, todo esto gracias a su modo de interactuar con el usuario y el ecosistema propuesto dentro de la red social ya mencionada y sin duda a su atractivo visual proporcionado.

La primera propuesta hecha para desarrollar el sistema multimedia fue mediante archivos de datos SVG, renderizados por HTML manejando áreas interactivas. El primer ejercicio a modo prototipo desarrollado, fue un mapa de Baja California, dividido por municipios, ejemplificado en la *figura 20*. Dicho mapa hasta ese momento solo contaba con la habilidad de seleccionar un área y cambiar su color.



Figura 23 Primer prototipo Mapa Interactivo
Fuente elaboración propia

Una vez establecida la factibilidad del desarrollo de animaciones sin hacer uso de la herramienta obvia (Adobe Flash). Realizamos una propuesta-prototipo del modo en el que el mapa interactuaría con el usuario a modo de mostrar el producto siendo este el centro de atención.

El concepto artístico inicial y prototipo que se presentó ante el grupo de investigación de FEyRI fue el de mostrar un ambiente muy al estilo de caricatura o serie animada con colores vivos y atractivos. Así como el de presentar una caracterización de los productos como si de un personaje se tratara.

El mapa muestra iconos representativos de alguna localidad dada, y posteriormente mostrar un menú de bienvenida por localidad, dicho menú mostrara un primer plano de la representación de la localidad a modo caricatura con una galería de los productos como si estos se trataran de habitantes de dicha localidad (ejemplificado en la *figura 22*).

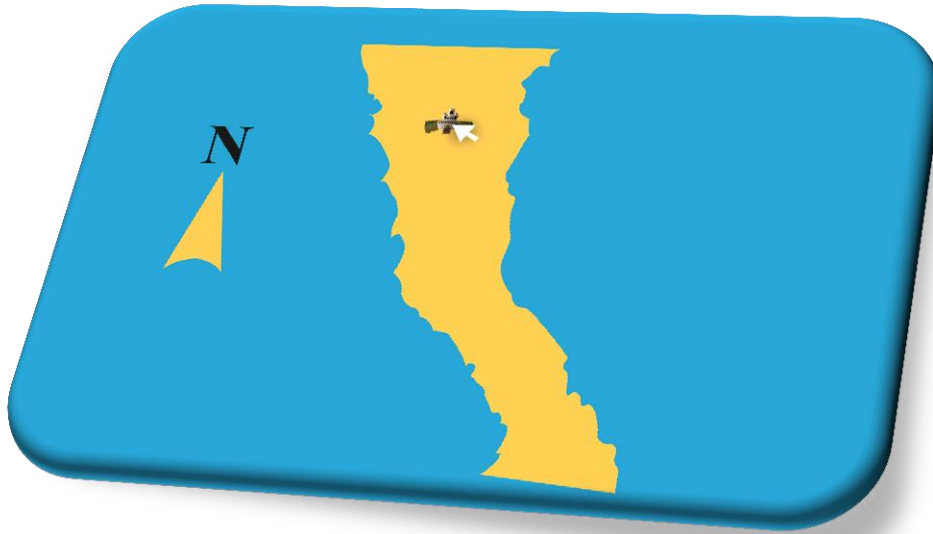


Figura 24 Prototipo de interacción básica con el usuario.
Fuente elaboración propia

En la figura 25 se muestra la propuesta de interacción básica con el usuario y primer acercamiento con el usuario, esto es la selección de localidad en la que se encuentra interesado el usuario, cabe mencionar que las localidades son puntos geográficos determinados como valles agrícolas.

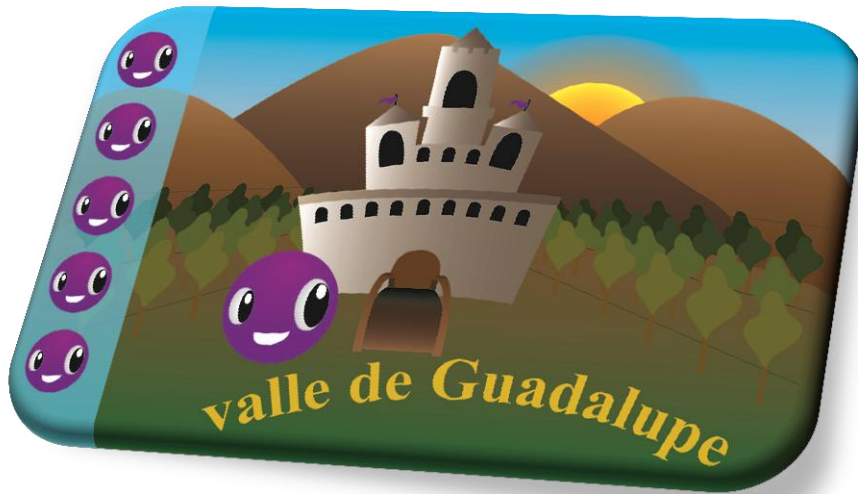


Figura 25 Prototipo pantalla de bienvenida a localidad.
Fuente elaboración propia

La figura 26 se muestra la propuesta del modo en que el producto se presentara a modo de habitante de alguna localidad, y una descripción básica de dicho producto.



Figura 26 Prototipo pantalla de presentación del producto.
Fuente elaboración propia

En la la pantalla que acontinuacion se presenta ilustra una descripción más detallada del producto, esto es sus indicadores de nutrición.

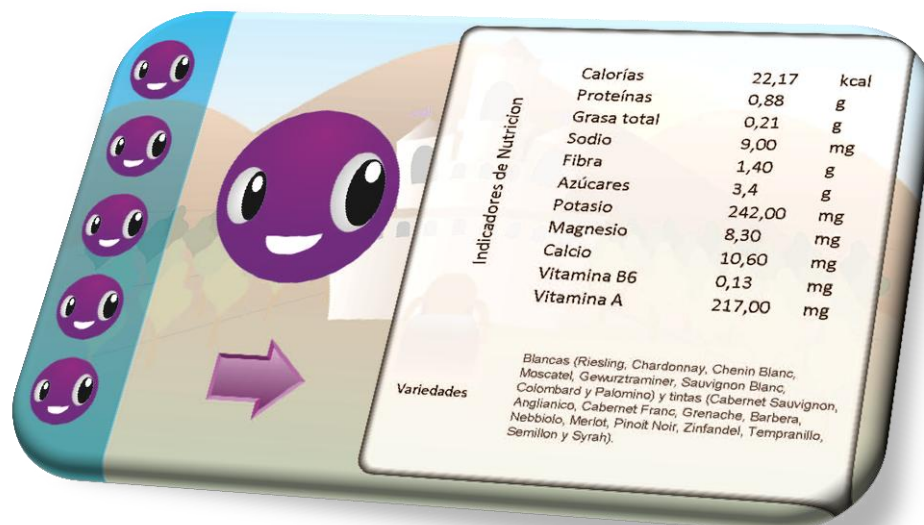


Figura 27 Prototipo pantalla detalle del producto.
Fuente elaboración propia

4.2.3 Desarrollo de Mapa Multimedia SIMA

El sistema multimedia fue un reto creativo más que un reto técnico. Como primera etapa se estableció la viabilidad técnica del desarrollo de un mapa con funcionalidad dinámica. Para esto determinamos que la funcionalidad del mapa no se puede limitar a ser desarrollada en HTML, es decir fue necesario elegir alguna herramienta o tecnología que nos permitiera tener eventos de selección de áreas dentro del mapa multimedia, así como eventos y que también nos permitiera gestionar la vista del mapa por capas. Pero tal vez lo más importante y difícil de conseguir sería que el mapa multimedia tuviese la habilidad de ejecutarse en los navegadores web más populares. Es por esto que se optó por utilizar como tecnología base solo HTML como visualizador y JavaScript como motor de funcionalidad en los eventos, descartando herramientas como Silverlight o Flash.

La librería de JavaScript necesario para establecer los eventos dentro del mapa ya existen en internet y es un código de acceso libre. Este código lo que hace es dibujar gráficos de vectores escalabres (SVG) y establecerles un nombre como identificador único para poder identificar los eventos que al se le asignaran.

```

(function() {
  var ms = 500;
  var r = Raphael('map', 1000.00, 670.00),
      attributes = {
        fill: 'url(#hiSiSi/Miniatu/as/mi1_pasto.png)',
        stroke: '#3899E6',
        'stroke-width': 1,
        'stroke-linejoin': 'round'
      },
      attributesII = {
        fill: '#D8D8D8',
        stroke: '#3899E6',

```

Figura 28 Representativa de JavaScript.
Fuente elaboración propia

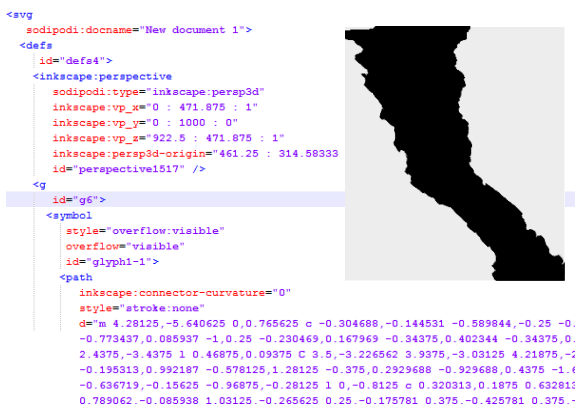


Figura 29 Representativa de SVG.
Fuente elaboración propia

Los archivos SVG, son interpretados nativamente como imágenes o dibujos en los navegadores mas utilizados. Cabe mencionar que la estructura de los archivos SVGeS XML avanzados, donde se establecen las dimensiones y resolución del dibujo o gráfico, así como las coordenadas de los puntos que permiten formar el contorno del gráfico, también se puede establecer un color código de color hexadecimal para relleno de dicho gráfico.

4.2.3.1 Desarrollo del mapa multimedia en capas.

Con la finalidad de llevar un mejor control y orden dentro de proyecto SIMA en el Mapa Multimedia, se estableció una jerarquía de capas con los códigos SVG y la nomenclatura que la librería de JavaScript interpretaría mas adelante.

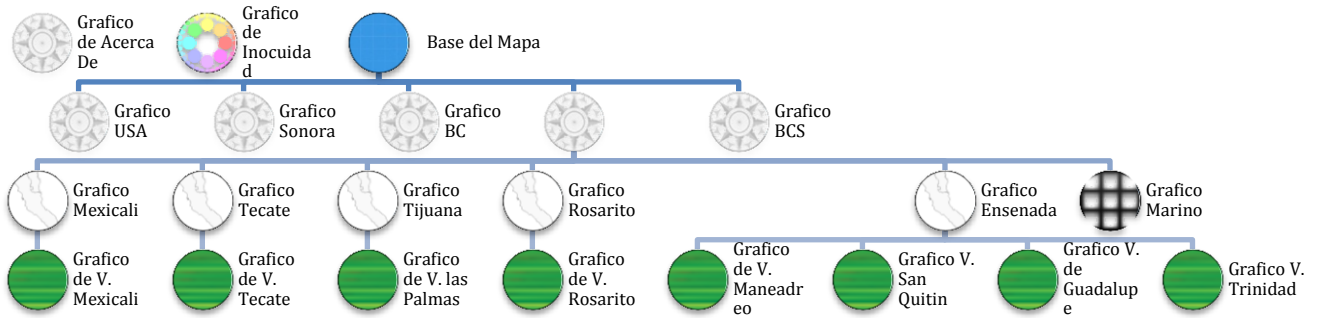


Figura 30 Esquema de Capas para los códigos SVG.
Fuente elaboración propia

El modelo representado por la figura 30 presenta las 4 capas que en el proyecto se definió como necesarias para poder administrar de manera dinámica los áreas establecidos en el sistema que ahora llamaremos valles agropecuarios o marinos.

- 1) En la primera capa se observa la base o fondo del mapa (Figura 30), que no es mas que un grafico de color azul, con detalles de cuadrícula.



Figura 31 Fondo Mapa SIMA.
Fuente elaboración propia



Figura 32 Icono de "Acerca De"
Fuente elaboración propia

También a este nivel encontraremos 2 gráficos (Figura 31 y 32) que nos representara el "Acerca De" y "Reglas de Inocuidad" respectivamente.

Estos iconos tendrán funcionalidad, es decir al dar clic sobre ellos lanzaran un evento.



Figura 33 Icono de Inocuidad.
Fuente elaboración propia

- 2) La segunda capa contempla todo lo gráficos que no tienen eventos dinámicos o no contemplan funcionalidad. Esto es los gráficos definidos para USA, Sonora y BCS. Véase figura 33

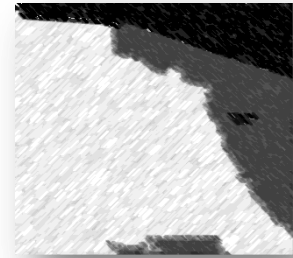


Figura 34 Gráficos estáticos
Fuente elaboración propia

- 3) La tercera capa contempla todo lo gráficos que no tienen eventos dinámicos pero tienen la funcionalidad de alojar los valles. La funcionalidad clave de esta capa es delimitar los municipios del estado de Baja California y enriquecer la dinámica del uso del mapa multimedia. Véase figura 34

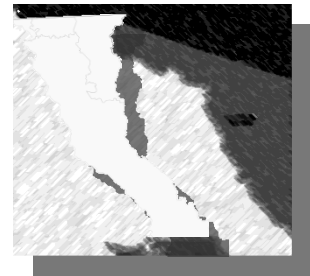


Figura 35 Gráficos delimitadores (Municipios)
Fuente elaboración propia

- 4) La cuarta capa contempla todo lo gráficos que tendrán funcionalidad y/o eventos dentro del mapa multimedia.

Se contempla que estas áreas denominadas valles, tengan evento de sobre posición (*OnOver*) y que al dar un clic nos muestre una pantalla de información de dicho valle seleccionado.

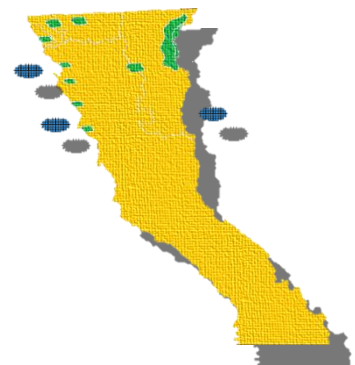


Figura 36 Gráficos con eventos (Valles)
Fuente elaboración propia

En la figura 37 podemos apreciar una imagen de una porción del código necesario para poder formar los gráficos de las diferentes capas dentro del mapa multimedia. Dicho código está estructurado en formato XML y es interpretado por el navegador como se ilustra en la figura.

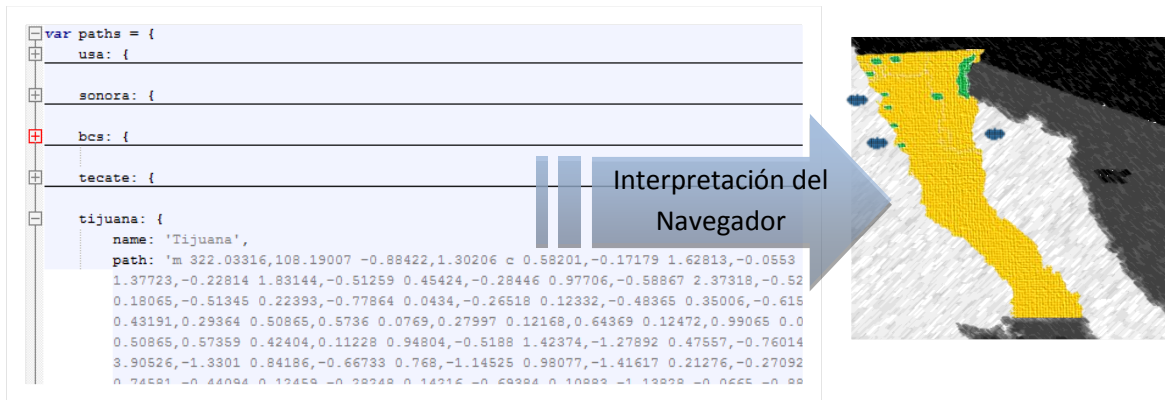


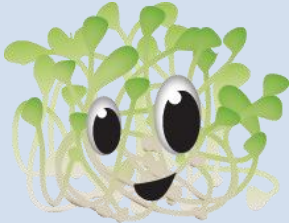







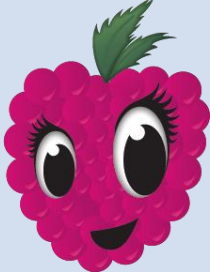
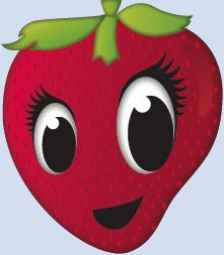

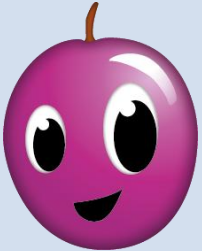




Figura 37 Prototipo pantalla detalle del producto.
Fuente elaboración propia

El código de todos los gráficos y elementos del mapa se encuentran como anexos a este trabajo de tesis.

4.2.3.2 Elementos gráficos y diseño del mapa multimedia.

A continuación encontraremos los diseños finales aprobados de los productos agropecuarios y pesqueros. Estas representaciones animadas de los productos están enfocadas al sector educativo por lo que es de suma importancia que sean amigables y agradables visualmente.

También encontraremos los diseños preliminares y finales de los valles agropecuarios y pesqueros.

 <p>Alfalfa</p>	 <p>Algodón</p>	 <p>Cebolla</p>	 <p>Cebollín</p>
 <p>Chile</p>	 <p>Esparrago</p>	 <p>Pepino</p>	 <p>Lechuga</p>
 <p>Frambuesa</p>	 <p>Fresa</p>	 <p>Tomate</p>	 <p>Uva</p>
 <p>Zanahoria</p>	 <p>Trigo</p>	 <p>Clavel</p>	 <p>Huevo</p>

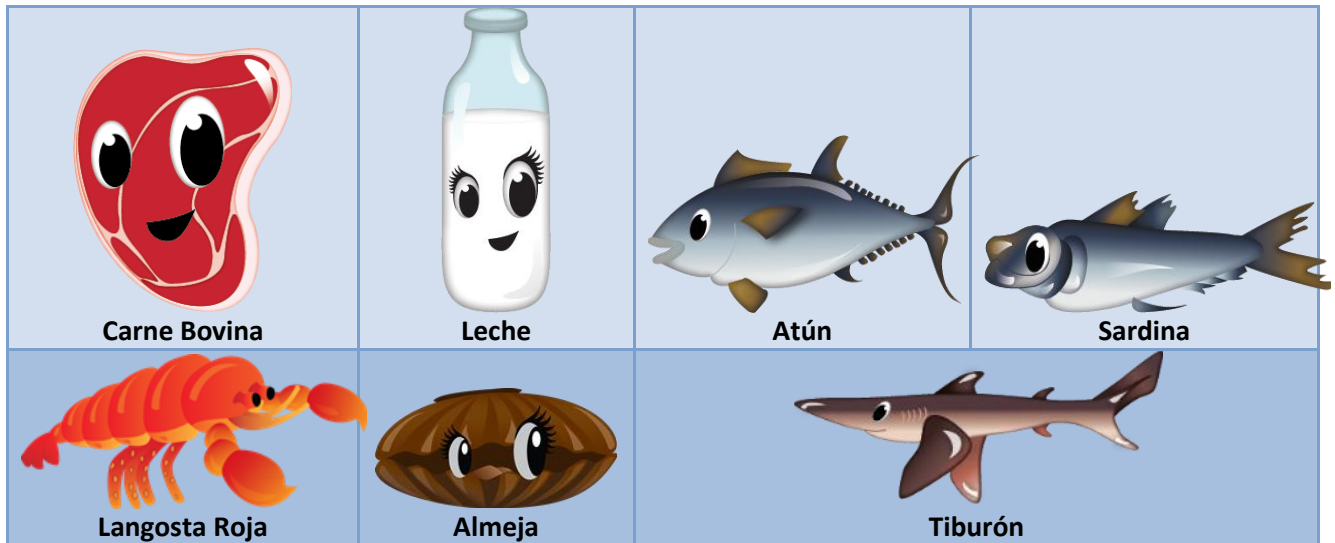


Tabla 10 Tabla de diseños finales de productos agropecuarios y pesqueros.
Fuente elaboración propia

En la ilustración 38 podemos observar los cambios y resultados obtenidos en el valle de Mexicali. En las imágenes del lado izquierdo podemos ver las propuestas y de lado derecho el resultado final. Se puede observar como la representación de este valle es un paisaje plan centrado en un amplio campo de cultivo, soleado y un amplio canal de riego.

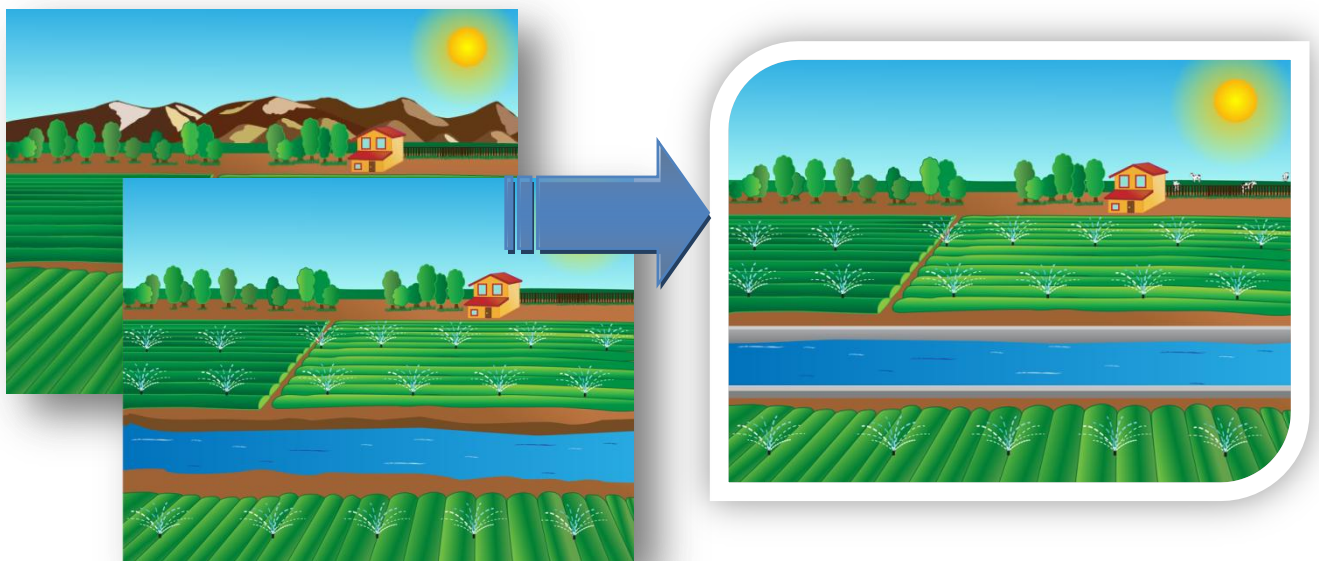


Figura 38 Gráficos Valle de Mexicali.
Fuente elaboración propia

Las imágenes representadas por la figura 39 podemos ver la radical transformación de Punta Colonet de un paisaje con puerto y casas coloridas a un paisaje campirano, con amplios valles de cosechas, soleado y con un puerto de fondo

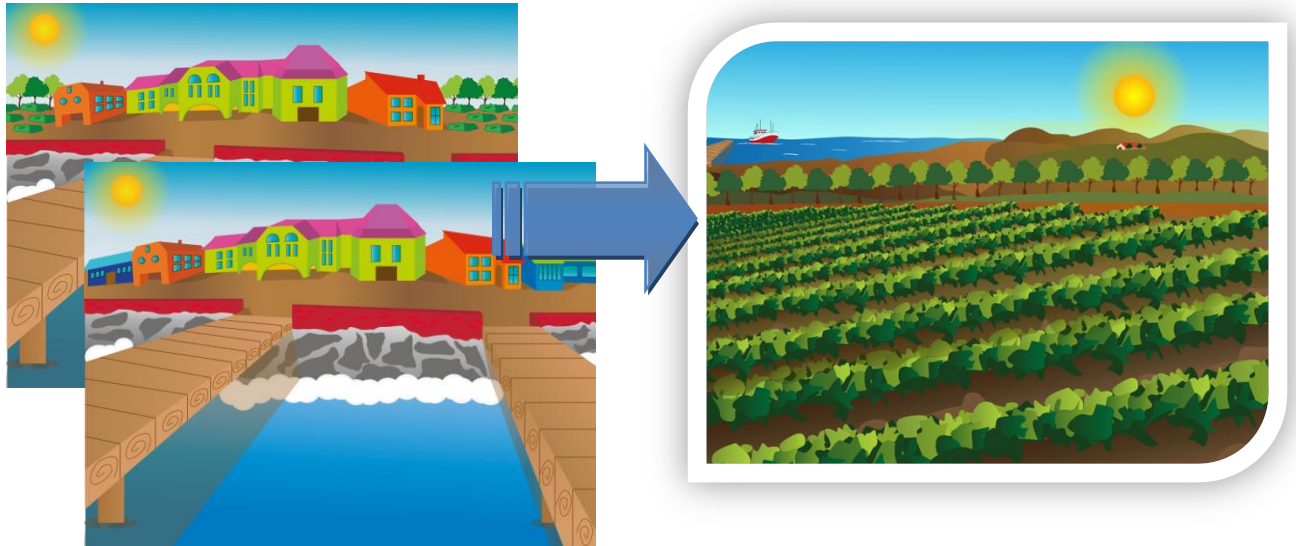


Figura 39 Gráficos Valle Punta Colonet.
Fuente elaboración propia

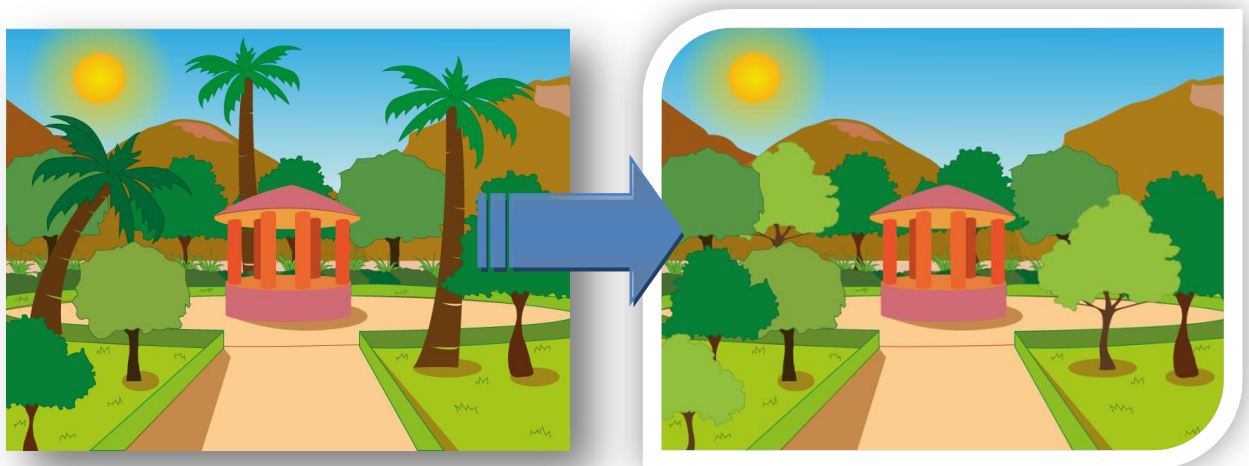


Figura 40 Gráficos Valle Tecate.
Fuente elaboración propia

En la ilustración 40 podemos observar los diseños del Valle de Tecate, el paisaje que se puede observar es la plaza representativa de Tétate. Los cambios que ocurrieron desde su prototipo hasta su versión final son mínimos y es el tipo de flora que se muestra, es decir cambio de palmeras por arboles.

