BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

FORISSSTEP – Sistema Informático para el Control de Diagnósticos y Licencias Médicas en las Unidades Foráneas del ISSSTEP

TESIS PARA OBTENER EL TÍTULO DE
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

Presenta
David Palacios Sosa

Asesor
Dra. Maya Carrillo Ruiz
Coasesor
Dr. Alberto Téllez Valero

Puebla, Pue. Abril 2015

Índice

CAPITULO 1. 6

Introducción. 6

1.1 Justificación 7

1.2 Objetivo General y Específico. 8

1.3 Hipótesis. 9

CAPITULO 2. 10

Marco Conceptual. 10

2.1 Sistemas de Información Hospitalaria (HIS). 10

2.2 Consulta Externa 13

2.3 Diagnóstico Médico. 14

2.4 Licencias Médicas. 16

2.5 Entorno de trabajo del Instituto de Seguridad y Servicios Sociales de los Trabajadores al Servicio de los Poderes del Estado de Puebla. 17

2.6 Conceptos Computacionales. 20

CAPITULO 3. 35

Análisis y Diseño del sistema. 35

3.1 Requerimientos del Sistema. 35

3.1.1 Casos de Uso. **35**

3.1.2 Diagrama de casos de uso. **36**

3.1.3 Diagrama de secuencias de los casos de uso. **40**

3.2 Modelo Conceptual. 42

3.3 Modelo de Navegación. 43

3.4 Modelo de Interfaces abstractas. 44

CAPITULO 4. 47

Implementación y Pruebas. 47

4.1 Instalación y Configuración de las Bases de Datos. 47

4.2 Instalación del marco de trabajo. 51

4.3 Creación de la aplicación Web en el marco de trabajo. 52

4.4 Acceso a la aplicación a través de roles de usuario RBCA. 53

4.5 Generación automática MVC en el marco de trabajo. 54

4.6 Pruebas y Análisis de Operación. 62

4.6.1 Tiempo de expedición y análisis de DM-2 y DM-3 ..... **62**

4.6.2 Cifras de codificaciones realizadas a formatos DM-3 por mes y región. 63

CAPITULO 5. 64

Conclusión. 64

Bibliografía. 66

Apéndice. 69

APENDICE A. FIGURAS DE DIAGRAMAS DE ACTIVIDADES. 69

APENDICE B. ESTADISTICAS DEL MARCO DE TRABAJO Yii. 71

APENDICE C. FORMATOS ISSSTEP DM-2 Y DM-3. 73

APENDICE D. CODIGO GENERADO PARA EL CONTROLADOR 74

APENDICE E. PROCEDIMIENTO DE CODIFICACIÓN ISSSTEP. 78

Índice de Figuras

Figura 1. Diagrama HIS ISSSTEP 19
Figura 2. Diagrama general de los gestores B.D. ISSSTEP 20
Figura 3. Diagrama de Actividades OOHDM. 26
Figura 4. Diagrama de la estructura estática de Yii. 31
Figura 5. Diagrama del flujo de tareas en Yii. 32
Figura 6. Actores de los Casos de Usos del sistema FORISSSTEP. 35
Figura 7. Diagrama general de los casos de uso del Administrador. 36
Figura 8. Diagrama general de los casos de uso del Medico. 36
Figura 9. Diagrama de Secuencia 1 Ingresar al sistema. 40
Figura 10. Diagrama de Secuencia 2 Crear DM-2 40
Figura 11. Diagrama de Secuencia 3 Modificar DM-2. 41
Figura 12. Diagrama de Secuencia 4 Buscar DM-2 41
Figura 13. Diagrama Entidad Relación 43
Figura 14. Diagrama de Navegación. 44
Figura 15. Boceto de Acceso a usuarios. 45
Figura 16. Boceto de creación de un DM-2 45
Figura 17. Boceto de Búsqueda de un DM-2 46
Figura 18. Página de Requisitos del marco de trabajo Yii. 52
Figura 19. Página de la Aplicación del marco de trabajo Yii. 53
Figura 20. Página de Gii Generador de Modelo. 55
Figura 21. Código del Modelo de la Tabla dm2. 56
Figura 22. Código del Controlador Usuarios. 58
Figura 23. Archivos del Controlador y la Vista creados. 59
Figura 24. Carpetas de las Vistas generadas de la aplicación. 59
Figura 25. Archivos de vistas del dm2 60
Figura 26. Página de Inicio. 60
Figura 27. Página de Creación de un dm2 61
Figura 28. Página de Búsqueda de un dm2 61

Tabla 1. Pasos del Marco de Trabajo YII..... 33
Tabla 2. Ingresar al Sistema..... 37
Tabla 3. Crear DM-2 37
Tabla 4.Modificar DM-2 38
Tabla 5. Buscar DM-2 38
Tabla 6. Visualizar DM-2..... 39
Tabla 7. Tabla de cifras del DM-3 durante 11 meses. 63

CAPITULO 1.

Introducción.

Hoy en día la tecnología es parte de nuestra vida y el ser humano participa en su desarrollo, solo hay que mirar a las grandes instituciones, desde aquellas encargadas de la producción de bienes materiales, como aquellas que proporcionan servicios. Estas instituciones han introducido diversos sistemas de información para el procesamiento y el control de datos, tarea de suma importancia para el buen funcionamiento de las mismas. Este avance tecnológico ha alcanzado una de las áreas de servicios más importantes; la salud. En una unidad médica se maneja la información de manera sistematizada, por lo que se han generado los Sistemas de Información Hospitalaria (*HIS, por sus siglas en inglés; Hospitalary Information System*), estos, son esencialmente sistemas computarizados que centralizan toda la información generada por los distintos servicios médicos a partir de un mismo paciente (Guerrero Martínez, 2011). En el caso de México, los hospitales de mayor capacidad son aquellos que adquieren estos sistemas, como son el Instituto Mexicano del Seguro Social (IMSS) y el Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado (ISSSTE), así como otras tantas instituciones que se han dado a la tarea de estar a la vanguardia en cuanto a la mejora de servicios, como es el caso del Instituto de Seguridad Y Servicios Sociales de los Trabajadores al Servicio de los Poderes del Estado de Puebla (ISSSTEP), lugar donde se llevó a cabo el presente proyecto, ya que se centra en hacer posible dentro del HIS de esta institución, la adición de un módulo extra, que permita la expedición automatizada de licencias médicas, así como, la documentación del diagnóstico médico, de forma especializada, fácil y eficiente, en particular en las unidades médicas que se ubican en lugares con escasa conectividad para la transición de datos electrónicos. Cabe mencionar que el ISSSTEP cuenta con una población aproximada de 158,000 derechohabientes tanto en la capital como en las unidades médicas foráneas donde desafortunadamente no se cuenta con la infraestructura tecnológica necesaria, quedando rezagado y con la imposibilidad de contar con un HIS propio o ser parte del HIS que utiliza la unidad central de la capital. Esto ha generado una serie de problemáticas relacionadas con el manejo de la información, ya que existe fuga y pérdida de la misma, así como, falsificación de

documentación hospitalaria, siendo la acción más común y dañina para estas instituciones.

Debido a esto existe la necesidad de dar alternativas de solución que lleven a resolver estas problemáticas, por lo que el aporte de esta investigación se centra en adicionar al HIS del ISSSTEP un módulo para las unidades foráneas, que les permita tener en sus manos una aplicación web capaz de realizar la expedición de licencias médicas y el registro de diagnósticos médicos de manera computarizada para poder trasmitir la información a la unidad sede en Puebla para su debido resguardo y posterior análisis, también permite al personal especializado tener la información de manera oportuna para tomar decisiones que generen el diagnostico pertinente para cada uno de los derechohabientes de esta institución.

Sin embargo la adición de un módulo al HIS del ISSSTEP para sus unidades foráneas, es una tarea que requiere reflexionar lo siguiente: ¿Cómo aligerar la información requerida en un HIS para su trasmisión en unidades con internet de poca velocidad?

1.1 Justificación

Día con día se producen cantidades significativas de datos que generalmente se encuentran dispersos y que en determinado momento se procesan, analizan e interpretan para convertirse en información y poder ser de utilidad para las personas. A raíz del surgimiento de las computadoras, comenzaron a crearse sistemas sencillos de análisis y procesamiento de información, con la finalidad tanto de administrar como de manejar recursos financieros.

En lo que se refiere a las instituciones hospitalarias fue hasta la década de los 70's cuando se perfilaron los primeros sistemas de información "médica" los cuales posteriormente, habrían de dar lugar a los Sistemas de Información Hospitalaria, de los que se sabe, son un elemento indispensable para el adecuado funcionamiento de estas instituciones en la actualidad. Estos sistemas han generado un impacto importante en las instituciones de salud, ya que han contribuido al mejoramiento en la calidad de atención al paciente, así como, de los servicios brindados y su aportación en la aplicación de información obtenida en las áreas de investigación, clínica, docencia, administración y por último ha contribuido a reducir costos y elevar la productividad.

En lo que se refiere al surgimiento de los Sistemas de Información Hospitalaria (HIS) en México, sé sabe que existen, pero no ha habido la suficiente difusión e intercambio de experiencias (Hernández López, 2003-Martinez, M.,2007). Para el caso del estado de Puebla, en específico el lugar donde se desarrolló esta investigación, el hospital ISSSTEP, que se encuentra ubicado en la capital poblana. Esta es una institución médica de vanguardia, donde el desarrollo de un HIS es una prioridad, pero como se mencionó anteriormente sus unidades médicas foráneas no cuentan con la infraestructura tecnológica que les permita una transmisión eficaz de datos a través de una intranet o red local, por lo que el propósito de esta investigación se basa en la creación e implementación de una aplicación web para el registro de licencias médicas y la captura de diagnósticos médicos. Esta aplicación es ligera y se encuentra interconectada con los servidores centrales de la capital de manera que toda la información quede concentrada y unificada. También permite el cotejamiento y posterior manejo de información, que da como resultado realizar las tareas de forma automatizada, eficiente, rápida y segura.

Por último es importante mencionar que este tipo de aplicaciones permiten la rápida y confiable transmisión de información pertinente al hospital y en tiempo real, por lo que en su momento, el desarrollo y perfeccionamiento de estas aplicaciones, permitirá en un futuro, la detección oportuna de casos médicos relevantes y el cotejamiento de documentación oficial hospitalaria, siendo estos, beneficios de suma importancia para el sistema de salud mexicano.

1.2 Objetivo General y Específico.

GENERAL.

Automatizar la administración y análisis de información referente a diagnósticos y licencias médicas en las Unidades Médicas Foráneas del ISSSTEP con internet de baja velocidad.

PARTICULARES.

- Desarrollar una plataforma Web que permita el registro de información en las Unidades Foráneas por medio de una red de Internet, misma que deberá interactuar con los sistemas existentes para consulta del padrón de médicos y derechohabientes.

- Implementar un acceso por roles, restringiendo la consulta y registro de información según la responsabilidad de cada usuario; además de documentar los procesos y manuales necesarios para la capacitación y utilización correcta del sistema.
- Desplegar reportes que presenten la información registrada.

1.3 Hipótesis.

Una aplicación Web permitirá la correcta administración de la información generada en la expedición de licencias médicas y el registro de diagnósticos médicos para las unidades foráneas del ISSSTEP.

CAPITULO 2.

Marco Conceptual.

En este capítulo se describirán los conceptos básicos inmersos en este proyecto con la finalidad de exponer de manera general los elementos utilizados, así como, las características de los mismos.

2.1 Sistemas de Información Hospitalaria (HIS).

Hoy en día, la información se genera en grandes cantidades, lo cual contribuye a un proceso de desarrollo global más rápido y mejor informado, esto ha permitido la incorporación de más y más herramientas tecnológicas a nuestras vidas, una de ellas es la llamada computadora, una herramienta que se puede encontrar fácilmente en la mayoría de los espacios públicos o privados y que se ha convertido en la clave para el desarrollo de la sociedad. La importancia de la computadora, radica principalmente, en el fácil acceso a la información, esto ha ido trasformando muchos aspectos de la vida social haciéndolos más prácticos. Dentro de los avances tecnológicos más importantes que ha generado esta herramienta, es proporcionar a los hospitales un sistema de información, el cual, ha sido un medio para recabar, almacenar, comunicar y trasportar gran cantidad de información por medio de un equipo de cómputo.

Se puede definir al sistema de información hospitalaria *(Hospital Information System, HIS)* como un sistema que centraliza toda la información generada por los distintos servicios del hospital acerca de un paciente. Por tanto, la característica básica que define a un HIS de otro sistema de información, es su capacidad integradora. En los sistemas tradicionales, existe una orientación en función de departamentos, por lo que los resultados generados por distintos servicios dentro del hospital (radiología, laboratorio clínica, etc.), son generalmente plasmados en algún tipo de soporte físico (papel, película radiográfica, etc.) y transmitidos manualmente al especialista que los precisa para el diagnóstico. Un HIS a diferencia de los sistemas tradicionales, está orientado al paciente, por lo que todos los sistemas que generan información están interconectados (Guerrero Martínez, 2011). De esta forma se logra hacer eficiente el acceso a la información necesaria, generando un resultado, fácil y preciso en la clasificación de aspectos

médicos, en el manejo del archivo clínico, en la recuperación y tratamiento de los datos del paciente etc.. Todo esto, como una solución integral del uso de la tecnología en un centro de salud como método para el manejo de la información, es decir; un Sistema de Información Hospitalario (Aghazadeh, Aliyev, & Ebrahimnezhad, 2012).

OBJETIVOS DEL SISTEMA DE INFORMACIÓN DE HOSPITALARIA (HIS).

El objetivo de un Sistema de Información Hospitalaria (HIS), es apoyar las actividades del hospital en los niveles de práctica, táctica y estrategia. En otras palabras, la meta del sistema de información hospitalaria (HIS), es el uso de computadoras y equipos de comunicación para recolectar, almacenar, procesar, dar lectura y comunicar entre los pacientes y los administrativos, datos sobre todas las actividades hospitalarias y de esta forma cumplir con las necesidades de todos los consumidores. En general, las metas principales de un Sistema de información hospitalaria (HIS) se pueden resumir de la siguiente manera:

- Mejorar la eficiencia del personal.
- Eliminar la duplicación y procedimientos innecesarios.
- La incorporación de una computadora como herramienta de trabajo.
- La realización de estadísticas, así como técnicas de minería de datos más precisas y rápidas.
- Mejorar la calidad de la atención de salud de los pacientes.
- Crear métodos y sistemas modernos de trabajo y normalizar los existentes en los hospitales.
- Modernizar los sistemas de comunicación de datos.
- Construir un buen sistema de transmisión de datos entre todos los hospitales dentro del país, alcanzando una base de datos distribuida y con ello tener una poderosa red de salud mundial
  (Aghazadeh, Aliyev, & Ebrahimnezhad, 2012).