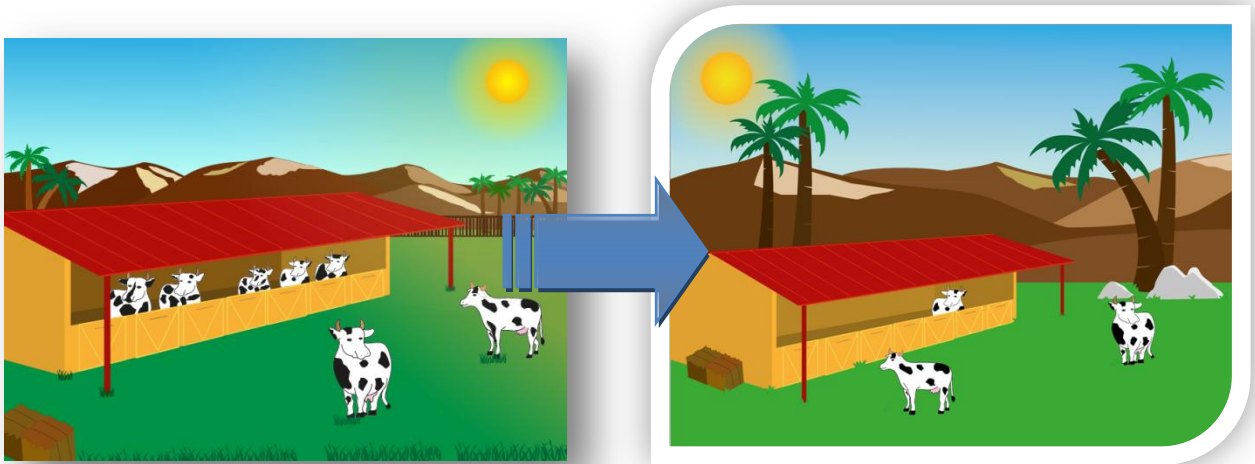


Figura 41 Gráficos Valle de las Palmas.
Fuente elaboración propia

La figura 41 presenta la representación del Valle de las palmas y como se puede observar ambas propuestas eran bastante buenas y al final quedo aquella con menos vacas como elementos principales.

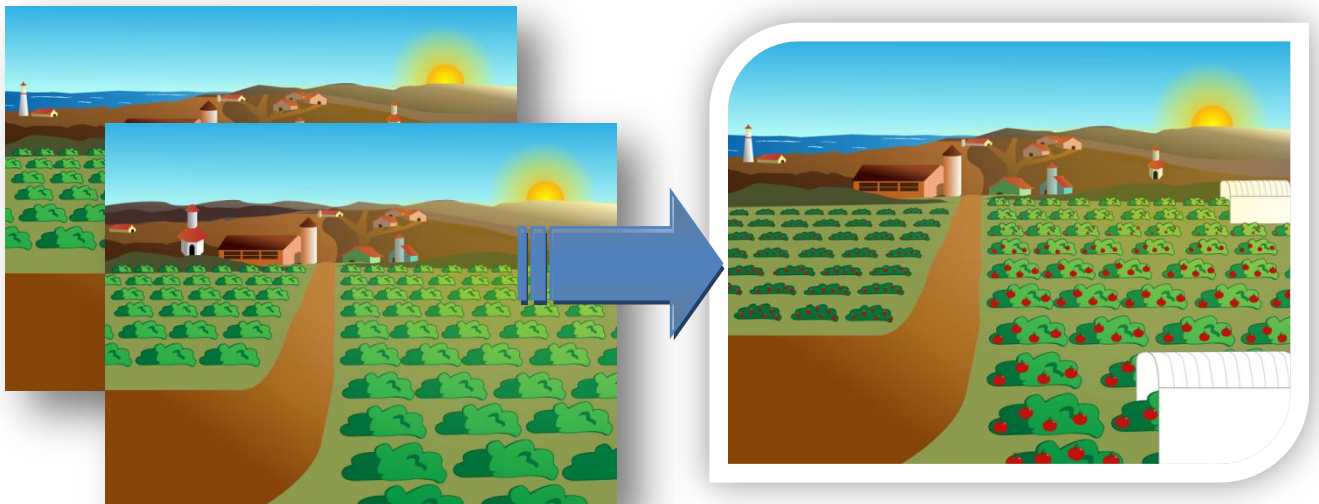


Figura 42 Gráficos Valle de las Palmas.
Fuente elaboración propia

En la ilustración a continuación se observan los diseños del Valle de San Quintín, en el prototipo inicial no se mostraba mar, y ya en un segundo prototipo se muestra tanto mar como el faro representativo. En el paisaje final se muestra un paisaje con amplios campos de cultivos, mar y colinas.

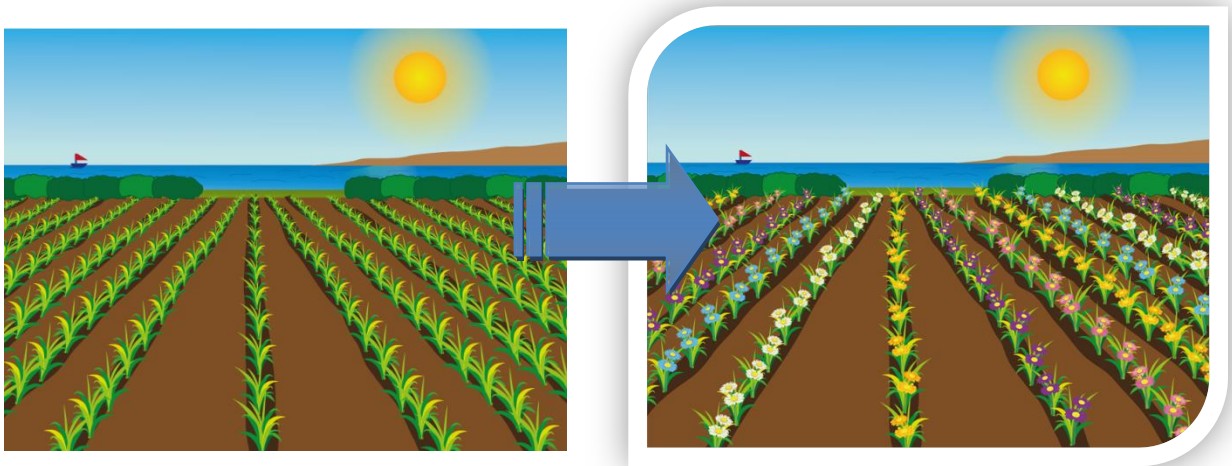


Figura 43 Gráficos Valle de Maneadero.
Fuente elaboración propia

La ilustración 43, se observan los diseños del Valle de Maneadero, el prototipo inicial y el diseño final son muy parecidos. Su cambio consta esencialmente del producto agrícola representativo que como se puede observar son flores en un paisaje soleado y con la bahía de ensenada al fondo.

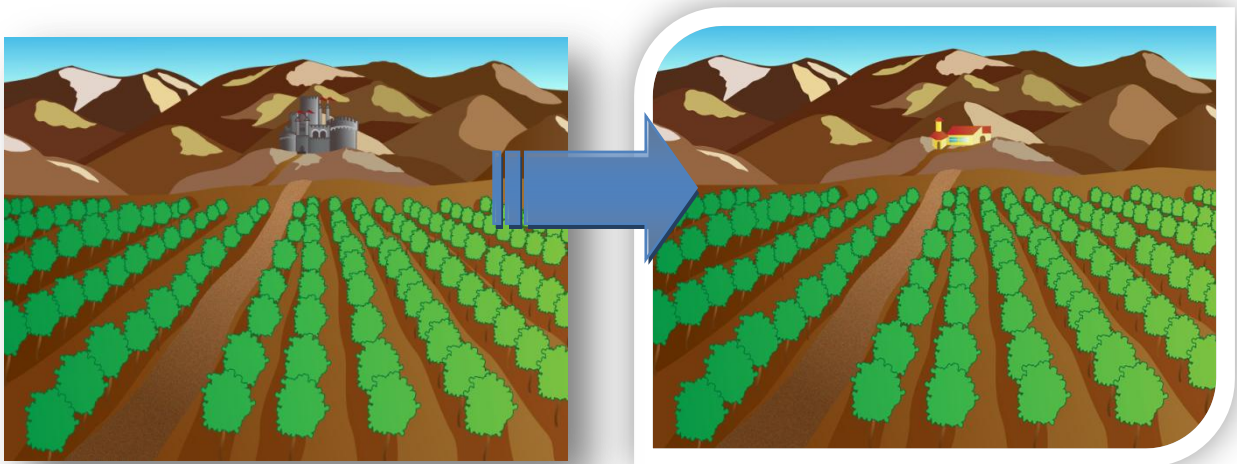


Figura 44 Gráficos Valle de Maneadero.
Fuente elaboración propia

A la figura anterior podemos observar los diseños del Valle de Guadalupe, el prototipo inicial y el diseño final son muy parecidos. Su cambio consta esencialmente en el diseño de un castillo por una casa tipo campirana en el cerro. En el paisaje podemos observar amplios campos agrícolas destinados a la uva.

A continuación se muestra el prototipo final (Figura 45) del Valle de Rosarito. Solo se muestra un diseño final ya que el prototipo fue evaluado y no hubo una propuesta concreta de mejora por lo que se dejó justo como el prototipo inicial.

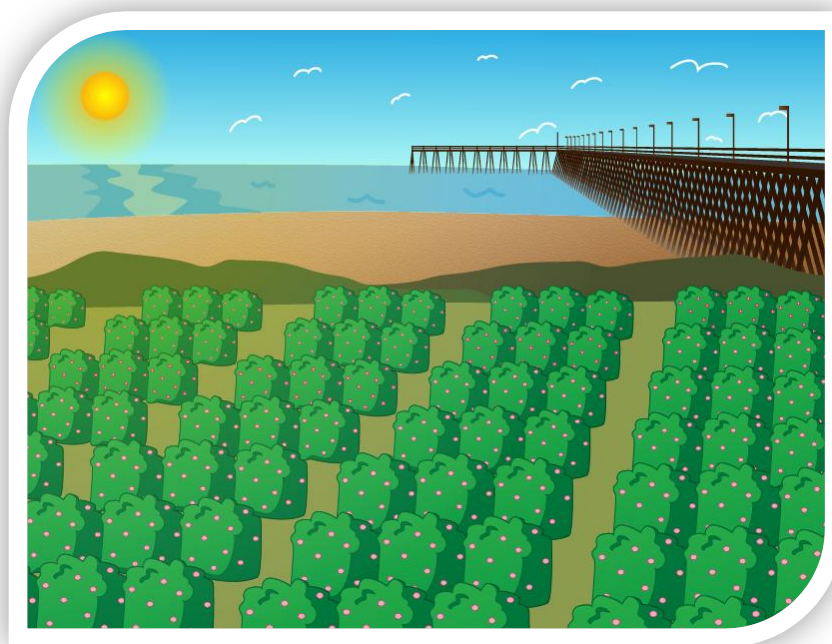


Figura 45 Gráficos Valle de Rosarito.
Fuente elaboración propia

La figura 46 presenta el prototipo final del Valle de la Trinidad. Solo se muestra un diseño final ya que el prototipo fue evaluado y no se encontraron mejoras a la propuesta. El paisaje nos muestra una amplia área de cosecha con un fondo de cerros y soleado.



Figura 46 Gráficos Valle de la Trinidad.
Fuente elaboración propia

A continuación se observa los diseños del Valle Marino, el prototipo inicial y el diseño final son completamente distintos. Inicialmente se tenía una idea de tener un paisaje de la superficie marina, para después decidirse que se usara una imagen del contorno marino.

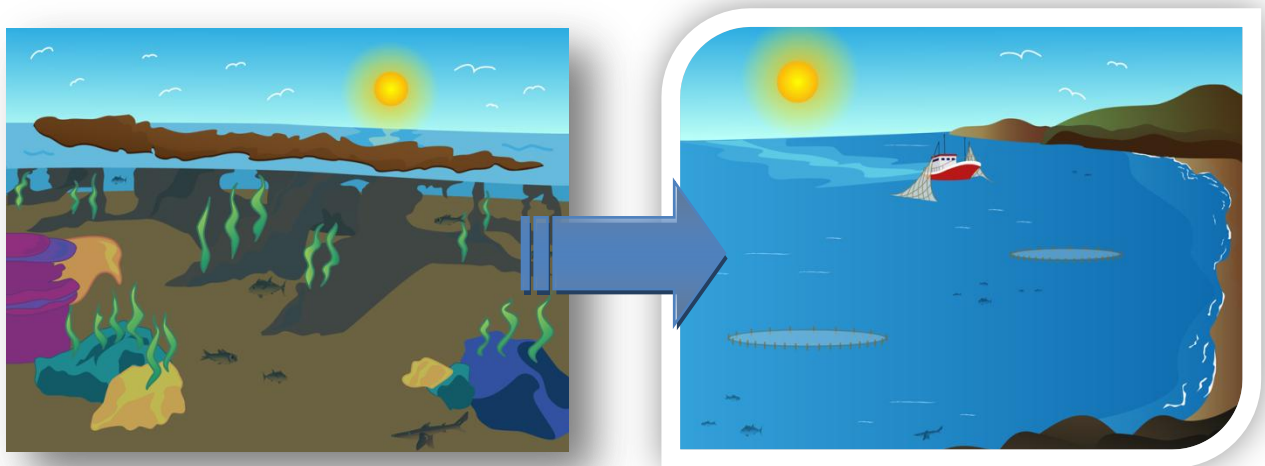


Figura 47 Gráficos Valle Marino.
Fuente elaboración propia

4.2.3.3 Contenedor de información de los productos.

Debido a que el sistema multimedia podría ser distribuido a través de disco compacto en el proyecto fue necesario implementar un método de distribución de la información fuera de línea (sin conexión a Internet). En la figura 49, podemos observar el la estructura del archivo XMLbase del sistema multimedia. Este archivo esta compuesto de la información general y detallada de los productos a manera de jerarquía misma que se ilustra en la figura 48. Esta información puede ser generada a través del sistema de captura, extrayendo la información de la Base de Datos de SIMA.

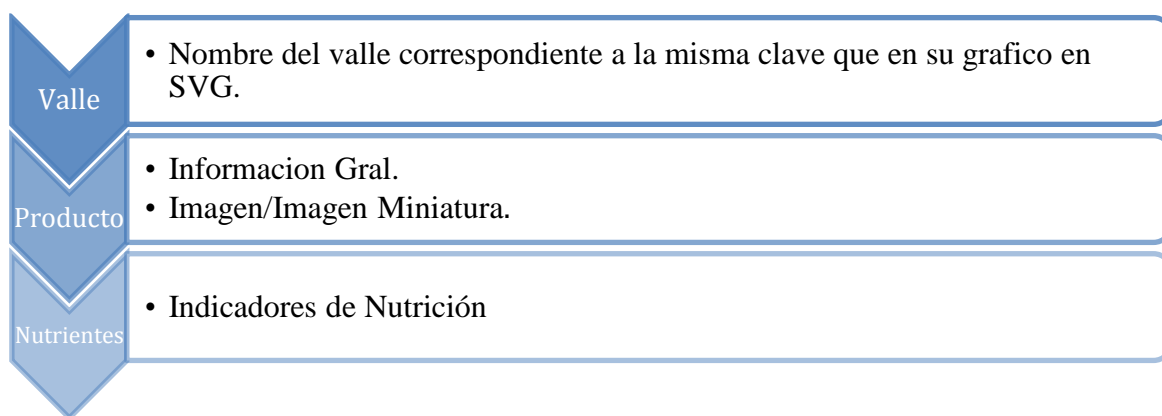


Figura 48 Jerarquía de contenido en la información de los productos.
Fuente elaboración propia.

```
content.xml
1 var content = {
2   "VMexicali" : [
3     { "Nombre" : "Alfalfa",
22    { "Nombre" : "Algodón Hueso",
41    { "Nombre" : "Carne Boylins",
60    { "Nombre" : "Cebolla",
79    { "Nombre" : "Espinaca", "Nombre" : "Uva",
98    { "Nombre" : "Lechuga", "imgUrl" : "pUva.png",
117   { "Nombre" : "Trépano", "imgUrlMini": "mini_uva.png",
136   { "Nombre" : "Zanahoria", "NombreC": "Vitis vinifera",
155   }, "vTecate" : [ "Descripcion" : 'Fruta. Baya de forma esférica que se agrupa en
156   { "Nombre" : "Lechuga", "Consumo" : 'Consumo Humano: En fresco, deshidratadas o secas; so
175   }, "vPalmas" : [ "Nutrientes" : { "Azucares": "15.48g",
176   { "Nombre" : "Lechuga", "Calcio": "17mg",
195   { "Nombre" : "Huevo", "Calorias": "70.3Kcal",
214   }, "vRosarito" : [ "Fibra": "0.8g",
215   { "Nombre" : "Cilantro", "Grasa_Total": "0.16g",
234   }, "vGuadalupe" : [ "Magnesio": "10mg",
235   { "Nombre" : "Uva", "Potasio": "250mg",
254   }, "vManeadero" : [ "Proteinas": "0.72g",
255   { "Nombre" : "Lechuga", "Sodio": "2mg",
274   { "Nombre" : "Cilantro", "vitamina_A": "1mg",
293   }, "vColonet" : [ "vitamina_B6": "0.11mg"
294   { "Nombre" : "Chile",
313   }, "vSanQuintín" : [
```

Figura 49 Representación archivo contenido Productos.
Fuente elaboración propia.

En la siguiente figura se puede observar el contenedor que se mostrara para detallar la información de los productos agropecuarios alojados en el mapa multimedia., dicho contenedor será invocado a manera de pantalla emergente al seleccionar algún valle y dar clic sobre el dentro del mapa. Este contenedor contendrá un listado de los productos que estén asociados en la base de datos a dicho valle.

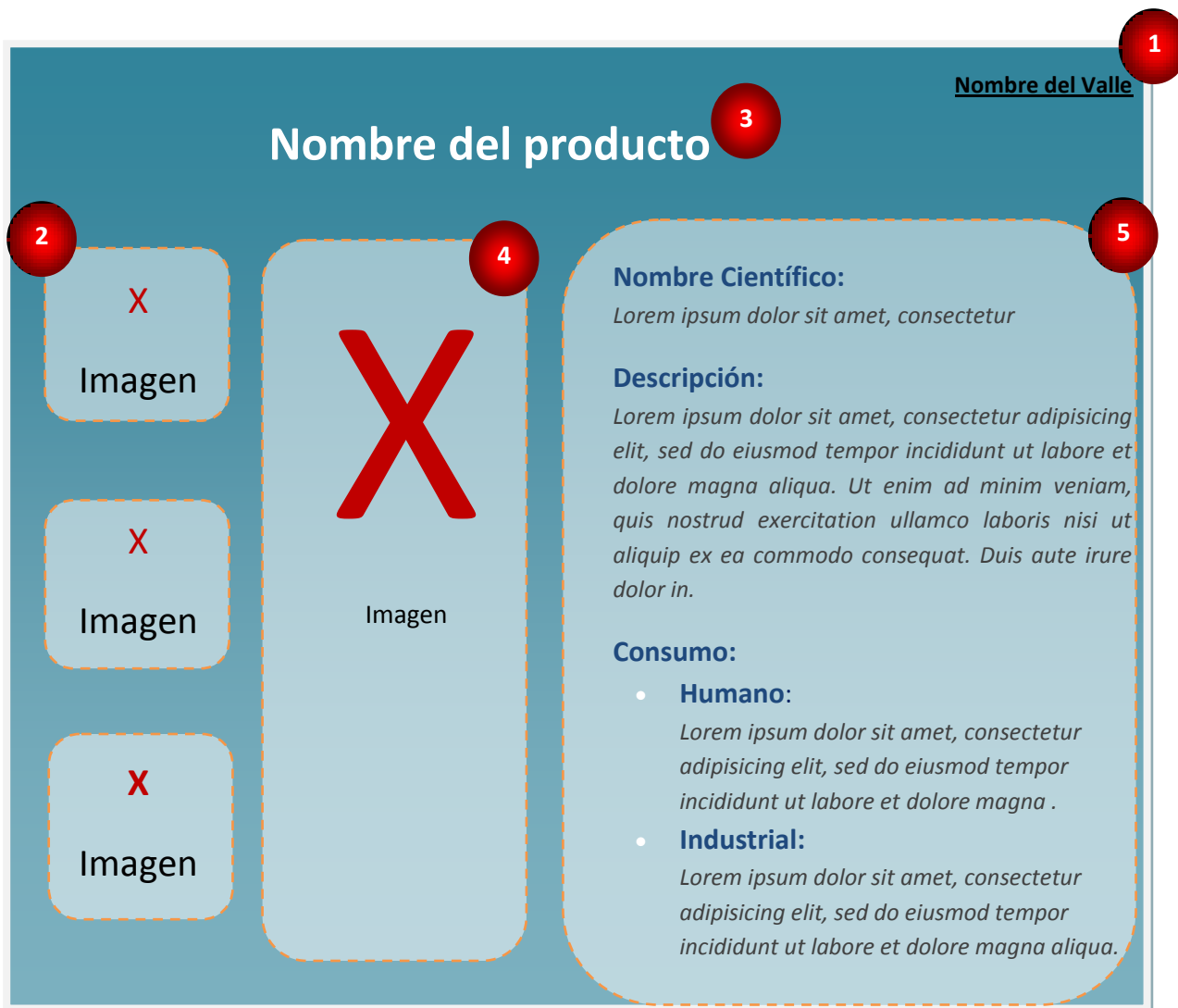


Figura 50 Gráficos Contenedor Productos y Detalle de Productos.
Fuente elaboración propia

Este contenedor se compone de 5 elementos principales:

- 1.- **Valle Seleccionado:** Es el valle que se selecciono en el mapa multimedia y al cual pertenece la información alojada en el contenedor.
- 2.- **Listado de Productos:** Es la lista de productos que pertenecen al valle seleccionado a manera de representaciones en miniatura de los productos.
- 3.- **Nombre del Producto:** Seria el nombre del producto seleccionado en el listado de productos.
- 4.- **Imagen del Producto:** Imagen representativas del producto seleccionado.
- 5.- **Detalle del Producto:** Es la descripción del producto seleccionado, “Nombre Científico”, “Descripción” y “Uso”.
- 6.- **Indicadores del Producto:** Es el listado de indicadores nutrimentales del producto seleccionado.

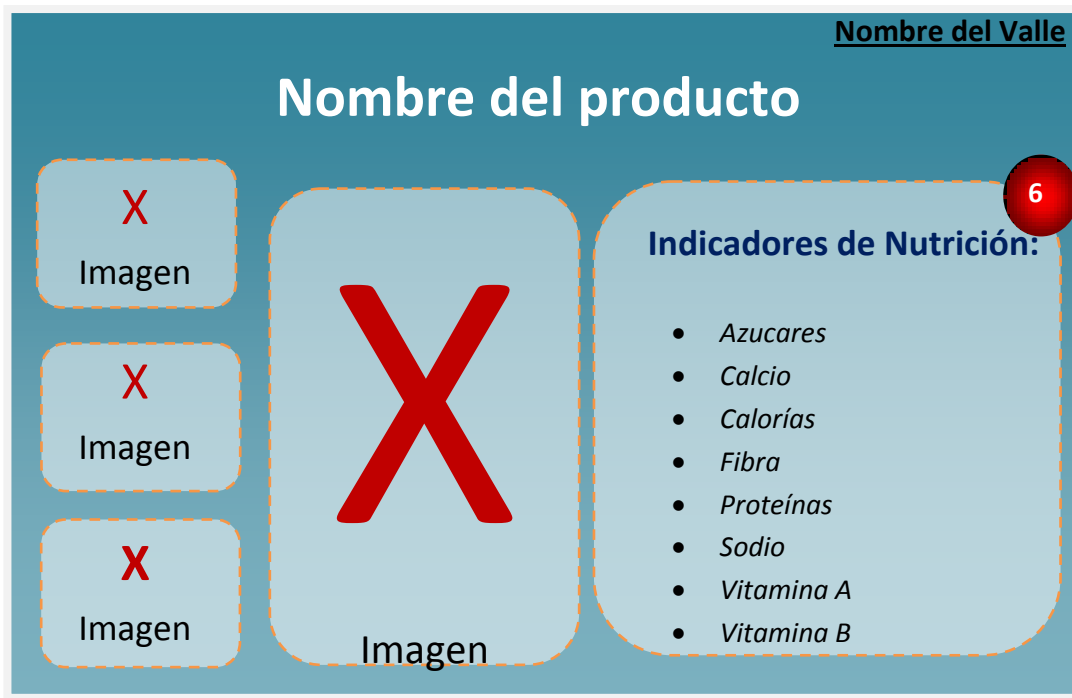


Figura 51 Gráficos Contenedor indicadores de nutrición de productos.
Fuente elaboración propia

Capítulo 5

Resultados

5.1 Implementación del sistema de captura

El sistema de captura de SIMA fue realizado como primera fase del proyecto, y este sistema estuvo publicado en el servidor de alumnos de licenciatura de la Facultad de Contabilidad y Administración de la UABC. Dicho servidor contaba ya con las características necesarias para cumplir con los requerimientos del sistema de captura.

En la figura 52 se representa la infraestructura de dicho servidor así como los servicios necesarios para el sistema de captura, esto es el motor de Base de Datos MySQL 5.5 y Lenguaje de programación web PHP 5.0.

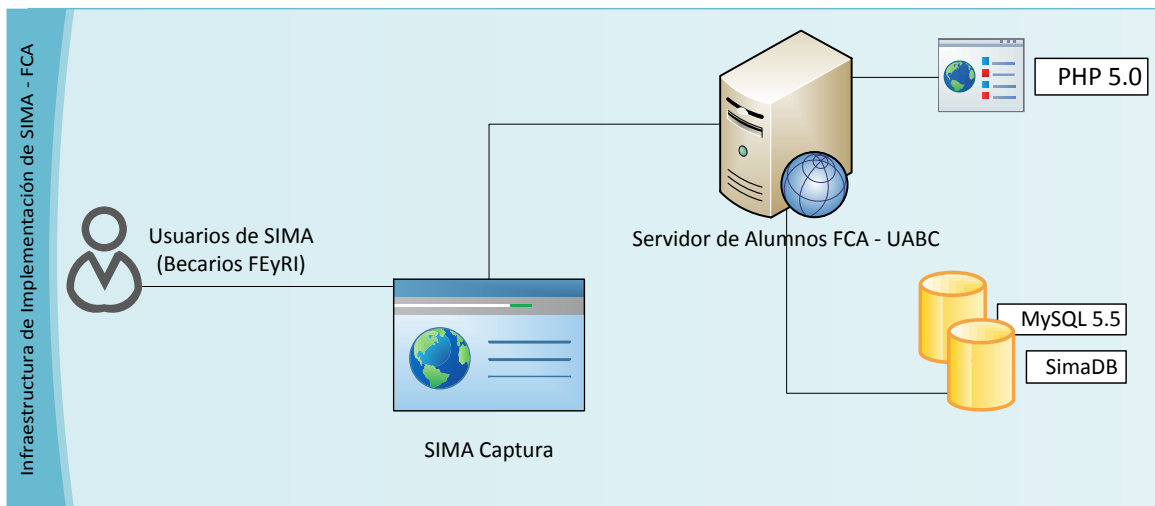


Figura 52 Infraestructura de la implementación en FCA.
Fuente elaboración propia.

Mientras el sistema estuvo alojado en el servidor de alumnos de FCA, fue posible realizar las pruebas del sistema en conjunto con el usuario, así como la captura de la información del conjunto de productos representativos de la Baja California.

Durante la fase de desarrollo del mapa multimedia y cuando el sistema de captura estaba completo se decidió migrar la base de datos y el sistema de captura al servidor de la Facultad de Economía y Relaciones Internacionales.

Para poder realizar la migración de la BD, se realizó la ejecución del script (figura 53) de respaldo del motor de base de datos MySQL.

```
1 $ mysqldump -u simadb -h localhost -p *****
2 simadb` > simadb.sql
```

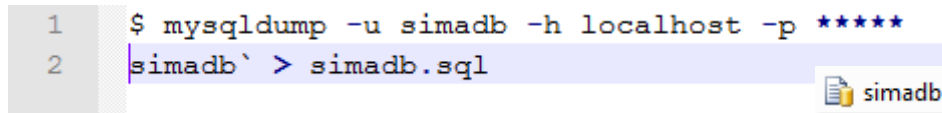


Figura 53 Respaldo de BD SIMA.
Fuente elaboración propia.