Por otra parte, todo sistema de información aporta muchas ventajas, ya que están diseñados para el correcto flujo de datos, asegurándonos que cada paso es pensado y realizado de manera que no exista perdida alguna de información. Este es un reto para los HIS, pues deben ser capaces de almacenar una gran cantidad de información valiosa ya que su oportuna y completa llegada a las personas indicadas refleja el éxito o fracaso de la atención a la salud de un paciente. Los objetivos son claros y su visión está ligada

no solo al aprovechamiento máximo de la información, sino también a ser capaz de procesarla, algo que ningún ser humano sería capaz de hacer con tal rapidez, agilidad y seguridad, como lo puede hacer una herramienta de cómputo que cuente con este sistema. Esto nos lleva a mencionar, cada una de las propiedades y características que tiene un sistema de información hospitalaria (HIS), y que se pueden enlistar de la siguiente manera:

- Actúa basado en la norma.
- No se ejerce ninguna actividad escrita, pero se obtiene el mismo resultado al final.
- Actúa basado en eventos médicos y es independiente del ciclo de actividades del paciente.
- Usando el sistema, lo que se realizaba de forma manual no cambia mucho, en cuanto a contenido y calidad.
- Trabaja a la par y sin perjudicar a los sistemas antiguos proponiendo mejoras a futuro.
- Ofrece la mejor solución para la coordinación entre las diferentes líneas de trabajo y las diferentes unidades dentro del hospital.
- Aumenta la calidad en la toma de decisiones para puestos gerenciales.
- Incluye extensas bases de datos sobre medicina basadas en el SNOMED y CIE-10.
- Solo en un 2% de los casos la entrada de datos se ingresa desde teclado y en el restante 98% de casos solo bastara con hacer un clic a través de un ratón.
- El funcionamiento es muy simple, fácil de usar y totalmente visual.
- Está equipado con sistemas de videoconferencia para los profesionales de la salud.
- También está equipado con inteligencia artificial basada en el conocimiento de los diagnósticos y tratamientos.
- Proporciona acceso a la información a través de Internet con los teléfonos móviles.
- Presenta y recupera información médica con acceso fácil de un conjunto complejo y estratégico de datos.
- Está equipado con los estándares de código abierto para la implementación de lenguaje local.

- Es posible el uso de tecnología multimedia, audio y video grabación para la diversa información que puede contener (Aghazadeh, Aliyev, & Ebrahimnezhad, 2012).

También cabe resaltar que los HIS, están primordialmente orientados a la facilitación de diversas acciones dentro de una institución hospitalaria incluyendo tanto técnicas, como recursos específicos pertenecientes al ámbito computacional que dan como resultado, la solución de problemas y la generación de diversos beneficios que pueden mejorar la calidad y eficacia de la atención a los pacientes dentro de las instituciones hospitalarias donde se implementan.

Por ultimo debemos decir que dependiendo de las necesidades más importantes de cada institución hospitalaria, serán implementados módulos para conformar un HIS y estos pueden ser desde los más básicos como son los de tipo administrativo, pasando por los de control de personal hospitalario, o los de tipo personalizado y completo, es decir en los que se implementan módulos en todas las áreas de la institución, siendo este el objetivo principal de estos sistemas y lo que permite automatizar todas las acciones que se dirigen a mejorar la atención médico-paciente.

2.2 Consulta Externa

La consulta externa es un servicio hospitalario, que tiene como finalidad cerciorarse del estado de salud de un paciente al ingresar a una institución hospitalaria. En una consulta externa se brinda atención médica a los pacientes con padecimientos en general, brindando una evaluación y en caso de ser necesario, derivar de forma posterior a los servicios de especialidad cuando se requiera (MÉXICO, 2014).

Por otra parte, la atención médica en consulta externa puede ser de diversa índole, pero principalmente consiste en el interrogatorio y examen médico que lleva al médico a un diagnóstico y a la prescripción de un tratamiento, por lo que la visita médica se concluye con la expedición de ese diagnóstico, el cual valora la sintomatología del paciente y cuya descripción, puede ser de gran ayuda a la comunidad médica ya sea para visitas, o tratamientos posteriores del paciente. Estos diagnósticos son resguardados en el archivo clínico y quedan plasmados en formatos con ciertos estándares, según la atención médica de cada país y de acuerdo a la institución donde se expiden. Todo esto, con la finalidad

de propiciar una adecuada valoración médica de la consulta externa, que permita la pronta salud del paciente y a su vez, lleven a cabo acciones para la mejora en la calidad del servicio de las instituciones hospitalarias.

2.3 Diagnóstico Médico.

Los diagnósticos médicos se integran en un expediente clínico, el cual, es un documento legal y confidencial, en el que se registran los datos necesarios para poder realizar una correcta historia médica de cada paciente. La elaboración de un diagnóstico médico es una de las acciones más cotidianas dentro de un hospital y es el resultado del análisis realizado por el médico sobre la salud de un paciente dentro de la consulta. Este diagnóstico, es una de las tareas más importantes a las que se enfrenta cualquier médico, ya que implica la tarea de definir si el paciente se encuentra sano o no, y en función de ello proponer las intervenciones correspondientes. (García García, 2010). Por lo tanto, el diagnóstico médico es el proceso donde se recaba información que posteriormente se analizará y servirá para generar la historia clínica del paciente. Es por ello que salvaguardar esta información, se convierte en una tarea de suma importancia dentro de una unidad médica, tanto en el presente como en el futuro, ya que en el presente ayuda a la mejora del paciente a través de la comunicación de datos en las diversas áreas del hospital, con la finalidad de que este sea atendido de manera correcta y recupere su estado de salud rápidamente, y en cuanto al futuro, estos datos sirven para el análisis estadístico, así como, para la detección de nuevas enfermedades y su grado de propagación, como es el caso de las epidemias, donde su oportuna detección podría permitir un plan de control y erradicación.

Por otra parte, el diagnostico medico contiene información necesaria de cada uno de los pacientes. Esta información puede agruparse de acuerdo a ciertos factores, como son:

- Biológicos:
  - Edad.
  - Sexo.
  - Peso.
  - Estatura.

- Etc.

- Psicológicos:
  - Rasgos de la personalidad.
  - Importancia atribuida a las manifestaciones de la enfermedad.
  - Oportunidad de búsqueda de atención.
- Sociales:
  - Datos personales como: Nombre, Estado Civil. Etc…

Recabar esta información con cada uno de los pacientes de una institución hospitalaria, es una tarea ardua y compleja, aun contando con el recurso humano capacitado y apto para realizar estos registros. Es por eso, que actualmente el personal encargado de estas tareas, ha encontrado una forma de facilitar esta actividad por medio del uso de la tecnología, utilizando la computadora como una herramienta que facilita el diagnóstico médico a través de la rápida captura de información, una acción indispensable para adaptarse a los nuevos ritmos de vida del siglo XXI, donde es necesario poder emitir un diagnóstico de manera eficaz, y en corto tiempo. Es por esto, que la correcta documentación de un diagnóstico médico en un HIS es una de las acciones que facilitan en mayor medida la comunicación de datos, tanto para mejorar la calidad de atención del paciente como para la solución de futuros casos similares y la detección oportuna de padecimientos del tipo epidemiológico, como se ha mencionado anteriormente.