Como se puede observar en la figura 54 el servidor cuenta con los requerimientos mínimos necesarios para la funcionalidad del sistema de captura. Ya con un espacio definido para el proyecto SIMA, se decidió realizar una publicación del sistema del mapa multimedia.

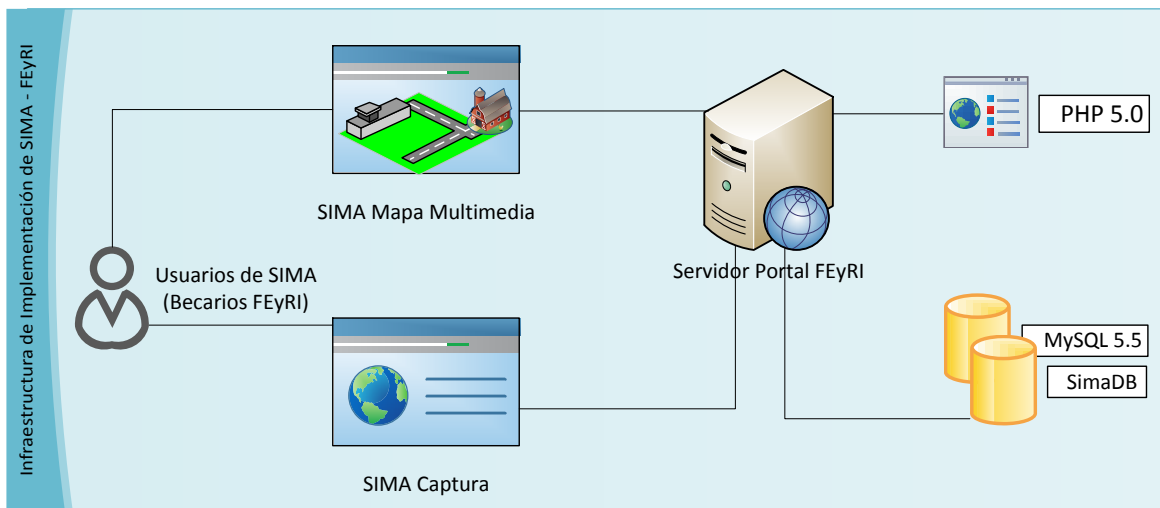


Figura 54 Infraestructura de la implementación en FEyRI.
Fuente elaboración propia.

Las ligas de las publicaciones dentro del servidor de la FEyRI son las siguientes:

Sistema de Captura: <http://feyri.tij.uabc.mx/ca/dyca/sima/>

Sistema de Multimedia: <http://feyri.tij.uabc.mx/ca/dyca/sima/mapa/>

1.1 Integración y pantallas de mapa multimedia

A continuación las pantallas finales del sistema multimedia.



Figura 55 Pantalla Inicial Mapa Multimedia.
Fuente elaboración propia.

El documento que se presenta en el apartado de “Acerca De SIMA” (Figura 56) es un boletín de divulgación que fue presentado en la Expo AgroBaja con sede en la ciudad de Mexicali, Baja California Marzo-2012.

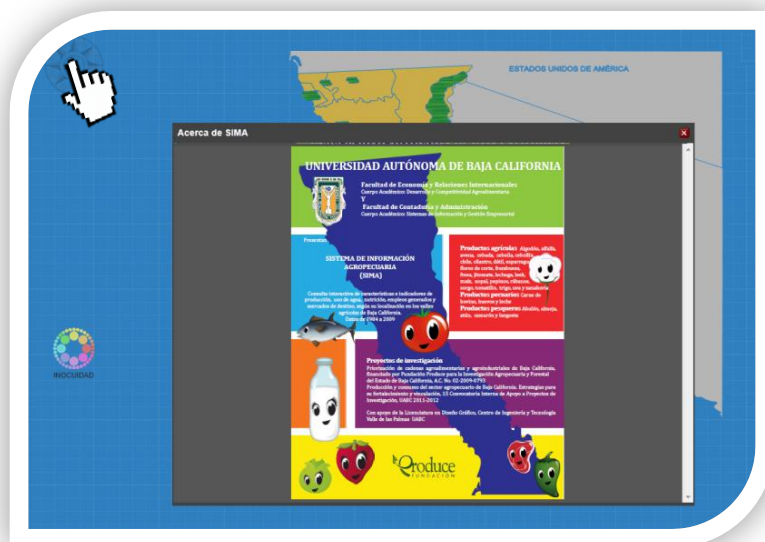


Figura 56 Pantalla “Acerca de SIMA”.
Fuente elaboración propia.

El documento de “Reglas de Inocuidad” (Figura 57) es un documento que de manera muy resumido pretende informar sobre las reglas básicas de inocuidad e higiene en los productos agropecuarios.

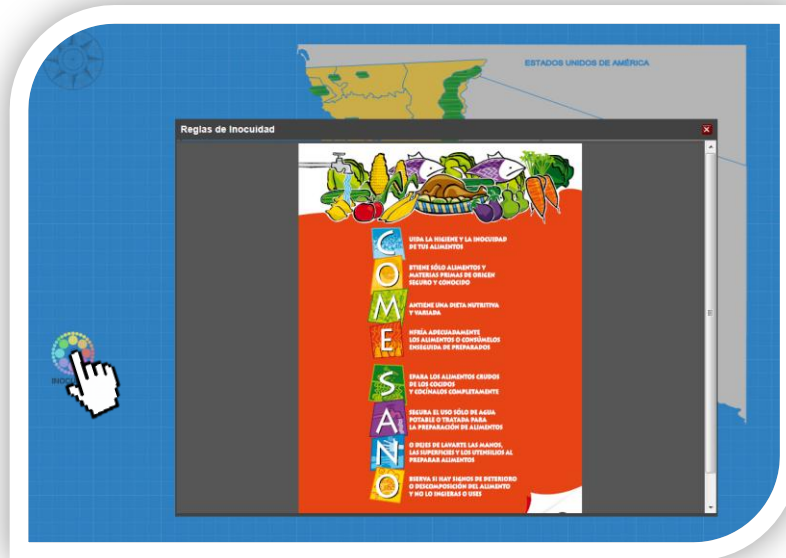


Figura 57 Pantalla “Reglas de Inocuidad”.
Fuente elaboración propia.



Figura 58 Pantalla selección de valles.
Fuente elaboración propia.

En la figura 58 se busca ilustrar como las áreas verdes denominados valles tiene la habilidad de pre-visualizar el valle al sobre el que se esta seleccionando con el movimiento del ratón.

En la figura 59 se ilustra el contenedor de los productos que le corresponden al valle seleccionado.

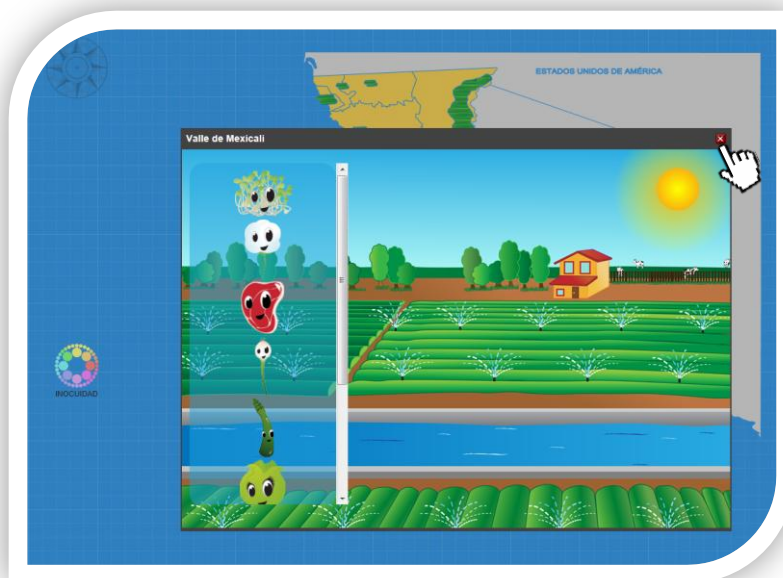


Figura 59 Pantalla listado de productos por valle.
Fuente elaboración propia.

Para poder ver a detalle cada producto será necesario seleccionar el producto dentro del listado que se encuentra en el extremo izquierdo de la pantalla contenedora.

En la figura 60 y 61 se muestra el detalle del producto así como los indicadores de nutrición respectivamente. En esta sección podremos encontrar información general del producto como su nombre científico, descripción y uso.

Para poder visualizar los indicadores de nutrición será necesario dar clic en la flecha naranja de navegación que se encuentra en la parte inferior derecha.

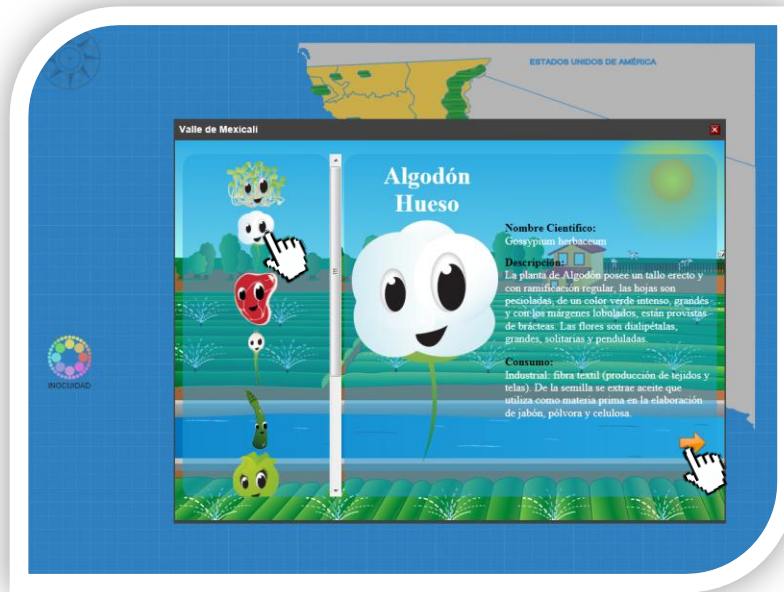


Figura 60 Pantalla detalle del producto.
Fuente elaboración propia.

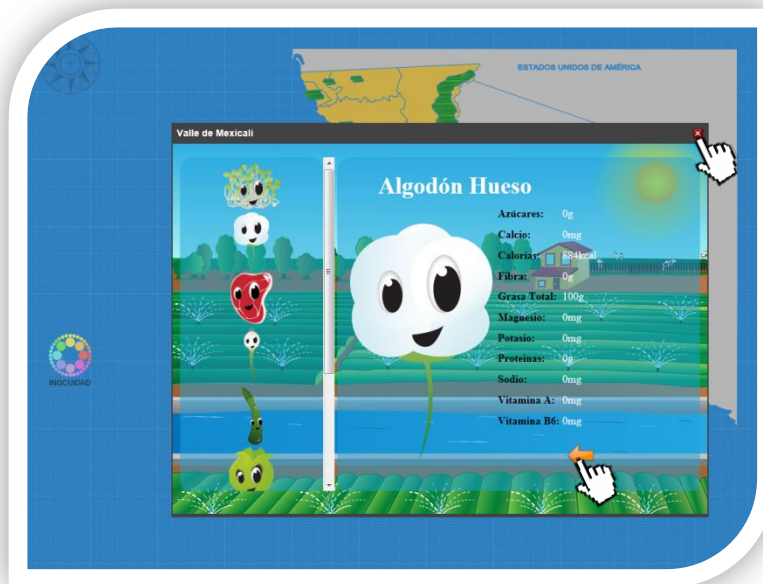


Figura 61 Pantalla indicadores de nutrición del producto.
Fuente elaboración propia.

Capítulo 6

Conclusiones

Del presente trabajo de tesis se desprenden una serie de conclusiones relevantes partiendo de que uno de los retos más importantes fue el entender el papel de la problemática agroalimentaria en un contexto social, es decir la falta de información en materia de inocuidad y el desconocimiento de la oferta de productos que existe en Baja California. Problemática que intentamos minimizar a través de el ejercicio de la divulgación de la oferta de productos agropecuarios enfocado a un universo de usuarios dentro del sector educativo básico, propiamente estudiantes del nivel educativo primaria.

Estando ya en contexto de la problemática el siguiente paso fue el de establecer las etapas del proyecto, los alcances y planes de trabajo, donde como primera etapa se estableció realizar la base de datos y desarrollar un sistema de captura ágil para la gestión de la información de los productos representativos del proyecto. Como segunda etapa del proyecto se estableció realizar un mapa multimedia con fines de divulgación de los productos, para el que se planteó considerar posibles apoyos en materia de contenido gráfico como fotografías, de igual modo la opinion de los expertos de marketing con mejores propuestas.

Durante la primera etapa del proyecto fue de suma importancia el poder establecer el modelo de base de datos de tal forma que sin importar el tipo de producto podríamos capturar de manera rápida e intuitiva la información, fue por esto que elegimos una interfaz a manera de grid de datos, en el cual es posible realizar desde una sola pantalla el mantenimiento de los datos (altas, bajas modificaciones y consultas).

Una vez que abordamos la segunda etapa del proyecto seguíamos sin tener algún recurso que nos asesorará en materia de diseño para afrontar la problemática de volver atractivo el sistema multimedia. Afortunadamente durante una presentación de avance del presente

trabajo de tesis, un docente del centro de ingeniería y tecnología de UABC unidad valle de las palmas se ofreció a establecer un vínculo con la coordinación de la carrera en diseño gráfico; De dicho vinculo se establecieron acuerdos para obtener el apoyo de siete alumnas del 9no. Semestre de la carrera Lic. Diseño Gráfico, mismas con quienes se estableció una dinámica de trabajo, donde los contenidos gráficos desarrollados para el presente trabajo de tesis pudieran contar como proyecto final para las alumnas y como ejercicio de experiencia profesional.

El trabajo de coordinación de este proyecto se centro entre otras actividades en poder interpretar las necesidades de divulgación del equipo de investigación de economía agroalimentaria y poder traducir estas a un contexto en el que las alumnas de diseño pudiesen interpretar y generar contenido gráfico de calidad y atractivo visualmente. Otra de las actividades de coordinación que constantemente se tuvieron que realizar fueron sesiones de entrenamiento para con los alumnos becarios que el grupo de investigación designó como capturistas, esto con el objetivo de lograr el buen uso del sistema de captura, y como apoyo para la realización de las pruebas del sistema.

Otro tema importante fue el reto que el sistema multimedia impuso para poder tener un diseño gráfico con el que todo el grupo de investigación estuviese de acuerdo, ya que buscábamos tener una representación animada de los productos y valles lo más cercano a la realidad sin descuidar la interacción humano-computadora. Es decir encontrar un equilibrio entre interactividad y el objetivo primordial que es el de la divulgación. Muchas fueron las sesiones de revisión que tuvimos con el grupo de investigación para poder establecer que los diseños gráficos de los productos y los valles fueran lo más apegado posible a la realidad. Constantemente durante dichas sesiones se daban cambios de decisiones sobre que productos eran los más representativos, que elementos visuales eran más importantes mostrar en los valles. En algunas sesiones lo más relevante fue el debate que se daba para no perder las características visualmente atractivas por mantener un fuerte apego a una imagen real de cierto valle.

Esta tesis sin duda alguna trascendió de aquella línea monodisciplinaria que suele ocurrir en los trabajos de tesis para volverse un trabajo de investigación de índole multidisciplinaria, es decir el poder involucrar a una problemática social y económica una estrategia de solución de tecnologías de la información multimedia, fue una de las características que volvió este trabajo muy interesante de conceptualizar y desarrollar.

Debido a que el sistema multimedia es el producto final del presente trabajo de tesis y ya que sólo se contemplaba como primera fase de un proyecto total, queda pendiente una segunda capa de interacción dentro del sistema multimedia, dicha capa contempla la divulgación de la información económica, es decir los indicadores económicos más relevantes de los productos agropecuarios de Baja California.

Al finalizar y evaluar el desarrollo de esta tesis podemos mencionar que hemos logrado cumplir con los objetivos planteados.

Bibliografía

- Allaire, J. (05 de 02 de 2010). *El futuro del Contenido Web, HTML5 vs Flash*. Recuperado el 10 de 06 de 2010, de techcrunch: <http://techcrunch.com/2010/02/05/the-future-of-web-content-html5-flash-mobile-apps/>
- Alvarez, G. A. (2010). *Introducción a HTML5*. Anaya Multimedia.
- Arroyo Cossío, A. (2011). El consumidor de Baja California. Preferencias por frutas y hortalizas frescas. *Tesis de maestría en ciencias económicas*.
- Avendaño, B. (2010). *Priorización de cadenas agroalimentarias y agroindustriales para Baja California*. Tijuana, BC. México.: PRODUCE Baja California.
- Bartolomé, P. A. (1994). *Multimedia interactivo y sus posibilidades en educación*. Recuperado el 1 de 12 de 2010, de Pixel-Bit. Revista de Medios y Educación, 1. 5-14.
- Bernal, A. (2010). *Metodología de la investigación*. Pearson Prentice Hall.
- Booch, G., & Rumbaugh, J. (2006). *EL LENGUAJE UNIFICADO DE MODELADO, UML 2. 0: GUIA DE USUARIO, APRENDA UML DIRECTAMENTE DE SUS CREADORES*. España: PEARSON.
- Boronczyk, T. (2009). *Php Y MySQL*. España: Anaya Multimedia - Anaya Interactiva.
- Bustamante, P. L. (2010). *Implementación de software educativo en las clases de ciencias naturales como parte del ciclo didáctico*. Recuperado el 06 de 10 de 2010, de <http://mx.globedia.com/implementacion-software-educativo-clases-ciencias-naturales-parte-ciclo-didactico>
- Castro, E. (2007). *HTML con XHTML y CSS*. Anaya Multimedia.
- Eguíluz, P. J. (2010). *Introducción a CSS*. Recuperado el 17 de 03 de 2011, de librosweb.es: <http://www.librosweb.es/css/>
- Eguíluz, P. J. (2010). *Introducción a Javascript*. Recuperado el 17 de 03 de 2011, de LibrosWeb.es: <http://www.librosweb.es/javascript/>
- Estrada, R. (2010). *Multimedia educativa: Entre fantasía y realidad*. Recuperado el 11 de 2010, de Universidad Virtual Anáhuac: http://www.sappiens.com/castellano/articulos.nsf/Formaci%C3%B3n_Virtual/Multimedia_educativa:_entre_fantas%C3%ADa_y_realidad/BCE8E42E14F80328C1256C9700470FAC!o pendocument

- FAO. (10 de 2003). Prioridad mundial al consumo de frutas y hortalizas. Recuperado el 11 de 2010, de La Organización de las Naciones Unidas para la Agricultura y la Alimentación: <http://www.fao.org/spanish/newsroom/focus/2003/fruitveg1.htm>
- Gayeski, D. M. (1992). *Making Sense of Multimedia*. Recuperado el 1 de 12 de 2010, de <http://www.dgayeski.com/dmgvita.pdf>.
- Guiza E., M. (2002). *Multimedia como herramienta en la educación*. Recuperado el 1 de 12 de 2010, de Congreso Internacional - Retos y Experiencias de la Universidad: http://www.congresoretosyexpectativas.udg.mx/Congreso%203/Mesa%205/Mesa5_18..pdf
- Huggers, E. (29 de 06 de 2010). *HTML5, open standards and the BBC*. Recuperado el 10 de 06 de 2010, de <http://apiblog.youtube.com/2010/06/flash-and-html5-tag.html>
- Lloréz, L., & Castro, M. L. (2008). *Didáctica de investigación: una propuesta formativa para el desarrollo de la creatividad y la inteligencia*. Universidad Autónoma de Baja California.
- Marton, P. (2010). Concepción pedagógica de sistemas de aprendizaje multimedia interactivo. *Perfiles Educativos, Universidad Autónoma de México*.
- OEIDRUS. (2010). *OEIDRUS*. Recuperado el 20 de 10 de 2010, de Oficina Estatal de Información para el Desarrollo Rural Sustentable: http://www.oeidrus-bc.gob.mx/oeidrus_bca/
- OMS. (2003). *OMS*. Recuperado el 10 de 2010, de Organización Mundial de la Salud: <http://www.fao.org/spanish/newsroom/news/2003/24439-es.html>
- Pacheco, M. L. (s.f.). *Internet en el proceso enseñanza aprendizaje*. Recuperado el 25 de 02 de 2011, de Monografías: <http://www.monografias.com/trabajos17/internet-aprendizaje/internet-aprendizaje.shtml>
- Pérez, T. J. (2002). *De la educación multimedia a la mediación del conocimiento: Estrategia de formación*. Recuperado el 12 de 2010, de Congreso de imagen y pedagogía: <http://www.gabinetecomunicacionyeducacion.com/files/adjuntos/De%20educ%20multimedia%20a%20mediacion%20conocimiento.pdf>
- Popkin, S. a. (2005). *Especificación UML*. Recuperado el 31 de 08 de 2011, de <http://www.popkin.com>: <http://www.hipertexto.info/documentos/uml.htm>
- PROFECO. (2010). *Consumo en México del sector pecuario*. Recuperado el 1 de 2010, de http://profeco.gob.mx/encuesta/brujula_2010/bol164.asp
- Rasmus, L. H. (24 de 06 de 2011). *Manual Php.Net*. Recuperado el 22 de 07 de 2011, de <http://php.net/manual/es/history.php.php>

SAGARPA. (2010). *SAGARPA*. Recuperado el 25 de 11 de 2010, de Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación: <http://www.sagarpa.gob.mx>

Salinas, J. M. (1996). *Multimedia en los procesos de enseñanza-aprendizaje: elemento de discusión*. Recuperado el 10 de 2010, de Ponencia en el Encuentro de Computación Educativa. Santiago de Chile, 2-4 mayo.: <http://www.uib.es/depart/gte/multimedia.html>

Sampieri, R. (2007). *Fundamentos de Metodología de la Investigación*. Mc Graw Hill.

Trippe, B. (25 de 10 de 2002). *SVG Design Classroom*. McGraw - Hill Education.

Capítulo 7

Anexos

7.1 Anexo 1 – Acerca de SIMA

El siguiente anexo corresponde al boletín de divulgación utilizado en el evento de Agrobaja 2011

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Economía y Relaciones Internacionales
Cuerpo Académico: Desarrollo y Competitividad Agroalimentaria
Y
Facultad de Contaduría y Administración
Cuerpo Académico: Sistemas de Información y Gestión Empresarial

Presentan

SISTEMA DE INFORMACIÓN AGROPECUARIA (SIMA)

Consulta interactiva de características e indicadores de producción, uso de agua, nutrición, empleos generados y mercados de destino, según su localización en los valles agrícolas de Baja California.
Datos de 1984 a 2009

Productos agrícolas Algodón, alfalfa, avena, cebada, cebolla, cebollín, chile, cilantro, dátil, espárrago, flores de corte, frambuesa, fresa, jitomate, lechuga, leek, maíz, nopal, pepinos, rábanos, sorgo, tomatillo, trigo, uva y zanahoria

Productos pecuarios Carne de bovino, huevos y leche

Productos pesqueros Abulón, almeja, atún, camarón y langosta


Proyectos de investigación
Priorización de cadenas agroalimentarias y agroindustriales de Baja California, financiado por Fundación Produce para la Investigación Agropecuaria y Forestal del Estado de Baja California, A.C. No. 02-2009-0793
Producción y consumo del sector agropecuario de Baja California. Estrategias para su fortalecimiento y vinculación, 15 Convocatoria Interna de Apoyo a Proyectos de Investigación, UABC 2011-2012

Con apoyo de la Licenciatura en Diseño Gráfico, Centro de Ingeniería y Tecnología Valle de las Palmas UABC

Produce FUNDACIÓN

7.2 Anexo 2 - Reglas de inocuidad

El siguiente anexo corresponde al boletín de divulgación de la FAO para la campaña de buenas prácticas de inocuidad alimentaria.



CUIDA LA HIGIENE Y LA INOCUIDAD DE TUS ALIMENTOS

OBTIENE SÓLO ALIMENTOS Y MATERIAS PRIMAS DE ORIGEN SEGURO Y CONOCIDO

MANTIENE UNA DIETA NUTRITIVA Y VARIADA

ENFRÍA ADECUADAMENTE LOS ALIMENTOS O CONSÚMELOS ENSEGUIDA DE PREPARADOS


SEPARA LOS ALIMENTOS CRUDOS DE LOS COCIDOS Y COCÍNALOS COMPLETAMENTE

ASEGURA EL USO SÓLO DE AGUA POTABLE O TRATADA PARA LA PREPARACIÓN DE ALIMENTOS

NO DEJES DE LAVARTE LAS MANOS, LAS SUPERFICIES Y LOS UTENSILIOS AL PREPARAR ALIMENTOS

OBSERVA SI HAY SIGNOS DE DETERIORO O DESCOMPOSICIÓN DEL ALIMENTO Y NO LO INGIERAS O USES

Oficina Regional de la FAO para América Latina y el Caribe
Grupo Inocuidad y Calidad de los Alimentos
www.rlc.fao.org - Teléfono: 56-2-923 2100



7.3 Anexo 3 - Tablas de descripción de casos de uso.

CASO DE USO		Inicio de Sesión	
CODIGO	UC01	Ref. Requerimiento	-
		Ref. Caso de Uso	-
Descripción general:			
Permite al usuario ingresar al sistema.			
Actores: Usuario			
Pre-Condiciones:			
<ul style="list-style-type: none"> • Carga del sitio exitosamente. • Se encuentre registrado el usuario en el sistema. 			
Pos-Condiciones:			
<ul style="list-style-type: none"> • Se mostrara el menú de opciones general al usuario. 			
FLUJO BASICO			
Código	Actor(es)	Sistema	
ESCENARIO: El usuario ingresara al sistema.			
FB01	El usuario ingresa los datos de usuario y contraseña respectivamente.		
FB02		El Sistema validara la información en Base de Datos.	
FB03		El sistema obtendrá los datos del usuario y su sesión será habilitada.	
FLUJO ALTERNATIVO			
Código	Actor(es)	Sistema	
ESCENARIO: -			
FLUJO ERROR			

Código	Actor(es)	Sistema
ESCENARIO: El usuario no recuerda su usuario o contraseña		
FE01	El usuario ingresa de forma incorrecta el usuario y/o contraseña.	
FE02		El sistema mostrara un mensaje indicando que el usuario o contraseña ingresados son incorrectos.

Tabla Descriptiva - Caso de Uso "Inicio de Sesión".

CASO DE USO		Cerrar Sesión	
CODIGO	UC02	Ref. Requerimiento	-
		Ref. Caso de Uso	-
Descripción general:			
Permite al usuario cerrar la sesión actual en el sistema.			
Actores: Usuario			
Pre-Condiciones:			
<ul style="list-style-type: none"> • Carga del sitio exitosamente. • Ingreso al sistema exitosamente. 			
Pos-Condiciones:			
<ul style="list-style-type: none"> • Se mostrará la pantalla de inicio de sesión. 			
FLUJO BASICO			
Código	Actor(es)	Sistema	
ESCENARIO: El selecciona el botón de cerrar sesión.			
FB01		El Sistema eliminará la sesión del usuario actual.	
FB02		El sistema obtendrá los datos del usuario	

		y su sesión será habilitada.
FLUJO ALTERNATIVO		
Código	Actor(es)	Sistema
ESCENARIO: -		
FLUJO ERROR		
Código	Actor(es)	Sistema
ESCENARIO: -		

Tabla Descriptiva - Caso de Uso "Cerrar Sesión".

CASO DE USO		Menú Productos	
CODIGO	UC03	Ref. Requerimiento	-
		Ref. Caso de Uso	-
Descripción general:			
Permite al usuario ingresar a las opciones del menú productos.			
Actores: Usuario			
Pre-Condiciones:			
<ul style="list-style-type: none"> • Ingreso al sistema exitosamente. 			
Pos-Condiciones:			
<ul style="list-style-type: none"> • Se mostrará las opciones del menú productos en el lado izquierdo de la pantalla. 			
FLUJO BASICO			
Código	Actor(es)	Sistema	
ESCENARIO: El selecciona el botón del menú "Productos".			