Por otro lado, es importante mencionar que cada institución hospitalaria tiene su propio formato, así como, su propia metodología para la captura de diagnósticos médicos. Sin embargo, desde hace algunos años se ha buscado que exista una unificación a nivel nacional e internacional de estos formatos, con la finalidad de que la información se pueda leer, extraer y analizar de forma simultánea y ser utilizada para diferentes propósitos. Debido a esto, es necesario llevar a cabo una codificación del diagnóstico médico bajo las normas que establece la Organización Mundial de salud (OMS), para que cada hospital pueda cumplir con esta serie de procedimientos y permitir el correcto funcionamiento del mismo. A pesar de esto se sabe que en México la mayoría de los hospitales a un optan por realizar la transcripción y codificación de diagnósticos médicos de manera manuscrita y solo en ocasiones particulares con ayuda de máquinas de escribir, por lo que desafortunadamente son muy pocos los hospitales que optan por el

uso de la computadora como herramienta de captura y un porcentaje aún más reducido son los que optan por centrar la información en un sistema de tipo HIS.

2.4 Licencias Médicas.

Una licencia médica, incapacidad médica o incapacidad laboral temporal, es la autorización otorgada por el médico, a un trabajador, con el fin de que se abstenga del ejercicio de sus actividades laborales por adolecer de una enfermedad que, además de asistencia médica, amerita reposo para su restablecimiento. También se define como la situación en que se encuentra el trabajador, mientras reciba asistencia sanitaria de seguridad social y esté impedido para el trabajo (Díaz Vélez & Peralta Lui, 2013). Por lo tanto, los datos más importantes recabados en una licencia médica son:

Datos Personales:

- Nombre
- Edad
- Sexo
- Edo. Civil.
- Etc..

Datos Hospitalarios:

- Unidad Medica
- Datos generales del medico
- Etc…

Datos de la empresa donde se labora:

- Nombre
- Dirección
- Etc…

Breve Descripción:

- Diagnóstico e indicaciones medicas

Fechas de solicitud o expedición y termino de la incapacidad

- Los días que tendrá validez la licencia médica desde su día de expedición hasta el día de expiración.

Es importante recordar que cada institución hospitalaria tiene su propia norma para expedir licencias médicas, así como, criterios de información necesaria para posteriores procesos de administración y concentrado de esta.

En lo que se refiere a la parte legal, obtener una licencia médica conlleva recibir una suma de dinero que corresponde al pago del período en que un trabajador ha estado bajo esa licencia, y cuya finalidad es reemplazar la remuneración o renta que deja de percibir mientras está enfermo (La Araucana, 2010). Por lo tanto, cada institución hospitalaria cuenta con la facultad y es responsable de otorgar estas licencias, así que, es indispensable que puedan llevar un adecuado control en la expedición de las mismas y verificar su autenticidad, de forma que con ello, se vean beneficiados tanto la institución como los derechohabientes que soliciten de este servicio.

A continuación, se describirá el entorno de trabajo, la metodología de desarrollo y la tecnología aplicada a la solución del problema, con la finalidad de exponer de manera detallada las características de los mismos.

2.5 Entorno de trabajo del Instituto de Seguridad y Servicios Sociales de los Trabajadores al Servicio de los Poderes del Estado de Puebla.

El Instituto de Seguridad y Servicios Sociales de los Trabajadores al Servicio de los Poderes del Estado de Puebla (ISSSTEP), es un Organismo Público Descentralizado de la Administración Pública Estatal, cuya misión fundamental es brindar servicios de salud a los trabajadores del Gobierno del Estado, incluidos sus familiares.

Para cumplir con tal propósito, el ISSSTEP cuenta con 37 Unidades Médicas Familiares repartidas por todo el Estado de Puebla, además de un Hospital de Especialidades ubicado en la metrópoli. A la fecha el ISSSTEP tiene en su plantilla cerca de 400 médicos, para atender a sus más de 158,000 derechohabientes.

El ISSSTEP cuenta con un HIS, que está integrado por diferentes módulos especializados en cada área de la institución, pero no sucede lo mismo con sus unidades médicas foráneas, donde la infraestructura tecnológica se encuentra en crecimiento paulatino, por lo que el propósito del instituto es incorporarlas a dicha infraestructura y con ello lograr un servicio de calidad para todos sus derechohabientes. Las Unidades Médicas Foráneas

beneficiadas por medio de este proyecto se presentan a continuación según su división demográfica:

REGION 1

CHOLULA
SAN MARTIN TEXMELUCAN
TEPEACA
TEPEXI DE RODRIGUEZ

REGION 2

CIUDAD SERDAN
GUADALUPE VICTORIA
LIBRES

REGION 3

IZUCAR DE MATAMOROS
ATLIXCO
CHIAUTLA DE TAPIA
ACATLAN DE OSORIO

REGION 4

TEHUACÀN
TECAMACHALCO
IXCAQUIXTLA

REGION 5

HUAUCHINANGO
MECAPALAPA
VILLA LAZARO CARDENAS
PAHUATLAN
XICOTEPEC
TLAXCO
VILLA AVILA CAMACHO

REGION 6

TEZIUTLAN
AYOTOXCO
HUEHUETLA
SAN JOSE ACATENO
TLATLAUQUITEPEC
ZACAPOAXTLA
CUETZALAN
ZARAGOZA

REGION 7

ZACATLAN
BIENVENIDO DE GALEANA
CHIGNAHUAPAN
OLINTLA
TETELA DE OCAMPO
AHUACATLAN

Es preciso decir que durante el ejercicio 2013, el personal que integra el Departamento de Tecnologías de la Información realizó 1,347 actividades encaminadas a mantener y mejorar la infraestructura en tecnologías de la información con que cuenta el ISSSTEP, facilitando la administración de información como apoyo para brindar un servicio oportuno y de calidad a los derechohabientes. (Palma Valdés, 2013)

A continuación se presenta un diagrama general y jerárquico del HIS ISSSTEP, en el que se puede visualizar en qué lugar se integra el modulo implementado. Es importante mencionar que todo desarrollo dentro del HIS tendrá que ser monitoreado por la unidad

de la capital y tener acceso a los datos del instituto, si es requerido los módulos deberán tener comunicación entre sí.

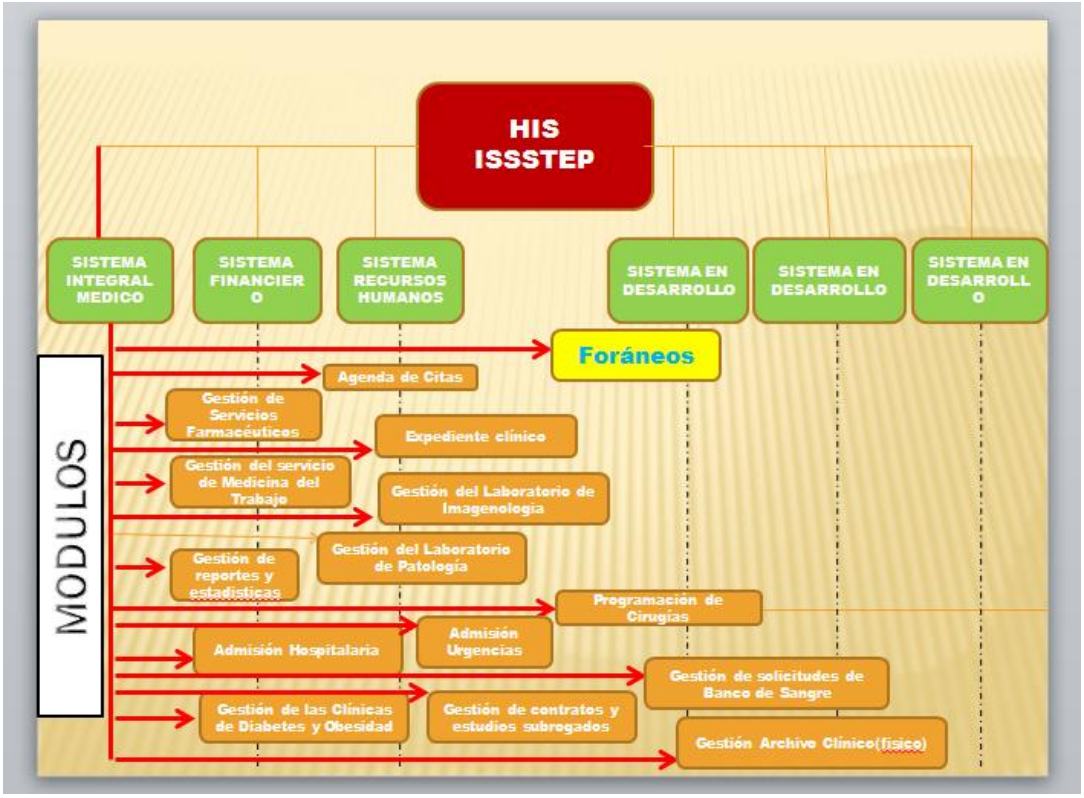


Figura 1. Diagrama HIS ISSSTEP

Por términos de privacidad solo se mostrara de manera gráfica los gestores de bases de datos, que contienen la información requerida para el proyecto, esta se divide de la siguiente manera; datos referentes a médicos (ORACLE 11G), a derechohabientes (SQLSERVER2000) y los que intervienen para el módulo de Foráneos (MySQL).

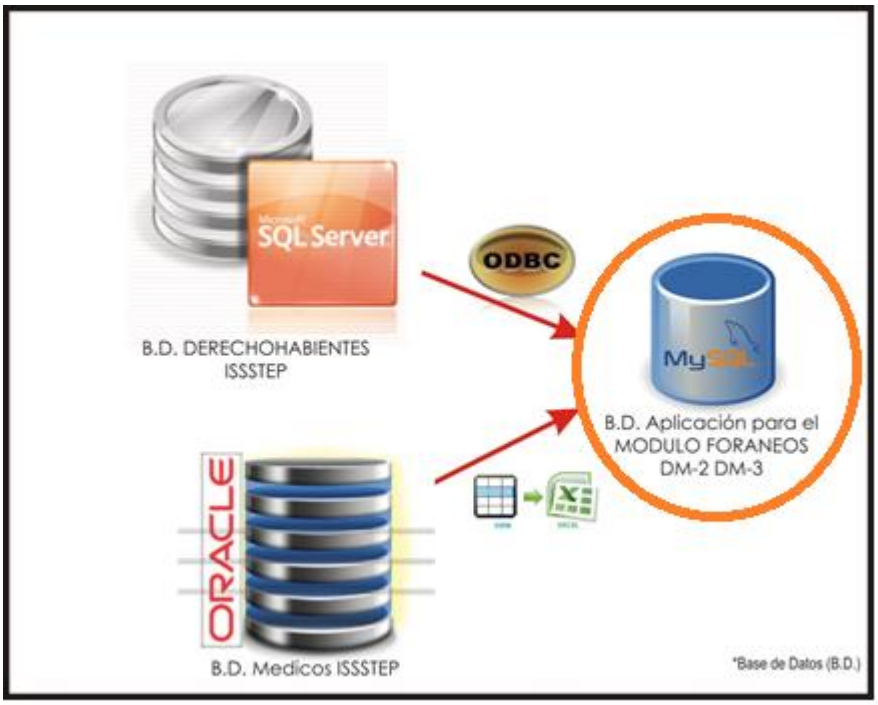


Figura 2. Diagrama general de los gestores B.D. ISSSTEP

En las Unidades Foráneas del ISSSTEP se realizan diferentes trámites a derechohabientes a través de formatos impresos y elaborados manualmente por médicos, como son: control de citas, recetas médicas, incapacidades médicas y diagnósticos médicos, este último como producto de una consulta externa. Los formatos de incapacidades y diagnósticos médicos en el ISSSTEP mantienen un estándar y cumplen con todas las normas de salud del Estado de Puebla para su realización y expedición. Los pasos en cada trámite deberán ser rápidos y su validación será de suma importancia para el instituto.

Estos formatos se expiden por médicos y administrativos identificados con una clave, para el caso de una incapacidad médica la clave es: DM-2, y para el diagnóstico médico de consulta externa es: DM-3 (Véase *Apéndice C*).

2.6 Conceptos Computacionales.

Existen diferentes herramientas para presentar o enriquecer el resultado de un proyecto en la Web, pero sin duda son las aplicaciones Web las que hoy en día han ganado más

popularidad por sus diversas características como son; rapidez de implementación y facilidad de uso.

APLICACIONES WEB.

Las aplicaciones Web se han desarrollado al mismo tiempo que la Web, y nos ayudan a la simplificación y visualización de información de manera rápida y ordenada. Por lo que su desarrollo e implementación se utilizan diferentes estándares, que las mejoran continuamente.

Las aplicaciones web permiten la generación automática de contenido, la creación de páginas personalizadas o el desarrollo del comercio electrónico. Una aplicación web permite interactuar con los sistemas informáticos de gestión de una empresa, como puede ser gestión de clientes, contabilidad o inventario, etc., todo a través de una página web (Pérez García, 2007). Al estar en un ambiente web sabemos que estamos frente a riesgos potenciales de acceso y mal uso de los servicios para la manipulación de la información disponible. Obviamente, existen diferentes criterios para la clasificación de los sistemas, esto depende directamente del negocio de la empresa, ya que unos deberán poseer e implementar muchas técnicas de seguridad más que otros, pero en general todas las aplicaciones Web deben estar protegidas y aseguradas ante los principales ataques.

Por otro lado la mayoría de las aplicaciones web se basan en la arquitectura cliente-servidor, donde un usuario por medio del navegador introduce información, mientras que el servidor almacena y recupera información. El correo electrónico es un ejemplo de esto, con empresas como Gmail ó Outlook que ofrecen a sus clientes correos electrónicos basados en la web. Por lo tanto, estas aplicaciones pueden proporcionar funcionalidad operativa importante dentro de las instituciones y obtener el beneficio de trabajar con una red comunicada por los diferentes miembros del lugar de trabajo.

Una aplicación puede llegar a automatizar tareas en cada departamento, actuando por ejemplo, como un procesador de textos y tener así un banco de almacenamiento de información, lo que permite "descargar" el documento en el disco duro personal para acceso a cualquier miembro de la organización, esto entre muchas otras tareas importantes (Naciones, 2012).