FB01		El Sistema generara la lista de opciones adecuadas al menú productos.
FB02		Se desplegaran las opciones en un espacio al lado izquierdo de la pantalla.
FLUJO ALTERNATIVO		
Código	Actor(es)	Sistema
ESCENARIO: -		
FLUJO ERROR		
Código	Actor(es)	Sistema
ESCENARIO: Fallo en la petición de respuesta del servidor.		
		El Navegador arrojará un error de respuesta no recibida.

Tabla Descriptiva - Caso de Uso "Menú Productos".

CASO DE USO	Menú Producción		
CODIGO	UC04	Ref. Requerimiento	-
		Ref. Caso de Uso	-
Descripción general:			
Permite al usuario ingresar a las opciones del menú producción.			
Actores: Usuario			
Pre-Condiciones:			
<ul style="list-style-type: none"> Ingreso al sistema exitosamente. 			
Pos-Condiciones:			
<ul style="list-style-type: none"> Se mostrara las opciones del menú productos en el lado izquierdo de la pantalla. 			

FLUJO BASICO		
Código	Actor(es)	Sistema
ESCENARIO: El selecciona el botón del menú “Producción”.		
FB01		El Sistema generara la lista de opciones adecuadas al menú producción.
FB02		Se desplegaran las opciones en un espacio al lado izquierdo de la pantalla.
FLUJO ALTERNATIVO		
Código	Actor(es)	Sistema
ESCENARIO: -		
FLUJO ERROR		
Código	Actor(es)	Sistema
ESCENARIO: Fallo en la petición de respuesta del servidor.		
		El Navegador arrojará un error de respuesta no recibida.

Tabla Descriptiva - Caso de Uso “Menú Producción”.

CASO DE USO		Menú Catálogos	
CODIGO	UC05	Ref. Requerimiento	-
		Ref. Caso de Uso	-
Descripción general:			
Permite al usuario ingresar a las opciones del menú catálogos.			
Actores: Usuario			
Pre-Condiciones:			

<ul style="list-style-type: none"> • Ingreso al sistema exitosamente. 		
Pos-Condiciones:		
<ul style="list-style-type: none"> • Se mostrara las opciones del menú productos en el lado izquierdo de la pantalla. 		
FLUJO BASICO		
Código	Actor(es)	Sistema
ESCENARIO: El selecciona el botón del menú “Catálogos”.		
FB01		El Sistema generara la lista de opciones adecuadas al menú catálogos.
FB02		Se desplegaran las opciones en un espacio al lado izquierdo de la pantalla.
FLUJO ALTERNATIVO		
Código	Actor(es)	Sistema
ESCENARIO: -		
FLUJO ERROR		
Código	Actor(es)	Sistema
ESCENARIO: Fallo en la petición de respuesta del servidor.		
		El Navegador arrojará un error de respuesta no recibida.

Tabla Descriptiva - Caso de Uso “Menú Catálogos”.

CASO DE USO		Productos	
CODIGO	UC06	Ref. Requerimiento	-
		Ref. Caso de Uso	UC03
Descripción general:			

La opción “Productos” mostrara una pantalla con un *grid* que permitirá gestionar la información sobre los productos en la base de datos.

Actores: **Usuario**

Pre-Condiciones:

- Ingreso al sistema exitosamente.
- Seleccionar el menú “productos”

Pos-Condiciones:

- Datos de los productos

FLUJO BASICO

Código	Actor(es)	Sistema
--------	-----------	---------

ESCENARIO: Agregar nuevo producto.

FB01	El usuario presionara el botón agregar nuevo registro	
FB02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas a los productos.
FB03	El usuario agregara los datos del nuevo producto.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario presionara el botón de agregar.	
FB06		El sistema agregara el nuevo registro en la base de datos.

ESCENARIO: Actualizar los productos existentes.

FB01	El usuario seleccionara el producto que quiera editar.	
FB02		El sistema volverá las columnas a modo

		editable y permitirá editar los datos.
FB03	El usuario realizara los cambios deseados en las columnas.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario seleccionara el botón de guardar los cambios.	
FB06		El sistema realizara los cambios en la base de datos para el producto editado.
ESCENARIO: Borrar algún producto seleccionado.		
FB01	El usuario seleccionara el producto que quiera borrar.	
FB02		El sistema mostrara un mensaje de confirmación
FB03	El usuario seleccionara aceptar para confirmar el borrado del producto	
FB04		El sistema borrara el registro en cuestión.
FLUJO ALTERNATIVO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo producto.		
FA01	El usuario presionara el botón agregar nuevo registro	
FA02		El sistema desplegará un espacio con las cajas de texto de las columnas relacionadas a los productos.
FA03	El usuario agregara los datos del nuevo producto.	

FB04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario presionara el botón de cancelar.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón agregar nuevo registro.
ESCENARIO: Actualizar los productos existentes.		
FA01	El usuario seleccionara el producto que quiera editar.	
FA02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FA03	El usuario realizara los cambios deseados en las columnas.	
FA04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario seleccionara el botón de cancelar los cambios.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón editar registro.
ESCENARIO: Borrar algún producto seleccionado.		
FB01	El usuario seleccionara el producto que quiera borrar.	
FA02		El sistema mostrara un mensaje de confirmación
FA03	El usuario seleccionara cancelar para evitar el borrado del producto	

FA04		El sistema ignorara la eliminación del registro.
FLUJO ERROR		
ESCENARIO: Datos incorrectos al agregar un nuevo producto.		
FE01	El usuario agregara los datos del nuevo producto.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.
ESCENARIO: Datos incorrectos al editar los datos de algún producto.		
FE01	El usuario realizara los cambios deseados en las columnas.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.

Tabla Descriptiva - Caso de Uso "Productos".

CASO DE USO		Nutrientes	
CODIGO	UC07	Ref. Requerimiento	-
		Ref. Caso de Uso	UC03
Descripción general:			
La opción "Nutrientes" mostrara una pantalla con un <i>grid</i> que permitirá gestionar la			

información sobre los nutrientes de los productos en la base de datos.		
Actores: Usuario		
Pre-Condiciones:		
<ul style="list-style-type: none"> • Ingreso al sistema exitosamente. • Seleccionar el menú “productos” 		
Pos-Condiciones:		
<ul style="list-style-type: none"> • Datos de los productos. 		
FLUJO BASICO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo nutriente de producto.		
FB01	El usuario presionara el botón agregar nuevo registro	
FB02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas a los nutrientes.
FB03	El usuario agregara los datos del nuevo nutriente.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario presionara el botón de agregar.	
FB06		El sistema agregara el nuevo registro en la base de datos.
ESCENARIO: Actualizar los nutrientes de producto existentes.		
FB01	El usuario seleccionara el nutriente que quiera editar.	
FB02		El sistema volverá las columnas a modo editable y permitirá editar los datos.

FB03	El usuario realizara los cambios deseados en las columnas.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario seleccionara el botón de guardar los cambios.	
FB06		El sistema realizara los cambios en la base de datos para el producto editado.
ESCENARIO: Borrar algún nutriente seleccionado.		
FB01	El usuario seleccionara el nutriente que quiera borrar.	
FB02		El sistema mostrara un mensaje de confirmación
FB03	El usuario seleccionara aceptar para confirmar el borrado.	
FB04		El sistema borrara el registro en cuestión.
FLUJO ALTERNATIVO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo producto.		
FA01	El usuario presionara el botón agregar nuevo registro	
FA02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas a los nutrientes.
FA03	El usuario agregara los datos del nuevo nutriente.	
FB04		El sistema valida las reglas de negocio en los datos.

FA05	El usuario presionara el botón de cancelar.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón agregar nuevo registro.
ESCENARIO: Actualizar los nutrientes existentes.		
FA01	El usuario seleccionara el nutriente que quiera editar.	
FA02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FA03	El usuario realizara los cambios deseados en las columnas.	
FA04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario seleccionara el botón de cancelar los cambios.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón editar registro.
ESCENARIO: Borrar algún nutriente de producto seleccionado.		
FB01	El usuario seleccionara el nutriente que quiera borrar.	
FA02		El sistema mostrara un mensaje de confirmación
FA03	El usuario seleccionara cancelar para evitar el borrado del registro	
FA04		El sistema ignorara la eliminación del registro.

FLUJO ERROR		
ESCENARIO: Datos incorrectos al agregar un nuevo nutriente.		
FE01	El usuario agregara los datos del nuevo nutriente.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.
ESCENARIO: Datos incorrectos al editar los datos de algún producto.		
FE01	El usuario realizara los cambios deseados en las columnas.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.

Tabla Descriptiva - Caso de Uso “Nutrientes”.

CASO DE USO		Atributos	
CODIGO	UC08	Ref. Requerimiento	-
		Ref. Caso de Uso	UC03
Descripción general:			
La opción “Atributos” mostrara una pantalla con un <i>grid</i> que permitirá gestionar la información sobre los atributos de los productos en la base de datos.			
Actores: Usuario			

Pre-Condiciones:		
<ul style="list-style-type: none"> • Ingreso al sistema exitosamente. • Seleccionar el menú “productos”. 		
Pos-Condiciones:		
<ul style="list-style-type: none"> • Datos de los atributos. 		
FLUJO BASICO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo atributo de producto.		
FB01	El usuario presionara el botón agregar nuevo registro	
FB02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas a los atributos.
FB03	El usuario agregara los datos del nuevo atributo de producto.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario presionara el botón de agregar.	
FB06		El sistema agregara el nuevo registro en la base de datos.
ESCENARIO: Actualizar los atributos de producto existentes.		
FB01	El usuario seleccionara el atributo que quiera editar.	
FB02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FB03	El usuario realizara los cambios deseados en las columnas.	

FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario seleccionara el botón de guardar los cambios.	
FB06		El sistema realizara los cambios en la base de datos para el atributo editado.
ESCENARIO: Borrar algún atributo seleccionado.		
FB01	El usuario seleccionara el atributo que quiera borrar.	
FB02		El sistema mostrara un mensaje de confirmación
FB03	El usuario seleccionara aceptar para confirmar el borrado del registro.	
FB04		El sistema borrará el producto en cuestión.
FLUJO ALTERNATIVO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo atributo.		
FA01	El usuario presionara el botón agregar nuevo registro	
FA02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas a los atributos de producto.
FA03	El usuario agregara los datos del nuevo atributo.	
FB04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario presionara el botón de	

	cancelar.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón agregar nuevo registro.
ESCENARIO: Actualizar los productos existentes.		
FA01	El usuario seleccionara el producto que quiera editar.	
FA02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FA03	El usuario realizara los cambios deseados en las columnas.	
FA04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario seleccionara el botón de cancelar los cambios.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón editar registro.
ESCENARIO: Borrar algún atributo seleccionado.		
FB01	El usuario seleccionara el atributo que quiera borrar.	
FA02		El sistema mostrara un mensaje de confirmación
FA03	El usuario seleccionara cancelar para evitar el borrado del atributo.	
FA04		El sistema ignorara la eliminación del registro.

FLUJO ERROR		
ESCENARIO: Datos incorrectos al agregar un nuevo atributo.		
FE01	El usuario agregara los datos del nuevo atributo de producto.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.
ESCENARIO: Datos incorrectos al editar los datos de algún atributo.		
FE01	El usuario realizara los cambios deseados en las columnas.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.

Tabla Descriptiva - Caso de Uso “Atributos”.

CASO DE USO		Producción	
CODIGO	UC09	Ref. Requerimiento	-
		Ref. Caso de Uso	UC04
Descripción general:			
La opción “Producción” mostrara una pantalla con un <i>grid</i> que permitirá gestionar la información sobre los años de la serie histórica del producto en la base de datos.			
Actores: Usuario			
Pre-Condiciones:			
<ul style="list-style-type: none"> Ingreso al sistema exitosamente. 			

<ul style="list-style-type: none"> • Seleccionar el menú “Producción” 		
Pos-Condiciones:		
<ul style="list-style-type: none"> • Datos de los productos . 		
FLUJO BASICO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo producto.		
FB01	El usuario presionara el botón agregar nuevo registro	
FB02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas a la serie histórica.
FB03	El usuario agregara los datos de la nueva serie histórica del producto.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario presionara el botón de agregar.	
FB06		El sistema agregara el nuevo registro en la base de datos.
ESCENARIO: Actualizar las series históricas existentes.		
FB01	El usuario seleccionara el producto que quiera editar.	
FB02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FB03	El usuario realizara los cambios deseados en las columnas.	
FB04		El sistema valida las reglas de negocio en los datos.

FB05	El usuario seleccionara el botón de guardar los cambios.	
FB06		El sistema realizara los cambios en la base de datos para la serie producto editado.
ESCENARIO: Borrar alguna serie histórica.		
FB01	El usuario seleccionara la serie que desea eliminar.	
FB02		El sistema mostrara un mensaje de confirmación
FB03	El usuario seleccionara aceptar para confirmar la eliminación de la serie.	
FB04		El sistema eliminara el registro en cuestión.
FLUJO ALTERNATIVO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nueva serie histórica.		
FA01	El usuario presionara el botón agregar nuevo registro.	
FA02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas a la serie histórica.
FA03	El usuario agregara los datos de la nueva serie histórica de producto.	
FB04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario presionara el botón de cancelar.	

FA06		El sistema regresa la pantalla al estado previo a presionar el botón agregar nuevo registro.
ESCENARIO: Actualizar las series históricas existentes.		
FA01	El usuario seleccionara la serie histórica que quiera editar.	
FA02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FA03	El usuario realizara los cambios deseados en las columnas.	
FA04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario seleccionara el botón de cancelar los cambios.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón editar registro.
ESCENARIO: Borrar alguna serie histórica seleccionado.		
FB01	El usuario seleccionara la serie histórica que quiera borrar.	
FA02		El sistema mostrara un mensaje de confirmación.
FA03	El usuario seleccionara cancelar para evitar el borrado de la serie histórica.	
FA04		El sistema ignorara la eliminación del registro.
FLUJO ERROR		

ESCENARIO: Datos incorrectos al agregar un nueva serie histórica.		
FE01	El usuario agregara los datos de la nueva serie histórica.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.
ESCENARIO: Datos incorrectos al editar los datos de alguna serie histórica.		
FE01	El usuario realizara los cambios deseados en las columnas.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.

Tabla Descriptiva - Caso de Uso “Producción”.

CASO DE USO		Historial Producción	
CODIGO	UC10	Ref. Requerimiento	-
		Ref. Caso de Uso	UC04
Descripción general:			
La opción “Historial de Producción” mostrara una pantalla muy parecida a una ficha técnica, que permitirá visualizar todo los datos que previamente se capturaron para cada producto en la base de datos.			
Actores: Usuario			
Pre-Condiciones:			
<ul style="list-style-type: none"> • Ingreso al sistema exitosamente. 			

<ul style="list-style-type: none"> • Seleccionar el menú “Producción” 		
Pos-Condiciones:		
<ul style="list-style-type: none"> • Datos de los productos 		
FLUJO BASICO		
Código	Actor(es)	Sistema
ESCENARIO: Visualizar Resumen del producto.		
FB01	El usuario selecciona un producto de la lista de productos.	
FB02		El sistema desplegará los datos generales del producto así como su tabla de nutrientes y serie histórica.

Tabla Descriptiva - Caso de Uso “Historial de Producción”.

CASO DE USO		Catalogo Atributos	
CODIGO	UC11	Ref. Requerimiento	-
		Ref. Caso de Uso	UC05
Descripción general:			
<p>La opción “Catalogo Atributos” mostrara una pantalla con un <i>grid</i> que permitirá gestionar la información sobre los atributos del producto en la base de datos.</p>			
Actores: Usuario			
Pre-Condiciones:			
<ul style="list-style-type: none"> • Ingreso al sistema exitosamente. • Seleccionar el menú “Catálogos” 			
Pos-Condiciones:			
<ul style="list-style-type: none"> • Atributos de los productos . 			

FLUJO BASICO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo Atributo.		
FB01	El usuario presionara el botón agregar nuevo registro	
FB02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas a los atributos.
FB03	El usuario agregara los datos del nuevo atributo del producto.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario presionara el botón de agregar.	
FB06		El sistema agregara el nuevo registro en la base de datos.
ESCENARIO: Actualizar los atributos existentes.		
FB01	El usuario seleccionara el atributo que quiera editar.	
FB02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FB03	El usuario realizara los cambios deseados en las columnas.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario seleccionara el botón de guardar los cambios.	
FB06		El sistema realizara los cambios en la base de datos para la serie producto

		editado.
ESCENARIO: Borrar algún atributo registrado.		
FB01	El usuario seleccionara el atributo que desea eliminar.	
FB02		El sistema mostrara un mensaje de confirmación
FB03	El usuario seleccionara aceptar para confirmar la eliminación del atributo.	
FB04		El sistema eliminara el registro en cuestión.
FLUJO ALTERNATIVO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo atributo.		
FA01	El usuario presionara el botón agregar nuevo registro.	
FA02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas al atributo.
FA03	El usuario agregara los datos del nuevo atributo.	
FB04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario presionara el botón de cancelar.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón agregar nuevo registro.

ESCENARIO: Actualizar los atributos existentes.		
FA01	El usuario seleccionara los atributos que quiera editar.	
FA02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FA03	El usuario realizara los cambios deseados en las columnas.	
FA04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario seleccionara el botón de cancelar los cambios.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón editar registro.
ESCENARIO: Borrar algún atributo seleccionado.		
FB01	El usuario seleccionara el atributo que quiera borrar.	
FA02		El sistema mostrara un mensaje de confirmación.
FA03	El usuario seleccionara cancelar para evitar el borrado del atributo.	
FA04		El sistema ignorara la eliminación del registro.
FLUJO ERROR		
ESCENARIO: Datos incorrectos al agregar un nuevo atributo.		
FE01	El usuario agregara los datos del nuevo atributo.	
FE02		El sistema valida las reglas de negocio en

		los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.
ESCENARIO: Datos incorrectos al editar los datos de algún atributo.		
FE01	El usuario realizara los cambios deseados en las columnas.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.

Tabla Descriptiva - Caso de Uso "Catalogo Atributos".

CASO DE USO		Catalogo Tipo de Producto	
CODIGO	UC12	Ref. Requerimiento	-
		Ref. Caso de Uso	UC05
<p>Descripción general:</p> <p>La opción "Catalogo Tipo Producto" mostrara una pantalla con un <i>grid</i> que permitirá gestionar la información sobre los atributos del producto en la base de datos.</p>			
Actores: Usuario			
<p>Pre-Condiciones:</p> <ul style="list-style-type: none"> • Ingreso al sistema exitosamente. • Seleccionar el menú "Catálogos" 			
<p>Pos-Condiciones:</p> <ul style="list-style-type: none"> • Atributos de los productos . 			
FLUJO BASICO			

Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo Tipo Producto.		
FB01	El usuario presionara el botón agregar nuevo registro.	
FB02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas a los tipos de producto.
FB03	El usuario agregara los datos del nuevo tipo de producto.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario presionara el botón de agregar.	
FB06		El sistema agregara el nuevo registro en la base de datos.
ESCENARIO: Actualizar los tipos de producto existentes.		
FB01	El usuario seleccionara el tipo de producto que quiera editar.	
FB02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FB03	El usuario realizara los cambios deseados en las columnas.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario seleccionara el botón de guardar los cambios.	
FB06		El sistema realizara los cambios en la base de datos para el tipo de producto editado.

ESCENARIO: Borrar algún tipo de producto registrado.		
FB01	El usuario seleccionara el atributo que desea eliminar.	
FB02		El sistema mostrara un mensaje de confirmación
FB03	El usuario seleccionara aceptar para confirmar la eliminación del tipo de producto.	
FB04		El sistema eliminara el registro en cuestión.
FLUJO ALTERNATIVO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo tipo de producto.		
FA01	El usuario presionara el botón agregar nuevo registro.	
FA02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas al tipo de producto.
FA03	El usuario agregara los datos del nuevo tipo de producto.	
FB04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario presionara el botón de cancelar.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón agregar nuevo registro.

ESCENARIO: Actualizar el tipos de producto existente.		
FA01	El usuario seleccionara el tipo de producto que quiera editar.	
FA02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FA03	El usuario realizara los cambios deseados en las columnas.	
FA04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario seleccionara el botón de cancelar los cambios.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón editar registro.
ESCENARIO: Borrar algún tipo de producto seleccionado.		
FB01	El usuario seleccionara el tipo de producto que quiera borrar.	
FA02		El sistema mostrara un mensaje de confirmación.
FA03	El usuario seleccionara cancelar para evitar el borrado del tipo de producto.	
FA04		El sistema ignorara la eliminación del registro.
FLUJO ERROR		
ESCENARIO: Datos incorrectos al agregar un nuevo tipo de producto.		
FE01	El usuario agregara los datos del nuevo tipo de producto.	
FE02		El sistema valida las reglas de negocio en

		los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.
ESCENARIO: Datos incorrectos al editar los datos de algún tipo de producto.		
FE01	El usuario realizara los cambios deseados en las columnas.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.

Tabla Descriptiva - Caso de Uso "Catalogo Tipo de Productos".

CASO DE USO		Catalogo Municipios	
CODIGO	UC13	Ref. Requerimiento	-
		Ref. Caso de Uso	UC05
<p>Descripción general:</p> <p>La opción "Catalogo Municipios" mostrara una pantalla con un <i>grid</i> que permitirá gestionar la información sobre los municipios del producto en la base de datos.</p>			
Actores: Usuario			
<p>Pre-Condiciones:</p> <ul style="list-style-type: none"> • Ingreso al sistema exitosamente. • Seleccionar el menú "Catálogos" 			
<p>Pos-Condiciones:</p> <ul style="list-style-type: none"> • Tipo de productos en la BD. 			
FLUJO BASICO			

Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo Municipio.		
FB01	El usuario presionara el botón agregar nuevo registro.	
FB02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas al municipio.
FB03	El usuario agregara los datos del nuevo municipio.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario presionara el botón de agregar.	
FB06		El sistema agregara el nuevo registro en la base de datos.
ESCENARIO: Actualizar los municipios existentes.		
FB01	El usuario seleccionara el municipio que quiera editar.	
FB02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FB03	El usuario realizara los cambios deseados en las columnas.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario seleccionara el botón de guardar los cambios.	
FB06		El sistema realizara los cambios en la base de datos para el municipio editado.

ESCENARIO: Borrar algún municipio registrado.		
FB01	El usuario seleccionara el municipio que desea eliminar.	
FB02		El sistema mostrara un mensaje de confirmación
FB03	El usuario seleccionara aceptar para confirmar la eliminación del municipio.	
FB04		El sistema eliminara el registro en cuestión.
FLUJO ALTERNATIVO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo municipio.		
FA01	El usuario presionara el botón agregar nuevo registro.	
FA02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas al municipio.
FA03	El usuario agregara los datos del nuevo municipio.	
FB04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario presionara el botón de cancelar.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón agregar nuevo registro.
ESCENARIO: Actualizar un municipio existente.		
FA01	El usuario seleccionara el municipio	

	que quiera editar.	
FA02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FA03	El usuario realizara los cambios deseados en las columnas.	
FA04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario seleccionara el botón de cancelar los cambios.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón editar registro.
ESCENARIO: Borrar algún municipio seleccionado.		
FB01	El usuario seleccionara el municipio que quiera borrar.	
FA02		El sistema mostrara un mensaje de confirmación.
FA03	El usuario seleccionara cancelar para evitar el borrado del municipio.	
FA04		El sistema ignorara la eliminación del registro.
FLUJO ERROR		
ESCENARIO: Datos incorrectos al agregar un nuevo municipio.		
FE01	El usuario agregara los datos del nuevo municipio.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que

		tienen datos incorrectos.
ESCENARIO: Datos incorrectos al editar los datos de algún municipio.		
FE01	El usuario realizara los cambios deseados en las columnas.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.

Tabla Descriptiva – Caso de Uso “Catalogo Municipios”.

CASO DE USO		Catalogo Clima	
CODIGO	UC14	Ref. Requerimiento	-
		Ref. Caso de Uso	UC05
<p>Descripción general:</p> <p>La opción “Catalogo Clima” mostrara una pantalla con un <i>grid</i> que permitirá gestionar la información sobre los climas del producto en la base de datos.</p>			
Actores: Usuario			
Pre-Condiciones:			
<ul style="list-style-type: none"> • Ingreso al sistema exitosamente. • Seleccionar el menú “Catálogos” 			
Pos-Condiciones:			
<ul style="list-style-type: none"> • Clima de los productos . 			
FLUJO BASICO			
Código	Actor(es)	Sistema	
ESCENARIO: Agregar nuevo clima.			

FB01	El usuario presionara el botón agregar nuevo registro.	
FB02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas al clima.
FB03	El usuario agregara los datos del nuevo clima.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario presionara el botón de agregar.	
FB06		El sistema agregara el nuevo registro en la base de datos.
ESCENARIO: Actualizar los climas existentes.		
FB01	El usuario seleccionara el clima que quiera editar.	
FB02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FB03	El usuario realizara los cambios deseados en las columnas.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario seleccionara el botón de guardar los cambios.	
FB06		El sistema realizara los cambios en la base de datos para el clima editado.
ESCENARIO: Borrar algún clima registrado.		
FB01	El usuario seleccionara el clima que	

	desea eliminar.	
FB02		El sistema mostrara un mensaje de confirmación
FB03	El usuario seleccionara aceptar para confirmar la eliminación del clima.	
FB04		El sistema eliminara el registro en cuestión.
FLUJO ALTERNATIVO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo clima.		
FA01	El usuario presionara el botón agregar nuevo registro.	
FA02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas al clima.
FA03	El usuario agregara los datos del nuevo clima.	
FB04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario presionara el botón de cancelar.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón agregar nuevo registro.
ESCENARIO: Actualizar un clima existente.		
FA01	El usuario seleccionara el clima que quiera editar.	
FA02		El sistema volverá las columnas a modo

		editable y permitirá editar los datos.
FA03	El usuario realizara los cambios deseados en las columnas.	
FA04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario seleccionara el botón de cancelar los cambios.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón editar registro.
ESCENARIO: Borrar algún clima seleccionado.		
FB01	El usuario seleccionara el clima que quiera borrar.	
FA02		El sistema mostrara un mensaje de confirmación.
FA03	El usuario seleccionara cancelar para evitar el borrado del clima.	
FA04		El sistema ignorara la eliminación del registro.
FLUJO ERROR		
ESCENARIO: Datos incorrectos al agregar un nuevo clima.		
FE01	El usuario agregara los datos del nuevo clima.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.

ESCENARIO: Datos incorrectos al editar los datos de algún clima.		
FE01	El usuario realizara los cambios deseados en las columnas.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.

Tabla Descriptiva - Caso de Uso "Catalogo Clima".

CASO DE USO		Catalogo Indicadores de Nutrición	
CODIGO	UC15	Ref. Requerimiento	-
		Ref. Caso de Uso	UC05
Descripción general:			
La opción "Catalogo Indicadores de Nutrición" mostrara una pantalla con un <i>grid</i> que permitirá gestionar la información sobre los indicadores de nutrición del producto en la base de datos.			
Actores: Usuario			
Pre-Condiciones:			
<ul style="list-style-type: none"> • Ingreso al sistema exitosamente. • Seleccionar el menú "Catálogos" 			
Pos-Condiciones:			
<ul style="list-style-type: none"> • Indicadores de Nutrición de los productos . 			
FLUJO BASICO			
Código	Actor(es)		Sistema
ESCENARIO: Agregar nuevo indicador de nutrición.			

FB01	El usuario presionara el botón agregar nuevo registro.	
FB02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas al indicador de nutrición.
FB03	El usuario agregara los datos del nuevo indicador de nutrición.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario presionara el botón de agregar.	
FB06		El sistema agregara el nuevo registro en la base de datos.

ESCENARIO: Actualizar los indicadores de nutrición existentes.

FB01	El usuario seleccionara el indicador de nutrición que quiera editar.	
FB02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FB03	El usuario realizara los cambios deseados en las columnas.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario seleccionara el botón de guardar los cambios.	
FB06		El sistema realizara los cambios en la base de datos para el indicador de nutrición editado.