Es importante señalar que para desarrollar aplicaciones Web, es necesario seguir siempre una metodología, ya que esto permite tener tareas más claras al momento de

empezar, desarrollar y desplegar un proyecto, esto con la finalidad de obtener los mejores resultados y satisfacer los requerimientos del cliente.

METODOLOGIAS PARA EL DESARROLLO DE APLICACIONES WEB.

En la actualidad existen diferentes metodologías que apoyan y facilitan el desarrollo de las aplicaciones web, sin embargo, continúan desarrollándose propuestas con nuevos enfoques para mejorar la calidad. Por lo que la Ingeniería Web, es un área de oportunidad para la investigación y desarrollo de herramientas que permitan crear aplicaciones de alta calidad. (Pacheco, 2013).

A continuación se presentan algunas metodologías importantes y sus características representativas.

METODO DE INGENIERIA WEB BASADO EN UML

El método de ingeniería web basado en UML (UWE - UML BASED WEB ENGINEERING) soporta el desarrollo de aplicaciones Web con un especial interés en la sistematización, personalización y generación semi-automática de código. Es una metodología iterativa e incremental orientada a objetos, basada en el Lenguaje de Modelado Unificado (UML) y en el Proceso Unificado de Desarrollo de Software (PUDS). UWE se centra en actividades de modelado como, el análisis de requerimientos y el diseño conceptual de navegación y de presentación complementado con modelado de tareas y visualización de escenarios Web. Existe una aplicación encargada de la generación semi-automática de código Web en base a los modelos de diseño generados llamada ArgoUML (Hennicker & Koch, 2000).

Características Importantes:

- En la etapa de requisitos separa las fases de captura, definición y validación.
- Se realiza una clasificación y tratamiento especial según el carácter de cada requisito.
- Uso de una notación estándar, para todos los modelos (UML : Lenguaje de modelado unificado).

- Definición de los pasos para la construcción de los diferentes modelos.
- Utilización de restricciones escritas (OCL: Lenguaje de restricciones de objetos) para aumentar la exactitud de los modelos.
- Incorpora una etapa de navegación, que ayudara a construir caminos más definidos para el perfil de cada usuario *(Véase Apéndice A).*

MÉTODO DE DISEÑO PARA SITIOS WEB

El Método de diseño para sitios Web (WSDM- The Website Design Method) se centra completamente en los usuarios del sitio web. Se definen las "clases de usuarios" y según las futuras visitas, y la forma en que estos usuarios recorrerán el sitio, se establecen los parámetros de diseño. Por lo que esta metodología esta orientada principalmente al desarrollo de sitios web de información y no de sitios interactivos o aplicaciones. (De Troyer, 1998).

Características Importantes:

- Define el sistema en base a los grupos de usuario o clases de usuarios.
- El proceso de definición de requisitos, tiene por objetivo detectar los perfiles de usuario a través de pocas tareas.
- La clasificación de usuarios en un modelo de usuarios mediante el estudio del entorno *(Véase Apéndice A.)*.
- Descripción de los grupos de usuario.

METODOLOGÍA DE DISEÑO ORIENTADA A OBJETOS HIPERMEDIA EN ESCENARIOS

La metodología de diseño orientada a objetos hipermedia en escenarios (SOHDM-Scenario-based Object-Oriented Hypermedia Design Methodology) se basa en un proceso a partir de escenarios, donde se modela conceptualmente el sistema, el cual se representa mediante un diagrama de clases. Esto permite que el proceso de SOHDM

reagrupe los diagramas para así conseguir un modelo de clases de navegación del sistema.

Sin embargo, en la actualidad esta metodología ha caído en desuso, principalmente por el uso que hace de diagramas de flujo de datos (DFD). (Heeseok, Choongseok, & Cheonsoo, 1999).

Características Importantes:

- Ofrece un modelo de escenarios propio para representar los requisitos, denominado escenario de cuadro de actividades (SAC-scenario activity chart).
- Se utiliza un diagrama de contexto propuesto en los DFD.
- Está compuesta por seis fases secuenciales y es posible volver a cualquier fase para refinarla y adaptarla si se requiere *(Véase Apéndice A)*.

Es importante mencionar que estas metodologías se revisaron para este proyecto, aunque, no se utilizaron, ya que se basan solamente en aspectos parciales de una aplicación web y en laboriosas tareas de modelado. Por lo tanto, a continuación se describe la metodología utilizada para el diseño, desarrollo e implementación de la aplicación web de este proyecto.

MÉTODO DE DISEÑO EN EL DESARROLLO DE HIPERMEDIA ORIENTADO A OBJETOS (OOHDM)

Este método se basa en modelos que facilitan el desarrollo e implementación de aplicaciones Web basadas en hipermedia. Estos modelos permiten describir información (conceptual), aspectos de interfaz y de navegación de la aplicación, así como, él mapeo de modelos en aplicaciones Web ejecutables, para distintos tipos de entornos.

Descripción de las etapas.

- Modelo conceptual, en esta etapa se construye un modelo conceptual dentro del dominio de la aplicación utilizando los principios del modelado orientado a objetos.

Sin embargo, los atributos pueden ser de múltiples tipos para representar perspectivas diferentes de las mismas entidades del mundo real. Esta etapa no considera a los tipos de usuarios y tareas, sino más bien al dominio semántico de la aplicación. En OOHDM, el esquema conceptual está construido por clases, relaciones y subsistemas. Las clases son descritas como en los modelos orientados a objetos tradicionales.

- Modelo de navegación, este es construido como una *vista* sobre un diseño conceptual, admitiendo la construcción de diferentes modelos de acuerdo con los diferentes perfiles de usuarios. Cada modelo de navegación provee una vista subjetiva del diseño conceptual. El diseño de navegación es expresado en dos modelos: el modelo de clases de navegaciones (nodos), el cual describe qué información puede ser accedida, y el modelo de contexto de navegación, que describe cómo puede ser ingresada dicha información. La semántica de los nodos y los enlaces son las tradicionales aplicaciones de tipo hipermedia, y las estructuras de acceso, como son los índices o recorridos guiados, representan los posibles caminos de acceso a los nodos.
- Modelo de interfaz abstracta, este permite definir el aspecto que tendrán las interfaces, significa definir la forma en la cual aparecerán los objetos de navegaciones, la manera en la cual los objetos de interfaz activarán la navegación y el resto de las funcionalidades de la aplicación. Una clara separación entre diseño de navegación y diseño de interfaz abstracta permite construir diferentes interfaces para el mismo modelo de navegación, dejando un alto grado de independencia de la tecnología de interfaz de usuario.
- Implementación, una vez terminadas las etapas anteriores, el desarrollador posee un completo conocimiento del dominio del problema. Así entonces, una vez que sea identificado la información que será mostrada, como estará organizada y qué funciones permitirá ejecutar la aplicación, comenzara la implementación, aquí el desarrollador deberá elegir donde almacenará los objetos y con qué lenguaje o herramienta desarrollará las interfaces. Es necesario aclarar que generalmente el desarrollador se encarga del lado técnico de la interfaz, la parte gráfica y la apariencia final de la interfaz será tarea del diseñador gráfico.

Las etapas antes descritas se pueden enumerar y visualizar de forma clara a través del diagrama de actividades que se presenta a continuación:

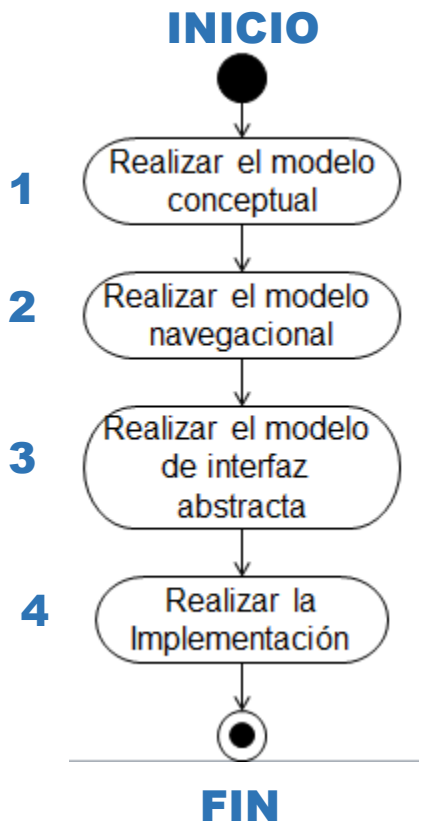


Figura 3. Diagrama de Actividades OOHDM.

PATRONES DE DISEÑO.

Los patrones de diseño son soluciones estudiadas que se han aplicado con éxito en diversos sistemas de producción y que han respondido con buenos resultados a las exigencias de los usuarios y desarrolladores, éstos permiten la reutilización de diseños abstractos que no incluyan detalles de implementación. Existen 23 patrones de diseño diferentes, clasificados en tres categorías en el libro "Design Patterns: Elements of Reusable Object-Oriented Software" (Gamma, Helm, Johnson, & Vlissides, 1994), todas estas atienden a la solución de un problema en términos de objetos e interfaces. Pero los patrones de diseño que se utilizan con mayor frecuencia son: Abstract Factory, Decorator, Observer y Modelo-Vista-Controlador(MVC) este último tiene la capacidad de actuar tanto como patrón de diseño o de arquitectura dependiendo de su aplicación. Por lo tanto, MVC es el patrón de diseño que por su separación de capas, origina una estructura más

robusta en el desarrollo de aplicaciones web, permitiendo una mejor organización del trabajo y mayor especialización de los desarrolladores y diseñadores.

Patrón de Diseño Modelo-Vista-Controlador .

El patrón de diseño modelo-vista-controlador (MVC), obtuvo éxito desde que fue presentado por primera vez en 1979 por Trygve Reenskaug. Este patrón permite generar una estructura lógica para las aplicaciones, su principal objetivo es separar la lógica del negocio, de la lógica de presentación o interfaz, es usado principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas donde se requiere una mejor separación de conceptos para que el diseño este estructurado de una mejor manera. Este patrón se separa en tres principales componentes:

- **Modelo.**

  El modelo contempla los datos de la aplicación y sus reglas de negocio. Su propósito es la representación de la información, es decir, modelar el comportamiento y los datos que maneja la aplicación.

- **Vista**

  La vista es la responsable de transformar el modelo para que pueda ser visualizado por el usuario, ya sea en un archivo o una página web desplegable que permita visualizar la información. El componente vista, por lo general, responde a los eventos generados por el componente controlador, el cual, a su vez, es provocado por una acción del usuario. Los eventos generados por el controlador pueden ser activados cuando el modelo ha sido actualizado y las vistas tienen la necesidad de actualizarse para reflejar los cambios en el modelo. Por tanto, el propósito de la vista es convertir los datos para que le sean significativos al usuario y pueda interpretarlos fácilmente.

- **Controlador**

  Es la parte lógica, responsable del procesamiento y comportamiento de acuerdo a las peticiones del usuario, permitiendo la construcción de un modelo apropiado, para su correcta visualización.

El funcionamiento de los tres componentes en conjunto es el siguiente:

Un usuario hace una solicitud al controlador con la información sobre lo que desea realizar. Entonces el controlador decide a quien debe delegar la tarea y es aquí, donde el Modelo empieza su trabajo, este se encarga de realizar operaciones sobre la información

que maneja para cumplir con lo que solicita el Controlador. Una vez que termina su labor, regresa al Controlador la información resultante de sus operaciones, el cual a su vez redirige a la Vista. La Vista se encarga de transformar los datos en información visualmente entendible para el usuario. Finalmente, la representación gráfica es transmitida de regreso al Controlador y este se encarga de transmitirla al usuario. Este, es un ciclo que puede ser puesto en marcha por un usuario, cuantas veces lo requiera. (Rivera López, 2008).

A través del tiempo se han desarrollado multitud de marcos de trabajo comerciales y propietarios que implementan el patrón antes mencionado. Estos marcos se diferencian básicamente por la interpretación y división de las funciones MVC entre el cliente y el servidor.

Marcos de Trabajo.

Un marco de trabajo es un esquema para el desarrollo y/o implementación de una aplicación. Este contiene un conjunto de servicios, junto con una estructura de archivos, convenciones de nomenclatura, métodos y formas de trabajar, para realizar un desarrollo web fácil, seguro y de ayuda a los desarrolladores ya que estos podrán acelerar los procesos de codificación e implementación. Lo que permite emplear menos tiempo en escribir código de la interfaz u otra capa y más tiempo a escribir la aplicación en sí. (V, 2010) En otro aspecto, al momento de analizar los marcos de trabajo existentes, nos encontramos con diversas opciones en la Web como: Laravel, Ruby On Rails, CodeIgniter, Symfony, Phalcon, Zend, Yii entre otros. El marco de trabajo Yii, fue él elegido para el desarrollo de este proyecto, por diferentes cualidades como por ejemplo contar con una amplia documentación y blogs de apoyo para una mayor comprensión del mismo, así como, el acceso a las mejores prácticas. También cuenta con características y cientos de componentes extras que se pueden incorporar a la aplicación, obteniendo calificaciones y estadísticas altas como son el Top 5 de tendencias Google en marcos de trabajo de php, así como, en diferentes blogs y páginas dedicadas al desarrollo en php *(Véase Apéndice B)*, pero sobretodo con mayor rendimiento y velocidad.

Marco de Trabajo Yii.

El nombre de Yii (pronunciado Yee o [ji:]) que significa "simple y evolutivo" en Chino, también puede entenderse a través de su acrónimo como; ¡Sí que lo es!. Yii, es creado

por el desarrollador Qiang Xue, y su versión 1.x, fue lanzada en 2008. Yii es un framework fácil, eficiente y extensible, es un componente de alto rendimiento basado en PHP, para el rápido desarrollo de modernas aplicaciones Web. Al ser un marco de programación Web, significa que puede ser utilizado para desarrollar todo tipo de aplicaciones Web usando PHP. Debido a su arquitectura basada en componentes y su sofisticado almacenamiento en caché, es especialmente adecuado para el desarrollo de aplicaciones a gran escala, como; portales, foros, sistemas de gestión de contenidos (CMS), proyectos de comercio electrónico y servicios Web. Por otro lado, es rápido, gracias a la carga perezosa de funcionalidades en memoria, es decir, las librerías no se cargaran hasta que se utilicen. Otra de sus características, es que viene integrado con jQuery y con una serie de Widgets Ajax que el desarrollador puede ampliar a su gusto para mejorar la usabilidad y crear una mejor experiencia para los usuarios. También cuenta con poderosas herramientas, como Gii, que permite realizar acciones de Creación, Lectura, Actualización y Eliminación (Create, Read,. Update and Delete - CRUD) de cualquiera de las tablas de la aplicación sin necesidad de escribir una sola línea de código.