ESCENARIO: Borrar algún indicador de nutrición registrado.

FB01	El usuario seleccionara el indicador de nutrición que desea eliminar.	
FB02		El sistema mostrara un mensaje de confirmación
FB03	El usuario seleccionara aceptar para confirmar la eliminación del indicador de nutrición.	
FB04		El sistema eliminara el registro en cuestión.
FLUJO ALTERNATIVO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo indicador de nutrición.		
FA01	El usuario presionara el botón agregar nuevo registro.	
FA02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas al indicador de nutrición.
FA03	El usuario agregara los datos del nuevo indicador de nutrición.	
FB04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario presionara el botón de cancelar.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón agregar nuevo registro.
ESCENARIO: Actualizar un indicador de nutrición existente.		
FA01	El usuario seleccionara el indicador de	

	nutrición que quiera editar.	
FA02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FA03	El usuario realizara los cambios deseados en las columnas.	
FA04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario seleccionara el botón de cancelar los cambios.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón editar registro.
ESCENARIO: Borrar algún indicador de nutrición seleccionado.		
FB01	El usuario seleccionara el indicador de nutrición que quiera borrar.	
FA02		El sistema mostrara un mensaje de confirmación.
FA03	El usuario seleccionara cancelar para evitar el borrado del indicador de nutrición.	
FA04		El sistema ignorara la eliminación del registro.
FLUJO ERROR		
ESCENARIO: Datos incorrectos al agregar un nuevo indicador de nutrición.		
FE01	El usuario agregara los datos del nuevo indicador de nutrición.	
FE02		El sistema valida las reglas de negocio en los datos.

FE03		El sistema muestra las columnas que tienen datos incorrectos.
ESCENARIO: Datos incorrectos al editar los datos de algún indicador de nutrición.		
FE01	El usuario realizara los cambios deseados en las columnas.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.

Tabla Descriptiva - Caso de Uso "Catalogo Indicadores de Nutrición".

CASO DE USO		Catalogo Localidades	
CODIGO	UC16	Ref. Requerimiento	-
		Ref. Caso de Uso	UC05
<p>Descripción general:</p> <p>La opción "Catalogo Localidades" mostrara una pantalla con un <i>grid</i> que permitirá gestionar la información sobre las localidades del producto en la base de datos.</p>			
Actores: Usuario			
<p>Pre-Condiciones:</p> <ul style="list-style-type: none"> • Ingreso al sistema exitosamente. • Seleccionar el menú "Catálogos" 			
<p>Pos-Condiciones:</p> <ul style="list-style-type: none"> • Localidades de los productos . 			
FLUJO BASICO			
Código	Actor(es)		Sistema

ESCENARIO: Agregar nueva localidad.		
FB01	El usuario presionara el botón agregar nuevo registro.	
FB02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas a la localidad.
FB03	El usuario agregara los datos de la nueva localidad.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario presionara el botón de agregar.	
FB06		El sistema agregara el nuevo registro en la base de datos.
ESCENARIO: Actualizar las localidades existentes.		
FB01	El usuario seleccionara el indicador de nutrición que quiera editar.	
FB02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FB03	El usuario realizara los cambios deseados en las columnas.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario seleccionara el botón de guardar los cambios.	
FB06		El sistema realizara los cambios en la base de datos para la localidad editada.
ESCENARIO: Borrar alguna localidad registrada.		

FB01	El usuario seleccionara la localidad que desea eliminar.	
FB02		El sistema mostrara un mensaje de confirmación
FB03	El usuario seleccionara aceptar para confirmar la eliminación de la localidad.	
FB04		El sistema eliminara el registro en cuestión.
FLUJO ALTERNATIVO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nueva localidad.		
FA01	El usuario presionara el botón agregar nuevo registro.	
FA02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas a la localidad.
FA03	El usuario agregara los datos de la nueva localidad.	
FB04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario presionara el botón de cancelar.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón agregar nuevo registro.
ESCENARIO: Actualizar un localidad existente.		
FA01	El usuario seleccionara la localidad que	

	quiera editar.	
FA02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FA03	El usuario realizara los cambios deseados en las columnas.	
FA04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario seleccionara el botón de cancelar los cambios.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón editar registro.
ESCENARIO: Borrar alguna localidad seleccionado.		
FB01	El usuario seleccionara la localidad que quiera borrar.	
FA02		El sistema mostrara un mensaje de confirmación.
FA03	El usuario seleccionara cancelar para evitar el borrado de la localidad.	
FA04		El sistema ignorara la eliminación del registro.
FLUJO ERROR		
ESCENARIO: Datos incorrectos al agregar una nueva localidad.		
FE01	El usuario agregara los datos de la nueva localidad.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que

		tienen datos incorrectos.
ESCENARIO: Datos incorrectos al editar alguna localidad.		
FE01	El usuario realizara los cambios deseados en las columnas.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.

Tabla Descriptiva - Caso de Uso "Catalogo Localidades".

CASO DE USO		Catalogo Indicadores Económicos	
CODIGO	UC17	Ref. Requerimiento	-
		Ref. Caso de Uso	UC05
<p>Descripción general:</p> <p>La opción "Catalogo Indicadores Económicos" mostrara una pantalla con un <i>grid</i> que permitirá gestionar la información sobre los indicadores económicos del producto en la base de datos.</p>			
Actores: Usuario			
<p>Pre-Condiciones:</p> <ul style="list-style-type: none"> • Ingreso al sistema exitosamente. • Seleccionar el menú "Catálogos" 			
<p>Pos-Condiciones:</p> <ul style="list-style-type: none"> • Indicadores económicos de los productos . 			
FLUJO BASICO			
Código	Actor(es)		Sistema

ESCENARIO: Agregar nuevo indicador económico.		
FB01	El usuario presionara el botón agregar nuevo registro.	
FB02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas al indicador económico.
FB03	El usuario agregara los datos del nuevo indicador económico.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario presionara el botón de agregar.	
FB06		El sistema agregara el nuevo registro en la base de datos.
ESCENARIO: Actualizar los indicadores económicos existentes.		
FB01	El usuario seleccionara el indicador económico que quiera editar.	
FB02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FB03	El usuario realizara los cambios deseados en las columnas.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario seleccionara el botón de guardar los cambios.	
FB06		El sistema realizará los cambios en la base de datos para el indicador económico editado.

ESCENARIO: Borrar algún indicador económico registrado.		
FB01	El usuario seleccionara el indicador económico que desea eliminar.	
FB02		El sistema mostrara un mensaje de confirmación
FB03	El usuario seleccionara aceptar para confirmar la eliminación del indicador económico.	
FB04		El sistema eliminara el registro en cuestión.
FLUJO ALTERNATIVO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo indicador económico.		
FA01	El usuario presionara el botón agregar nuevo registro.	
FA02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas al indicador económico.
FA03	El usuario agregara los datos del nuevo indicador económico.	
FB04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario presionara el botón de cancelar.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón agregar nuevo registro.
ESCENARIO: Actualizar un indicador económico existente.		

FA01	El usuario seleccionara el indicador económico que quiera editar.	
FA02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FA03	El usuario realizara los cambios deseados en las columnas.	
FA04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario seleccionara el botón de cancelar los cambios.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón editar registro.

ESCENARIO: Borrar algún indicador económico seleccionado.

FB01	El usuario seleccionara el indicador económico que quiera borrar.	
FA02		El sistema mostrara un mensaje de confirmación.
FA03	El usuario seleccionara cancelar para evitar el borrado del indicador económico.	
FA04		El sistema ignorara la eliminación del registro.

FLUJO ERROR

ESCENARIO: Datos incorrectos al agregar un nuevo indicador económico.

FE01	El usuario agregara los datos del nuevo indicador económico.	
FE02		El sistema valida las reglas de negocio en

		los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.
ESCENARIO: Datos incorrectos al editar los datos de algún indicador económico.		
FE01	El usuario realizara los cambios deseados en las columnas.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.

Tabla Descriptiva - Caso de Uso “Catalogo Indicadores Económicos”.

CASO DE USO		Catalogo Valles o Zona de Producción.	
CODIGO	UC18	Ref. Requerimiento	-
		Ref. Caso de Uso	UC05
Descripción general:			
La opción “Catalogo Valles o Zonas de Producción” mostrara una pantalla con un <i>grid</i> que permitirá gestionar la información sobre los valles o zonas de producción del producto en la base de datos.			
Actores: Usuario			
Pre-Condiciones:			
<ul style="list-style-type: none"> • Ingreso al sistema exitosamente. • Seleccionar el menú “Catálogos” 			
Pos-Condiciones:			
<ul style="list-style-type: none"> • valles o zona de producción de los productos . 			

FLUJO BASICO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo valle o zona de producción.		
FB01	El usuario presionara el botón agregar nuevo registro.	
FB02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas al valle o zona de producción.
FB03	El usuario agregara los datos del nuevo valle o zona de producción.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario presionara el botón de agregar.	
FB06		El sistema agregara el nuevo registro en la base de datos.
ESCENARIO: Actualizar los valle o zona de producción existentes.		
FB01	El usuario seleccionara el indicador económico que quiera editar.	
FB02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FB03	El usuario realizara los cambios deseados en las columnas.	
FB04		El sistema valida las reglas de negocio en los datos.
FB05	El usuario seleccionara el botón de guardar los cambios.	
FB06		El sistema realizara los cambios en la base de datos para el valle o zona de

		producción.
ESCENARIO: Borrar algún valle o zona de producción registrado.		
FB01	El usuario seleccionara el indicador económico que desea eliminar.	
FB02		El sistema mostrara un mensaje de confirmación
FB03	El usuario seleccionara aceptar para confirmar la eliminación del indicador económico.	
FB04		El sistema eliminara el registro en cuestión.
FLUJO ALTERNATIVO		
Código	Actor(es)	Sistema
ESCENARIO: Agregar nuevo valle o zona de producción.		
FA01	El usuario presionara el botón agregar nuevo registro.	
FA02		El sistema desplegara un espacio con las cajas de texto de las columnas relacionadas al valle o zona de producción.
FA03	El usuario agregara los datos del nuevo valle o zona de producción.	
FB04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario presionara el botón de cancelar.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón agregar nuevo

		registro.
ESCENARIO: Actualizar un valle o zona de producción existente.		
FA01	El usuario seleccionara el valle o zona de producción que quiera editar.	
FA02		El sistema volverá las columnas a modo editable y permitirá editar los datos.
FA03	El usuario realizara los cambios deseados en las columnas.	
FA04		El sistema valida las reglas de negocio en los datos.
FA05	El usuario seleccionara el botón de cancelar los cambios.	
FA06		El sistema regresa la pantalla al estado previo a presionar el botón editar registro.
ESCENARIO: Borrar algún valle o zona de producción seleccionado.		
FB01	El usuario seleccionara el valle o zona de producción que quiera borrar.	
FA02		El sistema mostrara un mensaje de confirmación.
FA03	El usuario seleccionara cancelar para evitar el borrado del valle o zona de producción.	
FA04		El sistema ignorara la eliminación del registro.
FLUJO ERROR		
ESCENARIO: Datos incorrectos al agregar un nuevo valle o zona de producción.		

FE01	El usuario agregara los datos del nuevo indicador económico.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.
ESCENARIO: Datos incorrectos al editar los datos de algún valle o zona de producción.		
FE01	El usuario realizara los cambios deseados en las columnas.	
FE02		El sistema valida las reglas de negocio en los datos.
FE03		El sistema muestra las columnas que tienen datos incorrectos.

Tabla Descriptiva - Caso de Uso "Catalogo Valle o Zona de Producción"

7.4 Anexo 4 – Script de base de datos SIMA

```
CREATE SCHEMA IF NOT EXISTS `simadb` DEFAULT CHARACTER SET latin1 ;
```

```
USE `simadb` ;
```

```
-----  
-- Table `simadb`.`atributos`  
-- Tabla catalogo de atributos del estado. Baja California.  
-----  
CREATE TABLE IF NOT EXISTS `simadb`.`atributos` (  
  `Id` INT(11) NOT NULL AUTO_INCREMENT ,  
  `Codigo` VARCHAR(45) NOT NULL ,  
  `Descripcion` VARCHAR(60) NULL DEFAULT NULL ,  
  PRIMARY KEY (`Id`))
```

```
-----  
-- Table `simadb`.`municipio`  
-- Tabla catalogo de municipios del estado. Baja California..  
-----  
CREATE TABLE IF NOT EXISTS `simadb`.`municipio` (  
  `Id` INT(11) NOT NULL AUTO_INCREMENT ,  
  `Nombre` VARCHAR(60) NOT NULL ,  
  `Estado` VARCHAR(60) NOT NULL ,  
  `Habitantes` VARCHAR(50) NULL DEFAULT NULL ,  
  PRIMARY KEY (`Id`))
```

```
-----  
-- Table `simadb`.`valle_zona_produccion`  
-- Tabla catalogo de valles del estado. Baja California.  
-----
```

```
CREATE TABLE IF NOT EXISTS `simadb`.`valle_zona_produccion` (  
  `Id` INT(11) NOT NULL AUTO_INCREMENT ,  
  `Nombre` VARCHAR(60) NOT NULL ,  
  `Descripcion` VARCHAR(100) NOT NULL ,  
  `Area_Geo` VARCHAR(40) NULL DEFAULT NULL ,  
  `Municipio_Id` INT(11) NOT NULL ,  
  PRIMARY KEY (`Id`),  
  INDEX `FK_Municipio` (`Municipio_Id` ASC),  
  CONSTRAINT `FK_Municipio`  
    FOREIGN KEY (`Municipio_Id` )  
    REFERENCES `simadb`.`municipio` (`Id` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```

```
-----  
-- Table `simadb`.`localidades`  
-- Tabla catalogo de localidades del estado. Baja California.  
-----
```

```
CREATE TABLE IF NOT EXISTS `simadb`.`localidades` (  
  `Id` INT(11) NOT NULL AUTO_INCREMENT ,  
  `Zona_Id` INT(11) NULL DEFAULT NULL ,  
  `Clima_Id` INT(11) NULL DEFAULT NULL ,  
  `Nombre` VARCHAR(70) NULL DEFAULT NULL ,  
  PRIMARY KEY (`Id`),  
  INDEX `FK_Zona` (`Zona_Id` ASC),  
  CONSTRAINT `FK_Zona`  
    FOREIGN KEY (`Zona_Id` )  
    REFERENCES `simadb`.`valle_zona_produccion` (`Id` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```

```

-----
-- Table `simadb`.`tipo_producto`
-- Tabla catalogo de tipos de producto dentro del SIMA.
-----
CREATE TABLE IF NOT EXISTS `simadb`.`tipo_producto` (
  `Id` INT(11) NOT NULL AUTO_INCREMENT ,
  `Codigo` VARCHAR(45) NOT NULL ,
  `Descripcion` VARCHAR(60) NULL DEFAULT NULL ,
  PRIMARY KEY (`Id`) )

```

```

-----
-- Table `simadb`.`productos`
-- Tabla entidad del producto dentro de SIMA.
-----
CREATE TABLE IF NOT EXISTS `simadb`.`productos` (
  `Id` INT(11) NOT NULL AUTO_INCREMENT ,
  `Producto` VARCHAR(60) NOT NULL ,
  `Nom_C` VARCHAR(100) NULL DEFAULT NULL ,
  `TProducto_Id` INT(11) NOT NULL ,
  `Descripcion` VARCHAR(300) NULL DEFAULT NULL ,
  `Localidad_Id` INT(11) NULL DEFAULT NULL ,
  PRIMARY KEY (`Id`) ,
  INDEX `FK_Localidades` (`Localidad_Id` ASC) ,
  INDEX `FK_TProducto` (`TProducto_Id` ASC) ,
  CONSTRAINT `FK_Localidades`
    FOREIGN KEY (`Localidad_Id` )
    REFERENCES `simadb`.`localidades` (`Id` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `FK_TProducto`
    FOREIGN KEY (`TProducto_Id` )
    REFERENCES `simadb`.`tipo_producto` (`Id` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```

```

-----
-- Table `simadb`.`atributos_producto`
-- Tabla entidad de los atributos asociados a ciertos productos en SIMA.
-----
CREATE TABLE IF NOT EXISTS `simadb`.`atributos_producto` (
  `Producto_Id` INT(11) NOT NULL ,
  `Atributo_Id` INT(11) NOT NULL ,
  `Valor` VARCHAR(300) NOT NULL ,
  PRIMARY KEY (`Producto_Id`, `Atributo_Id`),
  INDEX `FK_AP_Producto` (`Producto_Id` ASC),
  INDEX `FK_Atributos` (`Atributo_Id` ASC),
  CONSTRAINT `FK_AP_Producto`
    FOREIGN KEY (`Producto_Id`)
    REFERENCES `simadb`.`productos` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `FK_Atributos`
    FOREIGN KEY (`Atributo_Id`)
    REFERENCES `simadb`.`atributos` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```

```

-----
-- Table `simadb`.`clima`
-- Tabla catalogo de los climas asociados a las localidades en SIMA.
-----
CREATE TABLE IF NOT EXISTS `simadb`.`clima` (
  `Id` INT(11) NOT NULL AUTO_INCREMENT ,
  `Nombre` VARCHAR(45) NOT NULL ,
  `Descripcion` VARCHAR(100) NULL DEFAULT NULL ,
  PRIMARY KEY (`Id`))

```

```

-----
-- Table `simadb`.`indicador_economico`
-- Tabla catalogo de los indicadores economicos en SIMA.
-----
CREATE TABLE IF NOT EXISTS `simadb`.`indicador_economico` (
  `Id` INT(11) NOT NULL AUTO_INCREMENT ,
  `Nombre` VARCHAR(45) NOT NULL ,
  `Descripcion` VARCHAR(100) NULL DEFAULT NULL ,

```

```
`Unidades` VARCHAR(45) NOT NULL ,  
PRIMARY KEY (`Id`))
```

```
-----  
-- Table `simadb`.`localidades_producto`  
-- Tabla entidad de localidades asociados a los productos en SIMA.  
-----
```

```
CREATE TABLE IF NOT EXISTS `simadb`.`localidades_producto` (  
  `Producto_Id` INT(11) NOT NULL ,  
  `Localidad_Id` INT(11) NOT NULL ,  
  PRIMARY KEY (`Producto_Id`, `Localidad_Id`),  
  INDEX `FK_LPPProducto` (`Producto_Id` ASC),  
  INDEX `FK_LocalidadesLP` (`Localidad_Id` ASC))
```

```
-----  
-- Table `simadb`.`nutricion`  
-- Tabla catalogo de los indicadores de nutrición a los productos en SIMA.  
-----
```

```
CREATE TABLE IF NOT EXISTS `simadb`.`nutricion` (  
  `Id` INT(11) NOT NULL AUTO_INCREMENT ,  
  `Nombre` VARCHAR(60) NOT NULL ,  
  `Descripcion` VARCHAR(100) NULL DEFAULT NULL ,  
  `Unidades` VARCHAR(45) NOT NULL ,  
  PRIMARY KEY (`Id`))
```

```
-----  
-- Table `simadb`.`nutricion_producto`  
-- Tabla entidad de los indicadores de nutrición asociados a los productos en SIMA.  
-----
```

```
CREATE TABLE IF NOT EXISTS `simadb`.`nutricion_producto` (  
  `Nutricion_Id` INT(11) NOT NULL ,  
  `Producto_Id` INT(11) NOT NULL ,  
  `Valor` DOUBLE NOT NULL ,  
  PRIMARY KEY (`Nutricion_Id`, `Producto_Id`),  
  INDEX `FK_Nutricion` (`Nutricion_Id` ASC),  
  INDEX `FK_ProductoN` (`Producto_Id` ASC),  
  CONSTRAINT `FK_Nutricion`  
    FOREIGN KEY (`Nutricion_Id` )  
    REFERENCES `simadb`.`nutricion` (`Id` )
```

```

ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `FK_ProductoN`
FOREIGN KEY (`Producto_Id` )
REFERENCES `simadb`.`productos` (`Id` )
ON DELETE NO ACTION
ON UPDATE NO ACTION)

```

```

-----
-- Table `simadb`.`produccion_hdr`
-- Tabla entidad de los indicadores de producción anual en SIMA.
-----
CREATE TABLE IF NOT EXISTS `simadb`.`produccion_hdr` (
  `Id` INT(11) NOT NULL AUTO_INCREMENT ,
  `Producto_Id` INT(11) NOT NULL ,
  `Year` INT(11) NOT NULL ,
  PRIMARY KEY (`Id`),
  INDEX `FK_ProductoP` (`Producto_Id` ASC),
  CONSTRAINT `FK_ProductoP`
  FOREIGN KEY (`Producto_Id` )
  REFERENCES `simadb`.`productos` (`Id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)

```

```

-----
-- Table `simadb`.`produccion_dtl`
-- Tabla entidad de los indicadores de producción en SIMA.
-----
CREATE TABLE IF NOT EXISTS `simadb`.`produccion_dtl` (
  `id` INT(11) NOT NULL AUTO_INCREMENT ,
  `produccion_id` INT(11) NOT NULL ,
  `Valor` VARCHAR(30) NOT NULL ,
  `ieconomico_id` INT(11) NOT NULL ,
  PRIMARY KEY (`id`),
  INDEX `FK_ProduccionHdr` (`produccion_id` ASC),
  INDEX `FK_iEconomico` (`ieconomico_id` ASC),
  CONSTRAINT `FK_iEconomico`
  FOREIGN KEY (`ieconomico_id` )
  REFERENCES `simadb`.`indicador_economico` (`Id` )

```

```

ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `FK_ProduccionHdr`
    FOREIGN KEY (`produccion_id` )
  REFERENCES `simadb`.`produccion_hdr` (`Id` )
ON DELETE NO ACTION
  ON UPDATE NO ACTION)

```

```

-----
-- Table `simadb`.`usuarios`
-- Tabla entidad de los usuarios en SIMA.
-----
CREATE TABLE IF NOT EXISTS `simadb`.`usuarios` (
  `ID` INT(11) NOT NULL AUTO_INCREMENT ,
  `User` VARCHAR(60) NOT NULL ,
  `Password` VARCHAR(60) NOT NULL ,
  PRIMARY KEY (`ID`))

```

```

-----
-- Placeholder table for view `simadb`.`vw_producciondtl`
-- Prodedimiento que extrae los indicadores de producción por producto en SIMA.
-----
CREATE TABLE IF NOT EXISTS `simadb`.`vw_producciondtl` (`Producto_Id` INT,
`Producto` INT, `year` INT, `Nombre` INT, `Descripcion` INT, `Valor` INT, `Unidades`
INT);

```

```

-----
-- procedure getCombo_Atributos
-- Prodedimiento que extrae el catalogo de los atributos en SIMA.
-----
DELIMITER $$
USE `simadb` $$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `getCombo_Atributos`()
BEGIN
SELECT *
  FROM atributos;
END$$

```

```

-----
-- procedure getCombo_Clima
-- Procedimiento que extrae el catalogo de los climas en SIMA.
-----
DELIMITER $$
USE `simadb`$$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `getCombo_Clima`()
BEGIN
SELECT *
    FROM clima;
END$$

```

```

-----
-- procedure getCombo_Localidad
-- Procedimiento que extrae el catalogo de las localidades en SIMA.
-----
DELIMITER $$
USE `simadb`$$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `getCombo_Localidad`()
BEGIN
SELECT l.Id,v.Nombre
    FROM localidades l, valle_zona_produccion v
    WHERE l.Zona_Id = v.Id;

END$$

```

```

-----
-- procedure getCombo_Municipios
-- Procedimiento que extrae el catalogo de las municipios en SIMA.
-----
DELIMITER $$
USE `simadb`$$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `getCombo_Municipios`()
BEGIN
SELECT *
    FROM municipio;
END$$

```

```

-----
-- procedure getCombo_Nutrientes
-- Procedimiento que extrae el catalogo de las nutrientes en SIMA.
-----
DELIMITER $$
USE `simadb` $$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `getCombo_Nutrientes`()
BEGIN
    SELECT *
        FROM nutricion;
END$$

```

```

-----
-- procedure getCombo_ProdType
-- Procedimiento que extrae el catalogo de tipo de producto en SIMA.
-----
DELIMITER $$
USE `simadb` $$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `getCombo_ProdType`()
BEGIN
    SELECT *
        FROM tipo_producto;
END$$

```

```

-----
-- procedure getCombo_Productos
-- Procedimiento que extrae el catalogo de los productos existentes en SIMA.
-----
DELIMITER $$
USE `simadb` $$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `getCombo_Productos`()
BEGIN

    SELECT * FROM productos;

END$$

```

```

-----
-- procedure getCombo_Zonas
-- Procedimiento que extrae el catalogo de los productos existentes en SIMA.
-----
DELIMITER $$
USE `simadb` $$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `getCombo_Zonas`()
BEGIN
SELECT *
    FROM valle_zona_produccion;
END$$

```

```

-----
-- procedure getCombo_iEconomicos
-- Procedimiento que extrae el catalogo de los productos existentes en SIMA.
-----
DELIMITER $$
USE `simadb` $$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `getCombo_iEconomicos`()
BEGIN
SELECT *
    FROM indicador_economico;
END$$

```

```

-----
-- procedure getPTablaNutrientes
-- Procedimiento que extrae los nutrientes de cierto producto existentes en SIMA.
-----
DELIMITER $$
USE `simadb` $$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `getPTablaNutrientes`(
    IN in_productId INT
)
BEGIN
SELECT n.Id,n.Nombre,np.Valor,n.Unidades
    FROM simadb.nutricion_producto np,simadb.nutricion n
WHERE Producto_Id = in_productId

```

```

        AND np.Nutricion_Id = n.Id
    ORDER BY Nombre;
END$$

```

```

-----
-- procedure getProduccionColumns
-- Prodedimiento que extrae los los indicadores para cierto producto existentes en SIMA.
-----
DELIMITER $$
USE `simadb` $$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `getProduccionColumns`(
    IN in_productId INT
)
BEGIN
    SELECT DISTINCT Nombre,Unidades FROM simadb.vw_producciondtl
        WHERE Producto_id = in_productId
    ORDER BY year,Nombre;
END$$

```

```

-----
-- procedure getProductsAtributs
-- Prodedimiento que extrae los los atributos para cierto producto existentes en SIMA.
-----
DELIMITER $$
USE `simadb` $$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `getProductsAtributs`(
    IN in_productId INT
)
BEGIN

    SELECT a.codigo,ap.Valor
        FROM simadb.atributos_producto ap, simadb.atributos a
    WHERE ap.Atributo_Id = a.Id
        AND ap.Producto_Id = in_productId;
END$$

```

```

-----
-- procedure getProducts_byId
-- Prodedimiento que extrae cierto producto existentes por id en SIMA.
-----
DELIMITER $$
USE `simadb`$$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `getProducts_byId`(
    IN in_productId INT
)
BEGIN
    SELECT p.Id,p.Producto, p.Nom_C, p.Descripcion
    FROM simadb.productos p
    WHERE Id = in_productId;
END$$

```

```

-----
-- procedure getValidUser_login
-- Prodedimiento que valida las credenciales de los usuarios SIMA.
-----
DELIMITER $$
USE `simadb`$$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `getValidUser_login`(
    IN in_userName VARCHAR(60),
    IN in_password VARCHAR(60),
    OUT out_userId INT
)
BEGIN
    SELECT ID into out_userId
    FROM usuarios
    WHERE User = in_userName
    AND Password = in_password;
END$$

```

```

-----
-- procedure get_H_ProduccionDI
-- Procedimiento que extrae el historial de indicadores de los productos de SIMA.
-----
DELIMITER $$
USE `simadb`$$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `get_H_ProduccionDI`(
    IN in_productId INT
)
BEGIN
    SELECT * FROM simadb.vw_producciondtl
        WHERE Producto_id = in_productId
    ORDER BY year,Nombre;
END$$

```

```

-----
-- procedure setProduccionDtl
-- Procedimiento que extrae el historial de producción de los productos de SIMA.
-----
DELIMITER $$
USE `simadb`$$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `setProduccionDtl`(
    IN in_produccionId INT,
    IN in_valor VARCHAR(30),
    IN in_iEconomico INT,
    OUT out_ProduccionDtlId INT
)
BEGIN
    INSERT INTO simadb.produccion_dtl(Id,produccion_id,Valor,ieconomico_id)
        VALUES(NULL,in_produccionId,in_valor,in_iEconomico);

    SELECT LAST_INSERT_ID() INTO out_ProduccionDtlId FROM DUAL;
END$$

```

```

-----
-- procedure setProduccionHdr
-- Procedimiento que extrae el historial de producción anual de los productos de SIMA.
-----
DELIMITER $$
USE `simadb` $$
CREATE DEFINER=`MTIC`@`%` PROCEDURE `setProduccionHdr`(
    IN in_productId INT,
    IN in_year VARCHAR(11),
    OUT out_ProduccionId INT
)
BEGIN
    INSERT INTO produccion_hdr(Id,Producto_Id,Year)
        VALUES(out_ProduccionId,in_productId,in_year);

    SELECT LAST_INSERT_ID() INTO out_ProduccionId FROM DUAL;
END$$

```

```

-----
-- View `simadb`.`vw_producciondtl`
-- Vista que reúne la inf. de producción e indicadore econ. de los productos de SIMA.
-----
DROP TABLE IF EXISTS `simadb`.`vw_producciondtl`;
USE `simadb`;
CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=`MTIC`@`%`
SQL SECURITY DEFINER VIEW `simadb`.`vw_producciondtl` AS

select `p`.`Id` AS `Producto_Id`,`p`.`Producto` AS `Producto`,`h`.`Year` AS
`year`,`i`.`Nombre` AS `Nombre`,`i`.`Descripcion` AS `Descripcion`,`d`.`Valor` AS
`Valor`,`i`.`Unidades` AS `Unidades` from (((`simadb`.`produccion_hdr` `h` join
`simadb`.`produccion_dtl` `d`) join `simadb`.`productos` `p`) join
`simadb`.`indicador_economico` `i`) where ((`d`.`produccion_id` = `h`.`Id`) and
(`h`.`Producto_Id` = `p`.`Id`) and (`d`.`ieconomico_id` = `i`.`Id`));

```

7.5 Anexo 5 Código fuente sistema de captura

VIEW-LAYER

<Login.html>

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <title>SIMA</title>
    <link rel="stylesheet" type="text/css" href="web/css/login.css" />
    <script src="web/js/login.js" type="text/javascript"></script>
</head>

<body onload="onloadLogin();">
    <header>
        <h1> SIMA Project</h1>
    </header>
    <aside>
        <div>
            
        </div>
    </aside>
    <section>
    <article>
        <header>
            <h2>LOGIN</h2>
        </header>
        <div class="loginBox">
            <table width="100%">
                <tr>
                    <td colspan="2" align="center">BETA version</td>
                </tr>
                <tr>
                    <td align="right">Usuario:</td>
                    <td><input type="text" name="user_name" id="txt_user"/></td>
                </tr>
                <tr>
                    <td align="right">Contraseña:</td>
                    <td ><input type="password" name="user_pswd" id="txt_pswd"
                        onkeypress="onEnter();" /></td>
                </tr>
            </table>
        </div>
    </article>
</section>
</div>
```

```

        <tr>
            <td colspan="2" align="center"><input type="button"
                value="Acceder" onclick="login();"
/></td>
        </tr>
        <tr>
            <td colspan="2" align="center">
                <div id="response_log" class="errorMsg"></div>
            </td>
        </tr>
    </table>
</div>
</article>
</section>

<script type="text/javascript">
    $(document).ready(function() {
        onloadLogin();
    });
</script>
</body>
</html>

```

<main.php>

```

<?php
include('../classes/ctrl/lock.php');
?>
<html>
<head>
    <title>SIMA</title>
    <link rel="stylesheet" type="text/css" href="css/main.css" />
    <script src="js/main.js" type="text/javascript"></script>
</head>
<body onload="setHiddenLoading();
selectTab(this,'../classes/ctrl/MenuVCtrl.php?CmdCtrl=products');">
    <header>
        <hgroup>
            <h1> SIMA Project</h1>
        </hgroup>
        <nav id="main_menu">

```

```

        <ul>
            <li class="selected"
onclick="selectTab(this,'../clases/ctrl/MenuVCtrl.php?CmdCtrl=products');"><a>Products</a></li>
            <li
onclick="selectTab(this,'../clases/ctrl/MenuVCtrl.php?CmdCtrl=production');"><a>Produccion</a>
</li>
            <li
onclick="selectTab(this,'../clases/ctrl/MenuVCtrl.php?CmdCtrl=catalogs');"><a>Catalogs</a></li>
            <li
onclick="document.location.href='../clases/ctrl/logout.php'"><a>Logout</a></li>
        </ul>
    </nav>
</header>

<aside id="aside">
</aside>

<section>
    <article>
        <header>
            <div id="Content_title" class="titleBox"></div>
        </header>
        <iframe name="iFramecontent" id="iFramecontent"

                width="100%"

                frameborder="0"
                marginwidth="0"
                height="90%"
                scrolling="auto"></ifram>
        </article>
    </section>

<footer>
    MTIC 2011
</footer>
</body>
</html>

Scripts
<login.js>

function onloadLogin(){

```

```

        document.getElementById("txt_user").focus();
    }
    function onEnter(){
        if(event.keyCode=='13'){
            login();
        }
    }
}

function login(){
    var user = document.getElementById('txt_user').value;
    var pswd = document.getElementById('txt_pswd').value;
    var req = initRequest();
    req.onreadystatechange = function() {
        if (req.readyState == 4) {
            if (req.status == 200) {
                if(req.responseText.length>5)
                    document.getElementById('response_log').innerHTML = req.responseText;
            }
            else
                document.location = 'web/main.php';
        }
    }
};

    req.open("GET", 'classes/ctrl/CommonCtrl.php?time='+new
Date().getTime()+ '&CmdCtrl=logIn&user_name='+user+'&user_pswd='+pswd, true);
    req.send(null);
}

function initRequest() {
    if (window.XMLHttpRequest) {
        return new XMLHttpRequest();
    } else if (window.ActiveXObject) {
isIE = true;
        return new ActiveXObject("Microsoft.XMLHTTP");
    }
}
}

```

<main.js>

```
document.writeln("<div id=\"barra\" style=\"cursor:wait; height: 100%; width:100%; position: absolute; left: 0px; top: 0px; visibility: visible; filter: alpha(opacity=55); -moz-opacity: .55; \">");  
document.writeln("<center>");
```

```
for(c=0;c<5;c++)  
    document.writeln("<BR>");  
document.writeln("<IMG src=\"../classes/KoolControls/KoolAjax/loading/2.gif\" name=\"iLoading\" height=\"120\" width=\"120\">");  
document.writeln("</center></div>");
```

```
function setVisibleLoading(){  
    document.getElementById('barra').style.visibility='visible';  
    document.getElementById('barra').style.display = 'block';  
}
```

```
function setHiddenLoading(){  
    document.getElementById('barra').style.visibility='hidden';  
}
```

```
function selectTab(tab,url){  
    var liList = document.getElementById("main_menu").getElementsByTagName("li");  
    for(var i=0; i<liList.length; i++){  
        liList[i].className="";  
    }  
    tab.className='selected';  
    document.getElementById('Content_title').innerHTML = "  
    document.getElementById("iFramecontent").src="";  
    makeMenu(url);  
}
```

```
function makeMenu(url){  
    var req = initRequest();  
    if(url.length>1){  
        req.onreadystatechange = function() {  
            if (req.readyState == 4) {  
                if (req.status == 200) {  
                    if(req.responseText.length>1)
```

```

        document.getElementById('aside').innerHTML = req.responseText;
            }
        }
    };
    req.open("GET", url, true);
    req.send(null);
}
}

function loadTitle(varTitle,url){
    document.getElementById("iFramecontent").src="";
    setVisibleLoading();
    document.getElementById('Content_title').innerHTML = varTitle;
    document.getElementById("iFramecontent").src=url;
}

function initRequest() {
    if (window.XMLHttpRequest) {
        return new XMLHttpRequest();
    } else if (window.ActiveXObject) {
        isIE = true;
        return new ActiveXObject("Microsoft.XMLHTTP");
    }
}
}

```

CSS

```

html {
    margin: 0;
    padding: 0;
    /*background: url(../img/grid.jpg) center top repeat;*/
}
body {
    width: 940px;
    margin: 0 auto;
    font: 12pt/1.5em Helvetica,"Helvetica neue", Arial, sans-serif;
}
header,section,article,aside,footer{

```

<login.css>

```

display: block;
}
h1 {
font: 32pt Helvetica,"Helvetica neue", Arial, sans-serif;
/*background: url(..img/grid.jpg) center top repeat;*/
/*background-color: #72B8EB;*/
-webkit-border-radius: 10px;
text-align: left;
text-transform: uppercase;
text-shadow: 1px 1px 2px #fff;
}
aside{
float: left;
width: 611px;
height: 550px;
border-top: 2px solid #3898E2;
border-right: 2px solid #3898E2;
border-bottom: 2px solid #3898E2;
border-left: 2px solid #3898E2;
background: url(..img/grid.jpg) center top repeat;
-webkit-border-radius: 10px;
}
aside div{
padding-top:10px;
text-align:left;
}
section{
float: right;
width: 300px;
height: 250px;
padding-left: 20px;
/*padding-top: 10px;*/
background-color: #FFFFFF;
/*background: url(..img/grid.jpg) center top repeat;*/

border-top: 2px solid #3898E2;
border-right: 2px solid #3898E2;
border-bottom: 2px solid #3898E2;
border-left: 2px solid #3898E2;

-webkit-border-radius: 10px;
}

```

```

loginBox{
    background: #FFFFFF;
    -webkit-border-top-right-radius: 5px;
    -webkit-border-top-left-radius: 5px;
}

<main.css>
html {
    margin: 0;
    padding: 0;
}
body {
    width: 940px;
    margin: 0 auto;
    font: 12pt/1.5em Helvetica,"Helvetica neue", Arial, sans-serif;
}
header,section,article,aside,footer{
    display: block;
}
header nav ul {
    list-style: none;
    text-align:left;
    padding-left: 270px;
    margin:0;
}
header nav ul li {
    display: inline;
    border: solid;
    position: relative;
    top: 3px;
    border: 1px solid #336699;
    background: #7DC6F4;
    -webkit-border-top-right-radius: 5px;
    -webkit-border-top-left-radius: 5px;
}
header nav li a {
    padding: 0 1em;
}
header nav li:hover {
    position: relative;
    top: 3px;
    background-color:white;
}

```

```

}
header nav ul li.selected a{
    position: relative;
    top: 3px;
    background-color:white;
}
header nav ul li.selected {
    background-color:white;
}
aside{
    float: left;
    width: 200px;
    min-height: 550px;
    border: 1px solid #336699;
    background-color: #7DC6F4;
    background-image: url(img/blueBack.png) center top repeat;
}
#aside ul {
    list-style: none;
    text-align:center;
    margin:0;
    padding-left: 0px;
    padding-top: 10px;
}
#aside ul li {
    padding-top: 10px;
    position: relative;
    margin-top:10;
    background: #FFFFFF;
    height: 2em;
    -webkit-border-top-right-radius: 10px;
    -webkit-border-top-left-radius: 10px;
    -webkit-border-bottom-right-radius: 10px;
    -webkit-border-bottom-left-radius: 10px;
}
#aside ul li:hover {
    position: relative;
    left: 8px;
    background-color:white;
    border: 0px solid #336699;
}
#aside nav ul li.selected {

```

```

        background-color:#336699;
    }
section{
    float: right;
    width: 718px;
    min-height: 550px;
    padding-left: 20px;
    padding-top: 10px;
    margin: 0 0 20px 0px;
    border-top: 1px solid #336699;
}
titleBox{
    font: 14pt Helvetica,"Helvetica neue", Arial, sans-serif;
    font-weight: lighter;
    text-transform: uppercase;
    color: #493831;
    padding-bottom: 10px;
    margin: 0;
}
footer{
    clear:both;
    width:930px;
    height: 50px;
    color:#493831 ;
    background:#99CCCC;
}
/*Headings*/
h1 {
    text-align: left;
}
a:link, a:visited {
    font:10pt normal Helvetica,"Helvetica neue", Arial, sans-serif;
    color:#d2543c;
    text-decoration:none;
}

```

CONTROLER-LAYER

<CommonCtrl.php>

```
<?php
include("../beans/entity/UserEntity.php");
$cmdCtrl = $_GET["CmdCtrl"];
$Rresult = null;

switch ($cmdCtrl) {
    case "login":
        $Rresult = login();
        break;
    default:
        $Rresult = "Evento no Encontrado";
}
echo $Rresult;
function login(){
    $nombre = $_GET["user_name"];
    $pswd = $_GET["user_pswd"];

    $usuario = new UserEntity($nombre,$pswd);
    $user_Id = $usuario->getValidaUser();
    if ($user_Id!=0){
        $usuario->setUserId($user_Id);
        session_start();
        $_SESSION['login_user']=$usuario;
        return $user_Id;
    }else {
        return "Usuario o Password no validos";
    }
}
?>
```

<GridCtrl.php>

```
<?php
include("../beans/biz/GridBiz.php");
include("../beans/entity/ComboEntity.php");

ini_set('display_errors', '0');
```

```

error_reporting(E_ALL | E_STRICT);

$KoolControlsFolder = "../KoolControls";
$cmdCtrl = $_GET["CmdCtrl"];
$cmdCmb = $_GET["CmdCmb"];

$gridBiz = new GridBiz();

require $KoolControlsFolder."/KoolAjax/koolajax.php";
require $KoolControlsFolder."/KoolGrid/koolgrid.php";
$koolajax->scriptFolder = $KoolControlsFolder."/KoolAjax";
$grid->scriptFolder = $KoolControlsFolder."/KoolGrid";

$grid = new KoolGrid("grid");
$grid->AjaxLoadingImage = $KoolControlsFolder."/KoolAjax/loading/15.gif";

include("../beans/biz/eventsGrid.php");

$grid->EventHandler = new MyGridEventHandler();

switch ($cmdCtrl) {
    case "products":
        $gridBiz->getGridProducts($grid);
        break;
    case "municipios":
        $gridBiz->getGridMunicipio($grid);
        break;
    case "localidades":
        $gridBiz->getGridLocalidades($grid);
        break;
    case "p_type":
        $gridBiz->getGridProductType($grid);
        break;
    case "clima":
        $gridBiz->getGridClima($grid);
        break;
    case "i_econ":
        $gridBiz->getGridI_Economicos($grid);
        break;
    case "nutricion":
        $gridBiz->getGridNutricion($grid);
        break;
    case "v_zona":

```

```

        $gridBiz->getGridV_Zona($grid);
        break;
    case "atributos":
        $gridBiz->getGridAtributos($grid);
        break;
    case "prod_nutrientes":
        $gridBiz->getGridProdNutrientes($grid);
        break;
    case "prod_atributos":
        $gridBiz->getGridProdAtributos($grid);
        break;
    case "prod_localidades":
        $gridBiz->getGridProdLocalidades($grid);
        break;
    case "produccion":
        $gridBiz->getGridProduccion($grid);
        break;
    case "h_produccion":
        $gridBiz->getGridH_Produccion($grid);
        break;
    default:
        echo "Evento no Encontrado";
}

if($cmdCmb=='single')
    $arlCombo = $gridBiz->getCombo_Products();

if(isset($_POST["ExportToExcel"])){
    //$grid->ExportSettings->IgnorePaging = true;
    $grid->ExportSettings->FileName = $cmdCtrl;
    $grid->GetInstanceMasterTable()->ExportToExcel();
}

if(isset($_POST["ExportToPDF"])){
    //$grid->ExportSettings->IgnorePaging = true;
    $grid->ExportSettings->FileName = $cmdCtrl;
    $grid->GetInstanceMasterTable()->ExportToPDF();
}

unset($gridBiz);
echo $koolajax->Render();
?>
<html>
    <head>
        <script type="text/javascript">
            parent.setHiddenLoading();

```

```

        function filterProductsSingle(value) {
            var _mastertable = grid.getMasterTable();
            var _columns = _mastertable.getColumns();
            var _columnId = _columns[1];
            if(value!='0'){

                _columnId.filter("Equal",value);
            }else{
                _columnId.filter("No_Filter",value);
            }
            grid.commit();
        }

    </script>
</head>

<body>
    <form method="post">
        <input type="submit" name="ExportToPDF" value = "Export to PDF" />
        <input type="submit" name="ExportToExcel" value = "Export to Excel" />
        <br/>
        <?php
            if($cmdCmb=='single'){?>
                Productos:<select id="cbbCustomer"
onchange="filterProductsSingle(this.value);">
                    <option value="0">All</option>
                    <?php
                        foreach ($arlCombo as $combo) {

                            echo "<option value=" . $combo-
>getId()." ">". $combo->getCode() ."</option>";
                        }
                    ?>
                </select>
                <?php } echo $grid->Render(); ?>
            </form>
        </body>
    </html>

    <lock.php>

    <?php

```

```

session_start();
$login_session=$_SESSION['login_user'];

if(!isset($login_session)){
    header("Location: ../index.html");
}
?>

```

<logout.php>

```

<?php
session_start();
if(session_destroy()){
    header("Location: ../index.html");
}
?>

```

<MenuVCtrl.php>

```

<?php
ini_set('display_errors', '0');
error_reporting(E_ALL | E_STRICT);
$cmdCtrl = $_GET["CmdCtrl"];
switch ($cmdCtrl) {
    case "catalogs":
        menuCatalogs($KoolControlsFolder,$gridBiz);
        break;
    case "products":
        menuProducts($KoolControlsFolder,$gridBiz);
        break;
    case "production":
        menuProduction($KoolControlsFolder,$gridBiz);
        break;
    default:
}

function menuCatalogs(){
    $html = '<ul>
        <li onclick="parent.loadTitle('."Tipo de
Producto"',.'../clases/ctrl/GridCtrl.php?CmdCtrl=p_type".')">Tipo-Producto</li>

```

```

        <li
onclick="parent.loadTitle('."Municipios"','!'"../clases/ctrl/GridCtrl.php?CmdCtrl=municipios"'.');"
>Municipios</li>
        <li
onclick="parent.loadTitle('."Clima"','!'"../clases/ctrl/GridCtrl.php?CmdCtrl=clima"'.');">Clima</li
>
        <li onclick="parent.loadTitle('."Indicadores
Economicos"','!'"../clases/ctrl/GridCtrl.php?CmdCtrl=i_econ"'.');">Indicadores Economicos</li>
        <li
onclick="parent.loadTitle('."Nutrientes"','!'"../clases/ctrl/GridCtrl.php?CmdCtrl=nutricion"'.');">N
utricion</li>
<li onclick="parent.loadTitle('."Valle o Zona de
Produccion"','!'"../clases/ctrl/GridCtrl.php?CmdCtrl=v_zona"'.');">Valle-Zona-Produccion</li>
        <li
onclick="parent.loadTitle('."Localidades"','!'"../clases/ctrl/GridCtrl.php?CmdCtrl=localidades"'.');"
>Localidades</li>
        <li
onclick="parent.loadTitle('."Atributos"','!'"../clases/ctrl/GridCtrl.php?CmdCtrl=atributos"'.');">Atri
butos</li>
    </ul>;
    echo $html;
}
function menuProduction(){
    $html = '<ul>
        <li
onclick="parent.loadTitle('."Produccion"','!'"../clases/ctrl/GridCtrl.php?CmdCtrl=produccion"'.');"
>Produccion</li>
        <li onclick="parent.loadTitle('."Historial
Produccion"','!'"../clases/ctrl/ViewResumenCtrl.php"'.');">Ficha Producto</li>
    </ul>;
    echo $html;
}
function menuProducts(){
    $html = '<ul>
        <li
onclick="parent.loadTitle('."Productos"','!'"../clases/ctrl/GridCtrl.php?CmdCtrl=products"'.');">Pro
ductos</li>
        <li onclick="parent.loadTitle('."Localidades del
Producto"','!'"../clases/ctrl/GridCtrl.php?CmdCtrl=prod_localidades&CmdCmb=single"'.');">Local
idades</li>
        <li onclick="parent.loadTitle('."Atributos del
Producto"','!'"../clases/ctrl/GridCtrl.php?CmdCtrl=prod_atributos&CmdCmb=single"'.');">Atributo
s</li>

```

```

        <li onclick="parent.loadTitle('Nutrientes del
Producto"',.'../clases/ctrl/GridCtrl.php?CmdCtrl=prod_nutrientes&CmdCmb=single"',.)">Nutrien
tes</li>
    </ul>;
    echo $html;
}
?>

```

[<viewDetalleFicha.php>](#)

```

<?php
include("../beans/biz/GridBiz.php");
include("../beans/entity/ComboEntity.php");

$productId=$_GET["productId"];
$gridBiz = new GridBiz();
$arCombo = $gridBiz->getCombo_Products();
$arColumns = $gridBiz->getProduccionColumns($productId);
$arTable = $gridBiz->getGridH_Produccion($productId);
$arNutrientes = $gridBiz->getProductNutrientesTable($productId);
$arProductsData = $gridBiz->getProductsData($productId);
$arProductsAtributos = $gridBiz->getProductsAtributos($productId);
$yearCol = 0;
?>
<html>
    <head>
    </head>
<body>
    <br/>
    <table>
        <tr>
            <td>Producto:</td>
            <td><?php echo $arProductsData[0];?></td>
        </tr>
        <tr>
            <td>Nombre Cientifico:</td>
            <td><?php echo $arProductsData[1];?></td>
        </tr>
        <tr>
            <td>Descripci—n:</td>
            <td><?php echo $arProductsData[2];?></td>

```

```

        </tr>

        <?php
            foreach ($arlProductsAtributos as $combo) {

                echo "<tr><td>".$combo->getId()."</td><td>".$combo->getCode()
."</td></tr>";

            }
        ?>
    </table>
<br/>
<br/>
    <table>
        <tr>
            <td colspan="3">Tabla Nutritional</td>
        </tr>
        <?php
            foreach ($arlNutrientes as $arlRowNut) {
                echo "<tr>";
                echo "<td>".$arlRowNut["Nutriente"]."</td>";
                echo "<td>".$arlRowNut["Valor"]."</td>";
                echo "<td>".$arlRowNut["Unidades"]."</td>";
                echo "</tr>";
            }
        ?>
    </table>
<br/><br/>
    <table width="100%">
        <?php
            if(sizeof($arlColumns)>0){?>
                <tr>
                    <td colspan="<?php echo sizeof($arlColumns); ?>">
                        Serie Historica
                    </td>
                </tr>
            <?php }?>
        <tr>
            <?php
                if(sizeof($arlColumns)>0)
                    echo "<td>A-o</td>";

                foreach ($arlColumns as $column) {

```

```

        echo "<td>".$column."</td>";
    }
    ?>
</tr>
<?php
    foreach ($arlTable as $arlRow) {
        echo "<tr>";
        foreach ($arlRow as $valor) {
            echo "<td>".$valor."</td>";
        }
        echo "</tr>";
    }
    ?>
</table>
</body>
</html>

```

<viewResumenCtrl.php>

```

<?php
//include("../beans/populate/CommonPopulate.php");
include("../beans/biz/GridBiz.php");
include("../beans/entity/ComboEntity.php");
$gridBiz = new GridBiz();
$arlCombo = $gridBiz->getCombo_Products();
$arlColumns = $gridBiz->getProduccionColumns(4);
$arlTable = $gridBiz->getGridH_Produccion(4);
$yearCol = 0;
?>
<html>
    <head>
        <script type="text/javascript">
            parent.setHiddenLoading();

            function filterViewProduct(ProductId){
                if (ProductId==""){
                    document.getElementById("txtHint").innerHTML="";
                    return;
                }
                if (window.XMLHttpRequest){// code for IE7+, Firefox, Chrome, Opera,
Safari
                    xmlhttp=new XMLHttpRequest();

```

```

        }else{// code for IE6, IE5
            xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
        }
        xmlhttp.onreadystatechange=function(){
            if (xmlhttp.readyState==4 && xmlhttp.status==200){

document.getElementById("FichaDtl").innerHTML=xmlhttp.responseText;
            }
            }
        xmlhttp.open("GET","viewDetalleFicha.php?productId="+ProductId,true);
        xmlhttp.send();
    }    </script>
</head>

<body>

Selecciona Productos:
    <select id="cbbCustomer" onchange="filterViewProduct(this.value);">
        <option value="0">All</option>
        <?php
            foreach ($arlCombo as $combo) {

                echo "<option value='".$combo->getId()."'>".$combo->getCode()
."</option>";
            }
        ?>
    </select>
    <div id="FichaDtl"><b>Cargando Informaci—n Ficha Tecnica...</b></div>
</body>
</html>

```

CLASS-LAYER

<CommonBiz.php>

```
<?php
include("../beans/db/Connection.php");
include("../beans/populate/CommonPopulate.php");

class CommonBiz{
    private $commonPopulate;
    function CommonBiz(){
        $this->commonPopulate = new CommonPopulate();
    }

    public function getValidaUser($user_name,$user_pswd){
        return $this->commonPopulate->getUserValidate($user_name, $user_pswd);
    }
}
?>
```

<GridBiz.php>

```
<?php
include("../beans/db/Connection.php");
include("../beans/populate/GridPopulate.php");

class GridBiz{

    private $gridPopulate;

    function GridBiz(){
        $this->gridPopulate = new GridPopulate();
    }

    function __destruct() {
        unset($this->gridPopulate);
    }

    private function initGrid($grid){
        $grid->AjaxEnabled = true;
        $grid->AllowEditing = true;
        $grid->AllowDeleting = true;
    }
}
```

```

        $grid->RowAlternative = true;
        $grid->AllowSorting = true;
        $grid->MasterTable->Pager = new GridPrevNextAndNumericPager();
        $grid->Width = "99%";
        $grid->ColumnWrap = true;
        $grid->styleFolder = "default";
        $grid->MasterTable->ShowFunctionPanel = true;
        $grid->MasterTable->InsertSettings->Mode = "Form";
        $grid->MasterTable->InsertSettings->ColumnNumber = 2;
        return $grid;
    }

    private function initGridParam($grid,$editing,$deleting,$panel){
        $grid->AjaxEnabled = true;
        $grid->AllowEditing = $editing;
        $grid->AllowDeleting = $deleting;
        $grid->RowAlternative = true;
        $grid->AllowSorting = true;
        $grid->MasterTable->Pager = new GridPrevNextAndNumericPager();
        $grid->Width = "99%";
        $grid->ColumnWrap = true;
        $grid->styleFolder = "default";
        $grid->MasterTable->ShowFunctionPanel = $panel;
        $grid->MasterTable->InsertSettings->Mode = "Form";
        $grid->MasterTable->InsertSettings->ColumnNumber = 2;
        return $grid;
    }

    public function getGridProducts($grid){
        $this->initGrid($grid);
        $this->gridPopulate->getGridProducts($grid);
        $grid->Process();
        $this->gridPopulate->closeConnection();
    }

    public function getGridClima($grid){
        $this->initGrid($grid);
        $this->gridPopulate->getGridClima($grid);
        $grid->Process();
        $this->gridPopulate->closeConnection();
    }

    public function getGridProductType($grid){

```

```

        $this->initGrid($grid);
        $this->gridPopulate->getGridProductType($grid);
        $grid->Process();
        $this->gridPopulate->closeConnection();
    }

    public function getGridMunicipio($grid){
        $this->initGrid($grid);
        $this->gridPopulate->getGridMunicipio($grid);
        $grid->Process();
        $this->gridPopulate->closeConnection();
    }

    public function getGridLocalidades($grid){
        $this->initGrid($grid);
        $this->gridPopulate->getGridLocalidades($grid);
        $grid->Process();
        $this->gridPopulate->closeConnection();
    }

    public function getGridI_Economicos($grid){
        $this->initGrid($grid);
        $this->gridPopulate->getGridI_Economicos($grid);
        $grid->Process();
        $this->gridPopulate->closeConnection();
    }

    public function getGridNutricion($grid){
        $this->initGrid($grid);
        $this->gridPopulate->getGridNutricion($grid);
        $grid->Process();
        $this->gridPopulate->closeConnection();
    }

    public function getGridV_Zona($grid){
        $this->initGrid($grid);
        $this->gridPopulate->getGridV_Zona($grid);
        $grid->Process();
        $this->gridPopulate->closeConnection();
    }

    public function getGridAtributos($grid){
        $this->initGrid($grid);

```

```

        $this->gridPopulate->getGridAtributos($grid);
        $grid->Process();
        $this->gridPopulate->closeConnection();
    }

    public function getGridProdNutrientes($grid){
        $this->initGridParam($grid,false,false,false);
        $this->gridPopulate->getGridProdNutrientes($grid);
        $grid->Process();
        $this->gridPopulate->closeConnection();
    }

    public function getGridProdAtributos($grid){
        $this->initGridParam($grid,false,false,false);
        $this->gridPopulate->getGridProdAtributos($grid);
        $grid->Process();
        $this->gridPopulate->closeConnection();
    }

    public function getGridProdLocalidades($grid){
        $this->initGridParam($grid,false,false,false);
        $this->gridPopulate->getGridProdLocalidades($grid);
        $grid->Process();
        $this->gridPopulate->closeConnection();
    }

    public function getGridProduccion($grid){
        $this->initGrid($grid);
        $this->gridPopulate->getGridProduccion($grid);
        $grid->Process();
        $this->gridPopulate->closeConnection();
    }

    /*public function getGridH_Produccion($grid){
        $this->initGridParam($grid,false,false,false);
        $this->gridPopulate->getGridH_Produccion($grid);
        $grid->Process();
        $this->gridPopulate->closeConnection();
    }*/

    public function getGridH_Produccion($productId){
        return $this->gridPopulate->getGridH_Produccion($productId);
    }

```

```

public function getProduccionColumns($productId){
    return $this->gridPopulate->getProduccionColumns($productId);
}

public function getProductNutrientesTable($productId){
    return $this->gridPopulate->getProductNutrientesTable($productId);
}

public function getProductsData($productId){
    return $this->gridPopulate->getProductsData($productId);
}

public function getCombo_Products(){
    return $this->gridPopulate->getCombo_Products();
}

public function getProductsAtributos($productId){
    return $this->gridPopulate->getProductsAtributos($productId);
}

public function setProduccionHder($ProductId, $year){
    return $this->gridPopulate->setProduccionHder($ProductId, $year);
}

public function setProduccionDtl($produccionId,$Valor,$iEconoId){
    return $this->gridPopulate->setProduccionDtl($produccionId,$Valor,$iEconoId);
}
}
?>

```

<GridTemplates.php>

```

<?php
class NutrientesTmplate implements GridTemplate{
    private $arlNutrientes;
    function NutrientesTmplate($array){
        $this->arlNutrientes = $array;
    }
    function Render($_row){
        $tbl_teststeps = $_row->GetTableView();
    }
}

```

```

        $parent_row = $tbl_teststeps->GetParentRow();
        $value = $parent_row->DataItem["Id"];
        $html = "<input id='Producto_Id' name='Producto_Id' type='hidden'
value="" . $value . "'/>";
        $html .= "<table width='100%'>";
        $html .= "<tr>";
            $html .= "<td valign='top'>Nutriente Id:</td>";
            $html .= $this->getComboNutrientes();
            $html .= "<td valign='top'>Valor:</td>";
            $html .= "<td valign='top'><input id='Valor' name='Valor'
type='text' size='40'></td>";
            $html .= "</tr>";
            $html .= "<tr>";
                $html .= "<td colspan='4' align='right'>";
                    $html .= "<input type='button' value='Confirm'
onclick='grid_confirm_insert(this)'>"; //Render confirm button.
                    $html .= "<input type='button' value='Cancel'
onclick='grid_cancel_insert(this)'>"; //Render cancel button.
                $html .= "</td>";
            $html .= "</tr>";
            $html .= "</table>";
            return $html;
        }
        function GetData($_row){
            return
array("Producto_Id"=>$_POST["Producto_Id"],"Nutricion_Id"=>$_POST["Nutricion_Id"],"Valor"
=>$_POST["Valor"]);
        }

        function getComboNutrientes(){
            $htmlCmb = "<td valign='top'>";
            $htmlCmb .= "<select name='Nutricion_Id'>";
            foreach ($this->ariNutrientes as $combo) {
                $htmlCmb .= "<option value="" . $combo->getId() . "">" . $combo->getCode()
."</option>";
            }
            $htmlCmb .= "</select>";
            $htmlCmb .= "</td>";
            return $htmlCmb;
        }
    }
}
class LocalidadesTemplate implements GridTemplate{

```

```

private $arLocalidades;

function LocalidadesTemplate($array){
    $this->arLocalidades = $array;
}
function Render($_row){
    $tbl_teststeps = $_row->GetTableView();
    $parent_row = $tbl_teststeps->GetParentRow();
    $value = $parent_row->DataItem["Id"];
    $html = "<input id='Producto_Id' name='Producto_Id' type='hidden'
value="."$value."/>";
    $html .= "<table width='100%'>";
    $html .= "<tr>";
    $html .= "<td valign='top'>Locadalid Id:</td>";
    $html .= $this->getComboLocalidades();
    $html .= "</tr>";
    $html .= "<tr>";
        $html .= "<td colspan='4' align='right'>";
            $html .= "<input type='button' value='Confirm'
onclick='grid_confirm_insert(this)'/>"; //Render confirm button.
            $html .= "<input type='button' value='Cancel'
onclick='grid_cancel_insert(this)'/>"; //Render cancel button.
        $html .= "</td>";
    $html .= "</tr>";
    $html .= "</table>";
    return $html;
}
function GetData($_row){
    return
array("Producto_Id"=>$_POST["Producto_Id"],"Localidad_Id"=>$_POST["Localidad_Id"]);
}
function getComboLocalidades(){
    $htmlCmb = "<td valign='top'>";
    $htmlCmb .= "<select name='Localidad_Id'>";
    foreach ($this->arLocalidades as $combo) {
        $htmlCmb .= "<option value="."$combo->getId()." ">".$combo->getCode()
."</option>";
    }
    $htmlCmb .= "</select>";
    $htmlCmb .= "</td>";
    return $htmlCmb;
}
}
}

```

```

class AtributosTmplate implements GridTemplate{
    private $arlAtributos;
    function AtributosTmplate($array){
        $this->arlAtributos = $array;
    }
    function Render($_row){
        $tbl_teststeps = $_row->GetTableView();
        $parent_row = $tbl_teststeps->GetParentRow();
        $value = $parent_row->DataItem["Id"];
        $html ="<input id='Producto_Id' name='Producto_Id' type='hidden'
value="" . $value . "'/>";
        $html .="<table width='100%'>";
        $html .="<tr>";
        $html .="<td valign='top'>Atributo Id:</td>";

        $html .=" $this->getComboAtributos();
        $html .="<td valign='top'>Valor:</td>";
        $html .="<td valign='top'><input id='Valor' name='Valor' type='text'
size='40'></td>";

        $html .="</tr>";
        $html .="<tr>";
        $html .="<td colspan='4' align='right'>";
        $html .=" <input type='button' value='Confirm'
onclick='grid_confirm_insert(this)'>"; //Render confirm button.
        $html .=" <input type='button' value='Cancel'
onclick='grid_cancel_insert(this)'>"; //Render cancel button.
        $html .="</td>";
        $html .="</tr>";
        $html .="</table>";
        return $html;
    }

    function GetData($_row){
        return
array("Producto_Id"=>$_POST["Producto_Id"],"Atributo_Id"=>$_POST["Atributo_Id"],"Valor"=
>$_POST["Valor"]);
    }

    function getComboAtributos(){
        $htmlCmb ="<td valign='top'>";
        $htmlCmb .="<select name='Atributo_Id'>";
        foreach ($this->arlAtributos as $combo) {

```

```

        $htmlCmb .= "<option value='".$combo->getId()."'>".$combo->getCode()
    . "</option>";
    }
    $htmlCmb .= "</select>";
    $htmlCmb .= "</td>";
    return $htmlCmb;
}
}
class ProduccionDtlTemplate implements GridTemplate{
    private $arliEconomicos;
    function ProduccionDtlTemplate($array){
        $this->arliEconomicos = $array;
    }

    function Render($_row){
        $tbl_teststeps = $_row->GetTableView();
        $parent_row = $tbl_teststeps->GetParentRow();
        $value = $parent_row->DataItem["Id"];

        $html = "<input id='produccion_id' name='produccion_id' type='hidden'
value='".$value."'><Br/>";
        $html .= "<table width='100%'>";
        $html .= "<tr>";
        $html .= "<td valign='top'>Indicador Economico:</td>";
        $html .= $this->getComboiEconomico();
        $html .= "<td valign='top'>Valor:</td>";
        $html .= "<td valign='top'><input id='Valor' name='Valor' type='text'
size='40'></td>";
        $html .= "</tr>";
        $html .= "<tr>";
        $html .= "<td colspan='4' align='right'>";
        $html .= "<input type='button' value='Confirm'
onclick='grid_confirm_insert(this)'>"; //Render confirm button.
        $html .= "<input type='button' value='Cancel'
onclick='grid_cancel_insert(this)'>"; //Render cancel button.
        $html .= "</td>";
        $html .= "</tr>";
        $html .= "</table>";
        return $html;
    }
    function GetData($_row){

```

```

        return
array("produccion_id"=>$_POST["produccion_id"],"ieconomico_id"=>$_POST["ieconomico_id"],
"Valor"=>$_POST["Valor"]);
    }
    function getComboiEconomico(){
        $htmlCmb ="<td valign='top'>";
        $htmlCmb .="<select name='ieconomico_id'>";
        foreach ($this->arliEconomicos as $combo) {

            $htmlCmb .="<option value='".$combo->getId()."'>".$combo->getCode()
."</option>";
        }
        $htmlCmb .="</select>";
        $htmlCmb .="</td>";
        return $htmlCmb;
    }
}
?>

```

<Connection.php>

```

<?php
class Connection{
    private $host;
    private $port;
    private $user;
    private $passwd;
    private $dbName;
    function Connection(){
        $this->host='localhost';
        //$this->host='148.231.130.234';
        $this->port='3306';
        $this->user='MTIC';
        $this->passwd='mtic';
        $this->dbName='simadb';
    }
    function ConnectionNew($host,$port,$user,$passwd,$dbname){
        $this->host=$host;
        $this->port=$port;
        $this->user=$user;
        $this->passwd=$passwd;
        $this->dbName=$dbname;
    }
}

```

```

    }
    public function getMysqliConnection(){
        $con = new mysqli($this->host, $this->user,$this->paswd,$this->dbName);
        if ($con->connect_error) {
            printf("Connect failed: %s\n",$this->con->connect_error);
        }
        return $con;
    }
    public function getConnection(){
        $con = mysql_connect($this->host, $this->user,$this->paswd);
        if(!mysql_select_db($this->dbName)){
            printf("Connect failed: %s\n",mysql_error());
        }
        return $con;
    }
    public function closeConnection($con){
        if(mysql_close($con))
            return true;
        else
            return false;
    }
}
?>

```

<ComboEntity.php>

```

<?php
class ComboEntity{
    private $id;
    private $code;

    function ComboEntity($id,$code){
        $this->id = $id;
        $this->code = $code;
    }

    public function setId($id){
        $this->id = $id;
    }

    public function getId(){
        return $this->id;
    }
}

```

```

    }

    public function setCode($code){
        $this->code = $code;
    }

    public function getCode(){
        return $this->code;
    }
}
?>

```

<UserEntity.php>

```

<?php
include("../beans/biz/CommonBiz.php");
class UserEntity{
    private $id;
    private $user_name;
    private $user_pswd;

    function UserEntity($user_name,$user_pswd){
        $this->user_name = $user_name;
        $this->user_pswd = $user_pswd;
    }
    public function setUserName($user_name){
        $this->user_name = $user_name;
    }
    public function getUserId(){
        return $this->id;
    }
    public function setUserId($userID){
        $this->id = $userID;
    }
    public function getUserName(){
        return $this->user_name;
    }
    public function setUserPswd($user_pswd){
        $this->user_pswd = $user_pswd;
    }
    public function getUserPswd(){
        return $this->user_pswd;
    }
}

```

```

    }
    public function getValidaUserNew($user_name,$user_pswd){
        $biz = new CommonBiz();
        return $biz->getValidaUser($user_name, $user_pswd);
    }
    public function getValidaUser(){
        $biz = new CommonBiz();
        return $biz->getValidaUser($this->user_name, $this->user_pswd);
    }
}
?>

```

<CommonPopulate.php>

```

<?php
include("../beans/helper/CommonHelper.php");

class CommonPopulate{
    private $conDB;
    private $helper;
    private $stmt;
    private $con;

    function CommonPopulate(){
        $this->conDB = new Connection();
        $this->helper = new CommonHelper();
    }

    function __destruct() {
        unset($this->helper);
        unset($this->conDB);
    }

    public function getCombo_ClimaSP(){
        $arlCombo = array();
        $combo = null;
        if($con = $this->conDB->getMysqlConnection()){
            if($rs = $this->helper->getCombo_ClimaSP($con)){

                while ($row = $rs->fetch_array(MYSQLI_BOTH)) {

```

```

        $combo = new ComboEntity($row["Id"],$row["Nombre"]);

        $arlCombo[] = $combo;
    }
    $rs->close();
}
$con->close();
}
return $arlCombo;
}

public function getCombo_LocalidadSP(){
    $arlCombo = array();
    $combo = null;
    if($con = $this->conDB->getMysqlConnection()){
        if($rs = $this->helper->getCombo_LocalidadSP($con)){

            while ($row = $rs->fetch_array(MYSQLI_BOTH)) {

                $combo = new ComboEntity($row["Id"],$row["Nombre"]);

                $arlCombo[] = $combo;
            }
            $rs->close();
        }
        $con->close();
    }
    return $arlCombo;
}

public function getCombo_ZonaSP(){
    $arlCombo = array();
    $combo = null;
    if($con = $this->conDB->getMysqlConnection()){
        if($rs = $this->helper->getCombo_ZonaSP($con)){

            while ($row = $rs->fetch_array(MYSQLI_BOTH)) {

                $combo = new ComboEntity($row["Id"],$row["Nombre"]);

                $arlCombo[] = $combo;
            }
            $rs->close();
        }
    }
}

```

```

        }
        $con->close();
    }
    return $arlCombo;
}

public function getCombo_MunicipiosSP(){
    $arlCombo = array();
    $combo = null;
    if($con = $this->conDB->getMysqliConnection()){
        if($rs = $this->helper->getCombo_MunicipiosSP($con)){

            while ($row = $rs->fetch_array(MYSQLI_BOTH)) {

                $combo = new ComboEntity($row["Id"],$row["Nombre"]);

                $arlCombo[] = $combo;
            }
            $rs->close();
        }
        $con->close();
    }
    return $arlCombo;
}

public function getCombo_ProductosSP(){
    $arlCombo = array();
    $combo = null;
    if($con = $this->conDB->getMysqliConnection()){
        if($rs = $this->helper->getCombo_ProductosSP($con)){

            while ($row = $rs->fetch_array(MYSQLI_BOTH)) {
                $combo = new
ComboEntity($row["Id"],$row["Producto"]);
                $arlCombo[] = $combo;
            }
            $rs->close();
        }
        $con->close();
    }
    return $arlCombo;
}
}

```

```

public function getCombo_ProductTypeSP(){
    $arlCombo = array();
    $combo = null;
    if($con = $this->conDB->getMysqliConnection()){
        if($rs = $this->helper->getCombo_ProductTypeSP($con)){

            while ($row = $rs->fetch_array(MYSQLI_BOTH)) {

                $combo = new ComboEntity($row["Id"],$row["Codigo"]);

                $arlCombo[] = $combo;
            }
            $rs->close();
        }
        $con->close();
    }
    return $arlCombo;
}

public function getCombo_AtributosSP(){
    $arlCombo = array();
    $combo = null;
    if($con = $this->conDB->getMysqliConnection()){
        if($rs = $this->helper->getCombo_AtributosSP($con)){

            while ($row = $rs->fetch_array(MYSQLI_BOTH)) {

                $combo = new ComboEntity($row["Id"],$row["Codigo"]);

                $arlCombo[] = $combo;
            }
            $rs->close();
        }
        $con->close();
    }
    return $arlCombo;
}

public function getCombo_IEconomicosSP(){
    $arlCombo = array();
    $combo = null;
    if($con = $this->conDB->getMysqliConnection()){

```

```

        if($rs = $this->helper->getCombo_IEconomicosSP($con)){

            while ($row = $rs->fetch_array(MYSQLI_BOTH)) {

                $combo = new ComboEntity($row["Id"],$row["Nombre"]);

                $arlCombo[] = $combo;

            }
            $rs->close();
        }
        $con->close();
    }
    return $arlCombo;
}

public function getCombo_NutrientesSP(){
    $arlCombo = array();
    $combo = null;
    if($con = $this->conDB->getMysqlConnection()){
        if($rs = $this->helper->getCombo_NutrientesSP($con)){

            while ($row = $rs->fetch_array(MYSQLI_BOTH)) {

                $combo = new ComboEntity($row["Id"],$row["Nombre"]);

                $arlCombo[] = $combo;

            }
            $rs->close();
        }
        $con->close();
    }
    return $arlCombo;
}

public function getUserValidate($user_name,$user_pswd){
    $bResult = 0;
    if($con = $this->conDB->getMysqlConnection()){
        if($this->helper->getUserValidate($con,$user_name,$user_pswd)){
            do {
                if ($rs = $con->store_result() ) {
                    while ($row = $rs->fetch_array(MYSQLI_BOTH))
                    {
                        if(!is_null($row["@userId"]))

```

```

        $bResult = $row["@userId"];
    }
    $rs->close();
}
}while($con->more_results() && $con->next_result());
}
$con->close();
}
return $bResult;
}

public function setProduccionHder($ProductId,$year){
    $bResult = 0;
    if($con = $this->conDB->getMysqlConnection()){
        if($this->helper->setProduccionHder($con,$ProductId,$year)){
            do {
                if ($rs = $con->store_result() {
                    while ($row = $rs->fetch_array(MYSQLI_BOTH))
                    {
                        if(!is_null($row["@produccionId"]))
                            $bResult =
$row["@produccionId"];
                    }
                    $rs->close();
                }
            }while($con->more_results() && $con->next_result());
        }
        $con->close();
    }
    return $bResult;
}

public function setProduccionDtl($produccionId,$Valor,$iEconoId){
    $bResult = 0;
    if($con = $this->conDB->getMysqlConnection()){
        if($this->helper-
>setProduccionDtl($con,$produccionId,$Valor,$iEconoId)){
            do {
                if ($rs = $con->store_result() {
                    while ($row = $rs->fetch_array(MYSQLI_BOTH))
                    {
                        if(!is_null($row["@prodDtlId"]))
                            $bResult = $row["@prodDtlId"];
                    }
                }
            }while($con->more_results() && $con->next_result());
        }
        $con->close();
    }
    return $bResult;
}

```

```

        }
        $rs->close();
    }
    } while($con->more_results() && $con->next_result());
    }
    $con->close();
}
return $bResult;
}

public function getH_ProduccionDS($productId){
    $arlRow = array();
    $arl = array();
    $year = 0;
    $combo = null;
    if($con = $this->conDB->getMysqlConnection()){
        if($rs = $this->helper->get_H_ProduccionQ($con,$productId)){

            while ($row = $rs->fetch_array(MYSQLI_BOTH)) {

                if($year==0 || $year!=$row["year"]){

                    if($year!=0){
                        array_push($arl,$arlRow);
                    }
                    $arlRow = array();
                    array_push($arlRow,$row["year"]);
                    $year = $row["year"];
                }
                array_push($arlRow,$row["Valor"]);
            }
            $rs->close();
        }
        array_push($arl,$arlRow);
        $con->close();
    }
    return $arl;
}

public function getProduccionColumns($productId){
    $arl = array();
    $year = 0;
    $combo = null;

```

```

        if($con = $this->conDB->getMysqliConnection()){
            if($rs = $this->helper->getProduccionColumn($con,$productId)){

                while ($row = $rs->fetch_array(MYSQLI_BOTH)) {
                    array_push($arl,$row["Nombre"]."
(".$row["Unidades"].")");
                }
                $rs->close();
            }
            $con->close();
        }
        return $arl;
    }

    public function getProductsAtributos($productId){
        $arlCombo = array();
        $combo = null;
        if($con = $this->conDB->getMysqliConnection()){
            if($rs = $this->helper->getProductsAtributos($con,$productId)){

                while ($row = $rs->fetch_array(MYSQLI_BOTH)) {
                    $combo = new
ComboEntity($row["codigo"],$row["Valor"]);
                    $arlCombo[] = $combo;
                }
                $rs->close();
            }
            $con->close();
        }
        return $arlCombo;
    }

    public function getProductsData($productId){
        $arlCombo = array();
        $product = null;
        if($con = $this->conDB->getMysqliConnection()){
            if($rs = $this->helper->getProductsData($con, $productId)){

                while ($row = $rs->fetch_array(MYSQLI_BOTH)) {

                    array_push($arlCombo,$row["Producto"],$row["Nom_C"],$row["Descripcion"]);
                }
            }
        }
    }

```

```

        }
        $rs->close();
    }
    $con->close();
}
return $arlCombo;
}

public function getProductNutrientesTable($productId){
    $arl = array();
    $arlRow = null;

    if($con = $this->conDB->getMysqlConnection()){
        if($rs = $this->helper->getProductNutrientesTable($con,$productId)){

            while ($row = $rs->fetch_array(MYSQLI_BOTH)) {
                $arlRow = array();
                $arlRow["Id"] = $row["Id"];
                $arlRow["Nutriente"] = $row["Nombre"];
                $arlRow["Valor"] = $row["Valor"];
                $arlRow["Unidades"] = $row["Unidades"];

                array_push($arl,$arlRow);
            }
            $rs->close();
        }
        $con->close();
    }
    return $arl;
}
}
?>

```

<GridPopulate.php>

```

<?php
include("../beans/helper/GridHelper.php");
include("../beans/populate/CommonPopulate.php");

class GridPopulate{

    private $conDB;

```

```

private $con;
private $helper;
private $commonPopulate;

function GridPopulate(){

    $this->conDB = new Connection();
    $this->helper = new GridHelper();
    $this->commonPopulate = new CommonPopulate();
}

function __destruct() {
    unset($this->helper);
    unset($this->commonPopulate);
    unset($this->conDB);
}

function closeConnection(){
    $this->conDB->closeConnection($this->con);
}

private function getGridColumn($HeaderText,$DataField,$isReadO){
    $column = new GridBoundColumn();
    $column->HeaderText = $HeaderText;
    $column->DataField = $DataField;
    $column->ReadOnly = $isReadO;
    return $column;
}

private function getGridColumnFilter($HeaderText,$DataField,$isReadO){
    $column = new GridBoundColumn();
    $column->FilterOptions = array("Contain");
    //$column->Filter = array("Value"=>"Atun","Exp"=>"Contain");
    $column->HeaderText = $HeaderText;
    $column->DataField = $DataField;
    $column->ReadOnly = $isReadO;
    $column->AllowFiltering = true;
    $column->Width = "25%";

    return $column;
}

private function getGridTextAreaColumn($HeaderText,$DataField,$isReadO){

```

```

        $column = new GridTextAreaColumn();
        $column->HeaderText = $HeaderText;
        $column->DataField = $DataField;
        $column->ReadOnly = $isReadO;
        return $column;
    }

    private function getGridColumnValid($HeaderText,$DataField,$Validator){
        $column = new GridBoundColumn();
        $column->HeaderText = $HeaderText;
        $column->DataField = $DataField;
        $column->AddValidator($Validator);
        return $column;
    }

    private function getGridColumnCombo($HeaderText,$DataField,$isReadO,$arlCombo){
        $column = new GridDropDownColumn();
        $column->HeaderText = $HeaderText;
        $column->DataField = $DataField;
        $column->ReadOnly = $isReadO;
        $column->AddItem("Select",0);
        foreach ($arlCombo as $combo) {
            $column->AddItem($combo->getCode(),$combo->getId());
        }
        return $column;
    }

    public function getGridProducts($grid){
        if($this->con = $this->conDB->getConnection()){
            $ds = new MySQLDataSource($this->con);
            $this->helper->getGridProductsDS($ds);
            $grid->DataSource = $ds;
            $grid->MasterTable->AddColumn($this->getGridColumn("ID","Id",true));
            $grid->MasterTable->AddColumn($this->getGridColumn("Producto","Producto",false));
            $grid->MasterTable->AddColumn($this->getGridColumn("Nombre Cientifico","Nom_C",false));
            $grid->MasterTable->AddColumn($this->getGridColumnCombo("Tipo de Producto","TProducto_Id",false,$this->commonPopulate->getCombo_ProductTypeSP()));
            $grid->MasterTable->AddColumn($this->getGridTextAreaColumn("Descripcion","Descripcion",false));

```

```

        // $grid->MasterTable->AddColumn($this->
        >getGridColumnCombo("Localidad", "Localidad_Id", false, $this->commonPopulate->
        >getCombo_LocalidadSP()));

        $column = new GridEditDeleteColumn();
        $column->Width = "15%";
        $grid->MasterTable->AddColumn($column);

        $ds = new MySQLDataSource($this->con);
        $this->helper->getGridProductsDS($ds);
    }
}

public function getGridClima($grid){
    if($this->con = $this->conDB->getConnection()){
        $ds = new MySQLDataSource($this->con);
        $this->helper->getGridClimaDS($ds);
        $grid->DataSource = $ds;
        $grid->MasterTable->AddColumn($this->getGridColumn("ID", "Id", true));
        $grid->MasterTable->AddColumn($this->
        >getGridColumn("Nombre", "Nombre", false));
        $grid->MasterTable->AddColumn($this->
        >getGridColumn("Descripcion", "Descripcion", false));
        $column = new GridEditDeleteColumn();
        $column->Width = "15%";
        $grid->MasterTable->AddColumn($column);
    }
}

public function getGridProductType($grid){
    if($this->con = $this->conDB->getConnection()){
        $ds = new MySQLDataSource($this->con);
        $this->helper->getGridProductTypeDS($ds);
        $grid->DataSource = $ds;
        $grid->MasterTable->AddColumn($this->getGridColumn("ID", "Id", true));
        $grid->MasterTable->AddColumn($this->
        >getGridColumn("Codigo", "Codigo", false));
        $grid->MasterTable->AddColumn($this->
        >getGridColumn("Descripcion", "Descripcion", false));
        $column = new GridEditDeleteColumn();
        $column->Width = "15%";
        $grid->MasterTable->AddColumn($column);
    }
}

```

```

    }
}
public function getGridMunicipio($grid){
    if($this->con = $this->conDB->getConnection()){
        $ds = new MySQLDataSource($this->con);
        $this->helper->getGridMunicipioDS($ds);
        $grid->DataSource = $ds;
        $grid->MasterTable->AddColumn($this->getGridColumn("ID","Id",true));
        $grid->MasterTable->AddColumn($this-
>getGridColumn("Nombre","Nombre",false));
        $grid->MasterTable->AddColumn($this-
>getGridColumn("Estado","Estado",false));
        $validator = new RegularExpressionValidator();
        $validator->ValidationExpression = "/^([0-9])+$/";
        $validator->ErrorMessage = "El Valor debe ser Numerico";
        $grid->MasterTable->AddColumn($this-
>getGridColumnValid("Habitantes","Habitantes",$validator));
        $column = new GridEditDeleteColumn();
        $column->Width = "15%";
        $grid->MasterTable->AddColumn($column);
    }
}

public function getGridI_Economicos($grid){
    if($this->con = $this->conDB->getConnection()){
        $ds = new MySQLDataSource($this->con);
        $this->helper->getGridI_EconomicoDS($ds);
        $grid->DataSource = $ds;
        $grid->MasterTable->AddColumn($this->getGridColumn("ID","Id",true));
        $grid->MasterTable->AddColumn($this-
>getGridColumn("Nombre","Nombre",false));
        $grid->MasterTable->AddColumn($this-
>getGridColumn("Descripcion","Descripcion",false));
        $grid->MasterTable->AddColumn($this-
>getGridColumn("Unidades","Unidades",false));
        $column = new GridEditDeleteColumn();
        $column->Width = "15%";
        $grid->MasterTable->AddColumn($column);
    }
}

public function getGridNutricion($grid){
    if($this->con = $this->conDB->getConnection()){

```

```

        $ds = new MySQLDataSource($this->con);
        $this->helper->getGridNutricionDS($ds);
        $grid->DataSource = $ds;
        $grid->MasterTable->AddColumn($this->getGridColumn("ID", "Id", true));
        $grid->MasterTable->AddColumn($this-
>getGridColumn("Nombre", "Nombre", false));
        $grid->MasterTable->AddColumn($this-
>getGridColumn("Descripcion", "Descripcion", false));
        $grid->MasterTable->AddColumn($this-
>getGridColumn("Unidades", "Unidades", false));
        $column = new GridEditDeleteColumn();
        $column->Width = "15%";
        $grid->MasterTable->AddColumn($column);

    }

}

public function getGridLocalidades($grid){
    if($this->con = $this->conDB->getConnection()){
        $ds = new MySQLDataSource($this->con);
        $this->helper->getGridLocalidadesDS($ds);
        $grid->DataSource = $ds;
        $grid->MasterTable->AddColumn($this->getGridColumn("ID", "Id", true));
        $grid->MasterTable->AddColumn($this-
>getGridColumn("Nombre", "Nombre", false));
        $grid->MasterTable->AddColumn($this-
>getGridColumnCombo("Zona", "Zona_Id", false, $this->commonPopulate->getCombo_ZonaSP()));
        //$grid->MasterTable->AddColumn($this-
>getGridColumn("Latitud", "Latitud", false));
        //$grid->MasterTable->AddColumn($this-
>getGridColumn("Longitud", "Longitud", false));
        $grid->MasterTable->AddColumn($this-
>getGridColumnCombo("Clima", "Clima_Id", false, $this->commonPopulate-
>getCombo_ClimaSP()));
        $grid->MasterTable->AutoGenerateExpandColumn = true;
        $column = new GridEditDeleteColumn();
        $column->Width = "15%";
        $grid->MasterTable->AddColumn($column);
        $this->getTableV_Zona($grid);

    }

}

private function getGridBase_Products($grid){

```

```

        $ds = new MySQLDataSource($this->con);
        $this->helper->getGridProductsDS($ds);
        $grid->DataSource = $ds;
        $grid->MasterTable->AddColumn($this->getGridColumn("ID","Id",true));
        $grid->MasterTable->AddColumn($this-
>getGridColumn("Producto","Producto",true));
        $grid->MasterTable->AddColumn($this-
>getGridTextAreaColumn("Descripcion","Descripcion",false));
        $grid->MasterTable->AddColumn($this->getGridColumnCombo("Tipo de
Producto","TProducto_Id",false,$this->commonPopulate->getCombo_ProductTypeSP()));
        $grid->MasterTable->AutoGenerateExpandColumn = true;
    }

    private function getTableProduccionDtl($grid){
        $table = new GridTableView();

        $arliEconomicos = $this->commonPopulate->getCombo_IEconomicosSP();
        $table->Width = "100%";
        $ds = new MySQLDataSource($this->con);
        $this->helper->getGridProduccionDtlDS($ds);
        $table->DataSource = $ds;
        $table->AddRelationField("Produccion_Id","Id");
        $table->AddColumn($this->getGridColumn("Id","produccion_id",true));
        $table->AddColumn($this-
>getGridColumnCombo("Indicador","ieconomico_id",true,$arliEconomicos));
        $table->AddColumn($this->getGridColumn("Valor","Valor",false));
        $column = new GridEditDeleteColumn();
        $column->Width = "15%";
        $table->AddColumn($column);
        $table->AllowEditing = true;
        $table->AllowDeleting = true;
        $table->RowAlternative = true;
        $table->AllowSorting = true;
        $table->Pager = new GridPrevNextAndNumericPager();
        $table->ColumnWrap = true;
        $table->styleFolder = "default";
        $table->ShowFunctionPanel = true;
        $table->InsertSettings->Mode = "Template";
        include("../beans/biz/GridTemplates.php");
        $table->InsertSettings->Template = new ProduccionDtlTemplate($arliEconomicos);
        $grid->MasterTable->AddDetailTable($table);
    }

```

```

private function getTableV_Zona($grid){
    $table = new GridTableView();
    $table->Width = "100%";
    $ds = new MySQLDataSource($this->con);
    $this->helper->getGridV_ZonaDS($ds);
    $table->DataSource = $ds;
    $table->AddRelationField("Id","Zona_Id");
    $table->AddColumn($this->getGridColumn("Nombre","Nombre",false));
    $table->AddColumn($this->getGridColumn("Descripcion","Descripcion",false));
    $table->AddColumn($this->getGridColumn("Area
Geografica","Area_Geo",false));
    $table->AddColumn($this-
>getGridColumnCombo("Municipio","Municipio_Id",false,$this->commonPopulate-
>getCombo_MunicipiosSP()));
    $grid->MasterTable->AddDetailTable($table);
}
public function getGridAtributos($grid){
    if($this->con = $this->conDB->getConnection()){
        $ds = new MySQLDataSource($this->con);
        $this->helper->getGridAtributosDS($ds);
        $grid->DataSource = $ds;
        $grid->MasterTable->AddColumn($this->getGridColumn("ID","Id",true));
        $grid->MasterTable->AddColumn($this-
>getGridColumn("Codigo","Codigo",false));
        $grid->MasterTable->AddColumn($this-
>getGridColumn("Descripcion","Descripcion",false));

        $column = new GridEditDeleteColumn();
        $column->Width = "15%";
        $grid->MasterTable->AddColumn($column);
    }
}
public function getGridV_Zona($grid){
    if($this->con = $this->conDB->getConnection()){
        $ds = new MySQLDataSource($this->con);
        $this->helper->getGridV_ZonaDS($ds);
        $grid->DataSource = $ds;
        $grid->MasterTable->AddColumn($this->getGridColumn("ID","Id",true));
        $grid->MasterTable->AddColumn($this-
>getGridColumn("Nombre","Nombre",false));
        $grid->MasterTable->AddColumn($this-
>getGridColumn("Descripcion","Descripcion",false));

```

```

        $grid->MasterTable->AddColumn($this->getGridColumn("Area
Geografica","Area_Geo",false));
        $grid->MasterTable->AddColumn($this-
>getGridColumnCombo("Municipio","Municipio_Id",false,$this->commonPopulate-
>getCombo_MunicipiosSP()));
        $column = new GridEditDeleteColumn();
        $column->Width = "15%";
        $grid->MasterTable->AddColumn($column);

    }
}

public function getGridProdNutrientes($grid){
    if($this->con = $this->conDB->getConnection()){
        $arlcombo = $this->commonPopulate->getCombo_NutrientesSP();
        $table = new GridTableView();
        $table->Width = "100%";
        $ds = new MySQLDataSource($this->con);
        $this->helper->getProdNutrientes($ds);
        $table->AllowEditing = true;
        $table->AllowDeleting = true;
        $table->ShowFunctionPanel = true;
        $table->InsertSettings->Mode = "Template";
        include("../beans/biz/GridTemplates.php");
        $table->InsertSettings->Template = new NutrientesTpl($arlcombo);
        $table->DataSource = $ds;
        $table->AddRelationField("Producto_Id","Id");
        $table->AddColumn($this-
>getGridColumnCombo("Nutriente","Nutricion_Id",true,$arlcombo));
        $table->AddColumn($this->getGridColumn("Valor","Valor",false));
        $column = new GridEditDeleteColumn();
        $column->Width = "15%";
        $table->AddColumn($column);
        $this->getGridBase_Products($grid);
        $grid->MasterTable->AddDetailTable($table);
    }
}

public function getGridProdAtributos($grid){
    if($this->con = $this->conDB->getConnection()){
        $arlCombo = $this->commonPopulate->getCombo_AtributosSP();
        $table = new GridTableView();
        $table->Width = "100%";
        $ds = new MySQLDataSource($this->con);

```

```

        $this->helper->getProdAtributos($ds);
        $table->AllowEditing = true;
        $table->AllowDeleting = true;
        $table->ShowFunctionPanel = true;
        $table->InsertSettings->Mode = "Template";
        include("../beans/biz/GridTemplates.php");
        $table->InsertSettings->Template = new AtributosTmplate($arlCombo);

        $table->DataSource = $ds;
        $table->AddRelationField("Producto_Id", "Id");
        $table->AddColumn($this-
>getGridColumnCombo("Atributo", "Atributo_Id", true, $arlCombo));
        $table->AddColumn($this->GridColumn("Valor", "Valor", false));
        $column = new GridEditDeleteColumn();
        $column->Width = "15%";
        $table->AddColumn($column);
        $this->getGridBase_Products($grid);
        $grid->MasterTable->AddDetailTable($table);
    }
}

public function getGridProdLocalidades($grid){
    if($this->con = $this->conDB->getConnection()){
        $arlCombo = $this->commonPopulate->getCombo_LocalidadSP();
        $table = new GridTableView();
        $table->Width = "100%";
        $ds = new MySQLDataSource($this->con);
        $this->helper->getProdLocalidades($ds);
        $table->AllowEditing = true;
        $table->AllowDeleting = true;
        $table->ShowFunctionPanel = true;
        $table->InsertSettings->Mode = "Template";
        include("../beans/biz/GridTemplates.php");
        $table->InsertSettings->Template = new LocalidadesTmplate($arlCombo);
        $table->DataSource = $ds;
        $table->AddRelationField("Producto_Id", "Id");
        $table->AddColumn($this-
>getGridColumnCombo("Localidad", "Localidad_Id", true, $arlCombo));
        $column = new GridEditDeleteColumn();
        $column->ShowEditButton = false;
        $column->Width = "15%";
        $table->AddColumn($column);
        $this->getGridBase_Products($grid);
    }
}

```

```

        $grid->MasterTable->AddDetailTable($table);
    }
}

public function getGridProduccion($grid){
    if($this->con = $this->conDB->getConnection()){
        $ds = new MySQLDataSource($this->con);
        $this->helper->getGridProduccionDS($ds);
        $grid->DataSource = $ds;
        $grid->MasterTable->AddColumn($this->getGridColumn("ID","Id",true));
        $grid->MasterTable->AddColumn($this-
>getGridColumnCombo("Producto","Producto_Id",false,$this->commonPopulate-
>getCombo_ProductosSP()));

        $validator = new RangeValidator();
        $validator->MinValue = 1950;
        $validator->MaxValue = 2020;
        $validator->ErrorMessage = "Año no Valido!...";

        $grid->MasterTable->AddColumn($this-
>getGridColumnValid("Año","Year",$validator));
        $grid->MasterTable->AutoGenerateExpandColumn = true;
        $this->getTableProduccionDtl($grid);

        $column = new GridEditDeleteColumn();
        $column->Width = "15%";
        $grid->MasterTable->AddColumn($column);
    }
}

public function getCombo_Products(){
    return $this->commonPopulate->getCombo_ProductosSP();
}

public function getProductsAtributos($productId){
    return $this->commonPopulate->getProductsAtributos($productId);
}

public function getGridH_Produccion($productId){
    return $this->commonPopulate->getH_ProduccionDS($productId);
}

public function getProduccionColumns($productId){

```

```

        return $this->commonPopulate->getProduccionColumns($productId);
    }

    public function getProductNutrientesTable($productId){
        return $this->commonPopulate->getProductNutrientesTable($productId);
    }

    public function getProductsData($productId){
        return $this->commonPopulate->getProductsData($productId);
    }

    public function setProduccionHder($ProductId,$year){
        return $this->commonPopulate->setProduccionHder($ProductId, $year);
    }

    public function setProduccionDtl($produccionId,$Valor,$iEconoId){
        return $this->commonPopulate-
>setProduccionDtl($produccionId,$Valor,$iEconoId);
    }

    private function makeH_ProduccionDS($grid){
        $array_ds = $this->commonPopulate->getH_ProduccionDS();

        $ds = new ArrayDataSource($array_ds[1]);
        $grid->DataSource = $ds;
        return $array_ds[0];
    }
}

```

<CommonHelper.php>

```

<?php
class CommonHelper{

    function CommonHelper(){
    }

    public function getCombo_ProductosSP($con){

```

```

        $sqlQuery = "CALL getCombo_Productos()";
        if(!$rsResult = $con->query($sqlQuery)){
            printf("Error on execute query: %s\n", $mysqli->error);
        }
        return $rsResult;
    }

    public function getCombo_AtributosSP($con){
        $sqlQuery = "CALL getCombo_Atributos()";
        if(!$rsResult = $con->query($sqlQuery)){
            printf("Error on execute query: %s\n", $mysqli->error);
        }
        return $rsResult;
    }

    public function getCombo_IEconomicosSP($con){
        $sqlQuery = "CALL getCombo_iEconomicos()";
        if(!$rsResult = $con->query($sqlQuery)){
            printf("Error on execute query: %s\n", $mysqli->error);
        }
        return $rsResult;
    }

    public function getCombo_NutrientesSP($con){
        $sqlQuery = "CALL getCombo_Nutrientes()";
        if(!$rsResult = $con->query($sqlQuery)){
            printf("Error on execute query: %s\n", $mysqli->error);
        }
        return $rsResult;
    }

    public function getCombo_ZonaSP($con){
        $sqlQuery = "CALL getCombo_Zonas()";
        if(!$rsResult = $con->query($sqlQuery)){
            printf("Error on execute query: %s\n", $mysqli->error);
        }
        return $rsResult;
    }

    public function getCombo_LocalidadSP($con){
        $sqlQuery = "CALL getCombo_Localidad()";
        if(!$rsResult = $con->query($sqlQuery)){
            printf("Error on execute query: %s\n", $mysqli->error);
        }
    }

```

```

    }
    return $rsResult;
}

public function getCombo_ClimaSP($con){
    $sqlQuery = "CALL getCombo_Clima()";
    if(!$rsResult = $con->query($sqlQuery)){
        printf("Error on execute query: %s\n", $mysqli->error);
    }
    return $rsResult;
}

public function getCombo_ProductTypeSP($con){
    $sqlQuery = "CALL getCombo_ProdType()";
    if(!$rsResult = $con->query($sqlQuery)){
        printf("Error on execute query: %s\n", $mysqli->error);
    }
    return $rsResult;
}

public function getCombo_MunicipiosSP($con){
    $sqlQuery = "CALL getCombo_Municipios()";
    if(!$rsResult = $con->query($sqlQuery)){
        printf("Error on execute query: %s\n", $mysqli->error);
    }
    return $rsResult;
}

public function get_H_ProduccionQ($con,$productId){
    $sqlQuery = "CALL get_H_ProduccionDI(".$productId.)";
    if(!$rsResult = $con->query($sqlQuery)){
        printf("Error on execute query: %s\n", $mysqli->error);
    }
    return $rsResult;
}

public function getProduccionColumn($con,$productId){
    $sqlQuery = "CALL getProduccionColumns(".$productId.)";
    if(!$rsResult = $con->query($sqlQuery)){
        printf("Error on execute query: %s\n", $mysqli->error);
    }
    return $rsResult;
}

```

```

public function getProductNutrientesTable($con,$productId){
    $sqlQuery = "CALL getPTablaNutrientes('.$productId.')";
    if(!$rsResult = $con->query($sqlQuery)){
        printf("Error on execute query: %s\n", $mysqli->error);
    }
    return $rsResult;
}

public function getProductsData($con,$productId){
    $sqlQuery = "CALL getProducts_byId('.$productId.')";
    if(!$rsResult = $con->query($sqlQuery)){
        printf("Error on execute query: %s\n", $mysqli->error);
    }
    return $rsResult;
}

public function getProductsAtributos($con,$productId){
    $sqlQuery = "CALL getProductsAtributs('.$productId.')";
    if(!$rsResult = $con->query($sqlQuery)){
        printf("Error on execute query: %s\n", $mysqli->error);
    }
    return $rsResult;
}

public function getUserValidate($con,$user_name,$user_pswd){
    $sqlQuery = "CALL getValidUser_login('$user_name','$user_pswd',@userId);
SELECT @userId;";
    if(!$rsResult = $con->multi_query($sqlQuery)){
        printf("Error on execute query: %s\n", $mysqli->error);
    }
    return $rsResult;
}

public function setProduccionHder($con,$ProductId,$year){
    $sqlQuery = "CALL setProduccionHdr($ProductId,$year,@produccionId);
SELECT @produccionId;";
    if(!$rsResult = $con->multi_query($sqlQuery)){
        printf("Error on execute query: %s\n", $mysqli->error);
    }
    return $rsResult;
}

```

```

        public function setProduccionDtl($con,$produccionId,$Valor,$iEconoId){
            $sqlQuery = "CALL
setProduccionDtl($produccionId,'$Valor',$iEconoId,@prodDtlId); SELECT @prodDtlId";
            if(!$rsResult = $con->multi_query($sqlQuery)){
                printf("Error on execute query: %s\n", $mysqli->error);
            }
            return $rsResult;
        }
    }
}
?>

```

<GridHelper.php>

```

<?php
class GridHelper{

    function GridHelper(){
    }

    public function getGridMunicipioDS($ds){
        $ds->SelectCommand = "SELECT * FROM municipio";
        $ds->UpdateCommand = "UPDATE municipio SET Nombre='@Nombre',
Estado='@Estado', Habitantes='@Habitantes' WHERE Id=@Id";
        $ds->DeleteCommand = "DELETE FROM municipio WHERE Id=@Id";
        $ds->InsertCommand = "INSERT INTO municipio(Id,Nombre,Estado,Habitantes)
VALUES (NULL,'@Nombre','@Estado','@Habitantes)";
    }

    public function getGridLocalidadesDS($ds){
        $ds->SelectCommand = "SELECT * FROM localidades";
        $ds->UpdateCommand = "UPDATE localidades SET
Nombre=@Nombre,Zona_Id=@Zona_Id,Clima_Id=@Clima_Id WHERE Id=@Id";
        $ds->DeleteCommand = "DELETE FROM localidades WHERE Id=@Id";
        $ds->InsertCommand = "INSERT INTO
localidades(Id,Nombre,Zona_Id,Clima_Id) VALUES (NULL,@Nombre,@Zona_Id,'@Clima_Id)";
    }

    public function getGridProductTypeDS($ds){
        $ds->SelectCommand = "SELECT * FROM tipo_producto";
        $ds->UpdateCommand = "UPDATE tipo_producto SET Codigo='@Codigo',
Descripcion='@Descripcion' WHERE Id=@Id";
    }
}

```

```

        $ds->DeleteCommand = "DELETE FROM tipo_producto WHERE Id=@Id";
        $ds->InsertCommand = "INSERT INTO tipo_producto(Id,Codigo,Descripcion)
VALUES (NULL,'@Codigo','@Descripcion)";
    }

    public function getGridProductsDS($ds){
        $ds->SelectCommand = "SELECT * FROM productos";
        $ds->UpdateCommand = "UPDATE productos SET Producto='@Producto',
Nom_C='@Nom_C',Descripcion='@Descripcion',
        TProducto_Id=@TProducto_Id WHERE Id=@Id";
        $ds->DeleteCommand = "DELETE FROM productos WHERE Id=@Id";
        $ds->InsertCommand = "INSERT INTO
productos(Id,Producto,Nom_C,Descripcion,TProducto_Id)
VALUES
(NULL,'@Producto','@Nom_C','@Descripcion',@TProducto_Id)";
    }

    public function getGridClimaDS($ds){
        $ds->SelectCommand = "SELECT * FROM clima";
        $ds->UpdateCommand = "UPDATE clima SET Nombre='@Nombre',
Descripcion='@Descripcion' WHERE Id=@Id";
        $ds->DeleteCommand = "DELETE FROM clima WHERE Id=@Id";
        $ds->InsertCommand = "INSERT INTO clima(Id,Nombre,Descripcion) VALUES
(NULL,'@Nombre','@Descripcion)";
    }

    public function getGridI_EconomicoDS($ds){
        $ds->SelectCommand = "SELECT * FROM indicador_economico";
        $ds->UpdateCommand = "UPDATE indicador_economico SET
Nombre='@Nombre', Descripcion='@Descripcion', Unidades='@Unidades' WHERE Id=@Id";
        $ds->DeleteCommand = "DELETE FROM indicador_economico WHERE
Id=@Id";
        $ds->InsertCommand = "INSERT INTO
indicador_economico(Id,Nombre,Descripcion,Unidades) VALUES
(NULL,'@Nombre','@Descripcion','@Unidades)";
    }

    public function getGridNutricionDS($ds){
        $ds->SelectCommand = "SELECT * FROM nutricion";
        $ds->UpdateCommand = "UPDATE nutricion SET Nombre='@Nombre',
Descripcion='@Descripcion', Unidades='@Unidades' WHERE Id=@Id";
        $ds->DeleteCommand = "DELETE FROM nutricion WHERE Id=@Id";
    }

```

```

        $ds->InsertCommand = "INSERT INTO
nutricion(Id,Nombre,Descripcion,Unidades) VALUES
(NULL,'@Nombre','@Descripcion','@Unidades)";
    }

    public function getGridAtributosDS($ds){
        $ds->SelectCommand = "SELECT * FROM atributos";
        $ds->UpdateCommand = "UPDATE atributos SET Codigo='@Codigo',
Descripcion='@Descripcion' WHERE Id=@Id";
        $ds->DeleteCommand = "DELETE FROM atributos WHERE Id=@Id";
        $ds->InsertCommand = "INSERT INTO atributos(Id,Codigo,Descripcion)
VALUES (NULL,'@Codigo','@Descripcion)";
    }

    public function getGridProduccionDtlDS($ds){
        $ds->SelectCommand = "SELECT * FROM produccion_dtl";
        $ds->UpdateCommand = "UPDATE produccion_dtl SET Valor='@Valor' WHERE
produccion_id=@produccion_id AND ieconomico_id=@ieconomico_id";
        $ds->DeleteCommand = "DELETE FROM produccion_dtl WHERE
produccion_id=@produccion_id AND ieconomico_id=@ieconomico_id";
        $ds->InsertCommand = "INSERT INTO
produccion_dtl(produccion_id,ieconomico_id,Valor) VALUES
(@produccion_id,@ieconomico_id,'@Valor)";
    }

    public function getGridV_ZonaDS($ds){
        $ds->SelectCommand = "SELECT * FROM valle_zona_produccion";
        $ds->UpdateCommand = "UPDATE valle_zona_produccion SET
Nombre='@Nombre', Descripcion='@Descripcion',
Area_Geo='@Area_Geo',Municipio_Id='@Municipio_Id' WHERE Id=@Id";
        $ds->DeleteCommand = "DELETE FROM valle_zona_produccion WHERE
Id=@Id";
        $ds->InsertCommand = "INSERT INTO
valle_zona_produccion(Id,Nombre,Descripcion,Area_Geo,Municipio_Id) VALUES
(NULL,'@Nombre','@Descripcion','@Area_Geo','@Municipio_Id)";
    }

    public function getProdNutrientes($ds){
        $ds->SelectCommand = "SELECT * FROM nutricion_producto";
        $ds->UpdateCommand = "UPDATE nutricion_producto SET Valor='@Valor'
WHERE Nutricion_Id='@Nutricion_Id' AND Producto_Id='@Producto_Id";
        $ds->DeleteCommand = "DELETE FROM nutricion_producto WHERE
Nutricion_Id=@Nutricion_Id AND Producto_Id=@Producto_Id";
    }

```

```

        $ds->InsertCommand = "INSERT INTO
nutricion_producto(Nutricion_Id,Producto_Id,Valor) VALUES
(@Nutricion_Id,@Producto_Id,'@Valor)";
    }

    public function getProdAtributos($ds){
        $ds->SelectCommand = "SELECT * FROM atributos_producto";
        $ds->UpdateCommand = "UPDATE atributos_producto SET Valor='@Valor'
WHERE Atributo_Id='@Atributo_Id' AND Producto_Id='@Producto_Id";
        $ds->DeleteCommand = "DELETE FROM atributos_producto WHERE
Atributo_Id=@Atributo_Id AND Producto_Id=@Producto_Id";
        $ds->InsertCommand = "INSERT INTO
atributos_producto(Atributo_Id,Producto_Id,Valor) VALUES
(@Atributo_Id,@Producto_Id,'@Valor)";
    }

    public function getProdLocalidades($ds){
        $ds->SelectCommand = "SELECT * FROM localidades_producto";
        $ds->UpdateCommand = "UPDATE localidades_producto SET Valor='@Valor'
WHERE Localidad_Id='@Localidad_Id' AND Producto_Id='@Producto_Id";
        $ds->DeleteCommand = "DELETE FROM localidades_producto WHERE
Localidad_Id=@Localidad_Id AND Producto_Id=@Producto_Id";
        $ds->InsertCommand = "INSERT INTO
localidades_producto(Localidad_Id,Producto_Id) VALUES (@Localidad_Id,@Producto_Id)";
    }

    public function getGridProduccionDS($ds){
        $ds->SelectCommand = "SELECT * FROM produccion_hdr";
        $ds->UpdateCommand = "UPDATE produccion_hdr SET
Producto_Id=@Producto_Id,Year='@Year' WHERE Id=@Id";
        $ds->DeleteCommand = "DELETE FROM produccion_hdr WHERE Id=@Id";
        $ds->InsertCommand = "INSERT INTO produccion_hdr(Id,Producto_Id,Year)
VALUES (NULL,@Producto_Id,'@Year)";
    }
}
?>

```

Glosario de términos

- Agropecuaria - Es la actividad humana orientada al cultivo del campo y la crianza de animales. Reúne las palabras “Agricultura y Pecuaria”. Agricultura es el cultivo de la tierra para sembrar alimentos; y la Pecuaria que es sinónimo de ganadería que es la crianza de animales con fines de producción alimenticia. <<http://es.wikipedia.org/wiki/Agropecuaria>>
- FAO por sus siglas en inglés (The Food and Agriculture Organization of the United Nations), La Organización de las Naciones Unidas para la Agricultura y la Alimentación conduce las actividades internacionales encaminadas a erradicar el hambre. Al brindar sus servicios tanto a países desarrollados como a países en desarrollo, la FAO actúa como un foro neutral donde todos los países se reúnen en pie de igualdad para negociar acuerdos y debatir políticas. <<http://www.fao.org/about/en/>>
- La Organización Mundial de la Salud (OMS), es el organismo de la Organización de las Naciones Unidas (ONU) especializado en gestionar políticas de prevención, promoción e intervención en salud a nivel mundial.<<http://es.wikipedia.org/wiki/OMS>>
- La Procuraduría Federal del Consumidor, o la Oficina del Fiscal Federal para el Consumidor (PROFECO) es una organización del gobierno de México, a cargo del Fiscal General, para proteger a los consumidores contra los abusos o fraudes por parte de las empresas que operan en México. <<http://es.wikipedia.org/wiki/PROFECO>>
- Semiología es un término usualmente intercambiable con el de semiótica .La semiosis es cualquier forma de actividad, conducta o proceso que involucre signos. Incluye la creación de un significado. Es un proceso que se desarrolla en la mente del intérprete; se inicia con la percepción del signo y finaliza con la presencia en su mente del objeto del signo. <<http://es.wikipedia.org/wiki/Semiosis>>
- Inocuidad, Es la condición de los alimentos que garantiza que no causaran daño al consumidor cuando se preparen y /o consuman de acuerdo con el uso al que se destinan. <http://es.wikipedia.org/wiki/Seguridad_alimentaria>
- HTML, siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. <[http:// es.wikipedia.org/wiki/HTML](http://es.wikipedia.org/wiki/HTML)>
- XHTML, acrónimo en inglés de eXtensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. <[http:// es.wikipedia.org/wiki/ XHTML](http://es.wikipedia.org/wiki/XHTML)>
- XML [Extensible Markup Language o Lenguaje de marcado extendible]. Norma recomendada por el WC3 para definir nuevos tipos de documentos, permitiendo al usuario

definir sus propias etiquetas de marcado para extender las capacidades del html. Este language facilita las transacciones en las bases de datos
<[http:// es.wikipedia.org/wiki/ XML](http://es.wikipedia.org/wiki/XML)>