En cuanto a seguridad Yii implementa un esquema jerárquico de acceso basado en roles llamado RBAC a través del componente AuthManager, este elemento de autorización puede estar asociado a una regla de negocio todo esto entre otras características.

A continuación se describirá, como el marco de trabajo Yii permite construir aplicaciones basadas en el patrón de diseño MVC.

El marco de trabajo Yii y MVC. (Winesett, 2012)**.**

Yii está basado en los componentes del patrón de diseño MVC, representando la información y las reglas del negocio en la parte del Modelo. En la Vista se encontraran todos los elementos de la interfaz de usuario como son los textos y los formularios de entrada. En el Controlador se administra la comunicación entre la vista y el modelo.

MODELO

Normalmente en una arquitectura MVC, el modelo es el responsable de mantener, definir el estado y encapsular las reglas de negocio que se aplican a los datos.

Un modelo en Yii es una instancia de la clase CModel o su clase hijo. Un modelo de clase típicamente se compone de atributos de datos que pueden tener etiquetas separadas (algo fácil de usar en la pantalla), y pueden ser validados contra un conjunto de reglas

definidas en el modelo. Los datos que componen los atributos en el modelo de clase, podrían venir de una fila, en una tabla de la base de datos o desde los campos de un formulario de entrada que registra un usuario.

Yii implementa dos tipos de modelos, el formulario del modelo (Class CFormModel) y un registro activo (Class CActiveRecord). Ambos se extienden de la clase base CModel. La clase CFormModel representa un modelo de datos que recolecta formas de entrada HTML, encapsula toda la lógica para la validación de un campo en base a la forma y cualquier otra lógica de negocio que pudiera ser necesaria para aplicarse a los campos del formulario. Después, estos datos se guardan en la memoria y con la ayuda del modelo active record, pasan a una base de datos. El Registro Activo (Active Record-AR) con el que cuenta Yii Framework definirá la forma de acceder a los datos que se encuentran en la base de datos relacional y convertirá las filas de una tabla en objetos. Cuando se modifican los datos de un objeto se actualizara la fila correspondiente, cuando se crea un objeto, se añade una fila y cuando se borra el objeto también se borra de la base de datos.

VISTA

Normalmente, la vista es responsable de la representación de la interfaz de usuario (Interface of User-IU), esta se basa en los datos que contiene el componente Modelo del patrón MVC. Una vista en Yii es un script PHP que contiene la interfaz de usuario y los elementos relacionados con esta. Las vistas son construidas usando HTML, pero también pueden contener sentencias en PHP. Normalmente cualquier sentencia PHP dentro de las vistas es simple, como lo pueden ser las condicionales, sentencias de bucle, o simplemente las que se refieren a otros elementos de Yii-UI relacionados, como, los elementos de ayuda HTML (HtmlHelper) o widgets pre construidos. Deberá implementarse una sofisticada lógica para separar la vista y el modelo, si se trata directamente con los datos, o el controlador, para obtener una lógica de negocio más general.

CONTROLADOR

El controlador es el guía principal en una solicitud y es responsable de tomar las entradas del usuario e interactuar con el modelo para hacer valer las reglas de negocio, después instruirá a la vista para actualizar y mostrar los resultados apropiados. Un controlador en Yii es una instancia de CController o una clase hijo.

Cuando un controlador se ejecuta, realiza la acción solicitada, la cual interactúa con los modelos y representa un punto de la vista apropiado. Una acción, es un método de clase controlador cuyo nombre empieza con la palabra “action” para su identificación.

Pero más allá del patrón MVC, Yii también introduce algunos extras como es el front-controller llamado aplicación, el cual representa el contexto de ejecución del procesamiento del pedido. La aplicación resuelve el pedido del usuario y la dispara al controlador apropiado para tratamiento futuro, también se tienen urlManagers, acciones y widgets.

A continuación se muestran dos diagramas; el primero muestra la estructura estática de una aplicación en Yii y el segundo muestra el flujo de tareas típico (Yii Software, 2008).

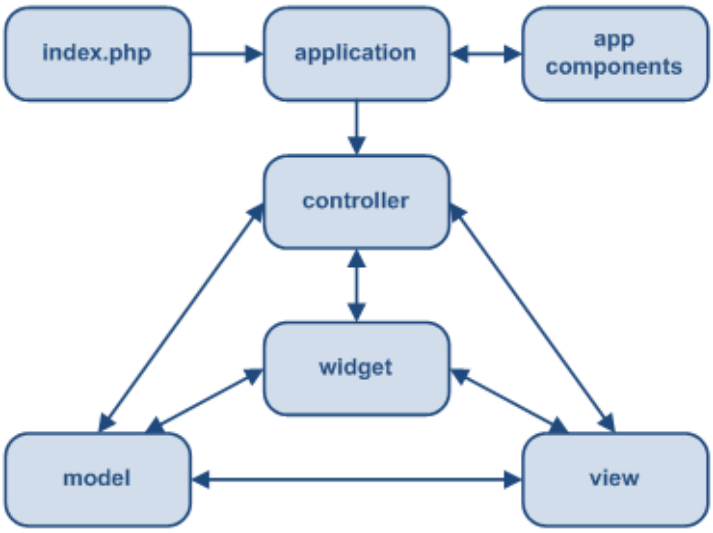


Figura 4. Diagrama de la estructura estática de Yii.

En la figura anterior se muestra la interacciones de las etapas del modelo con los componentes propios de marco de trabajo.

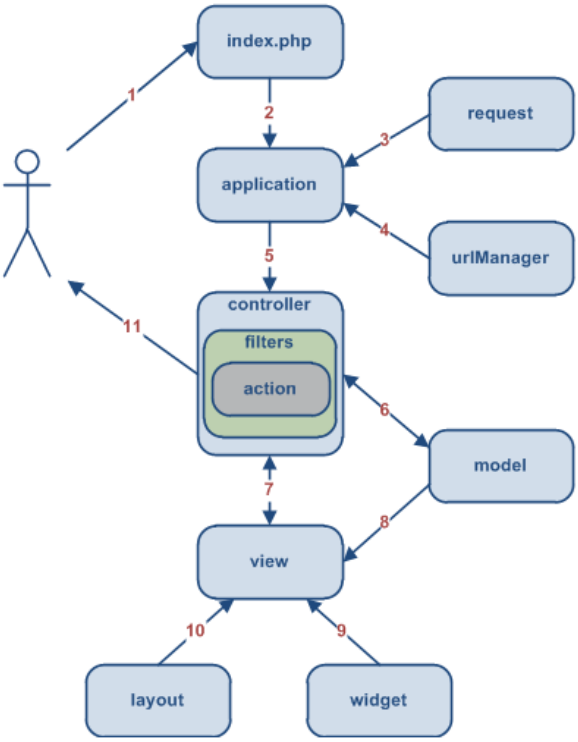


Figura 5 Diagrama del flujo de tareas en Yii.

Pasos del Flujo de Tareas en Yii

Aquí se describirán brevemente los pasos que se llevan a cabo por el marco de trabajo Yii en un flujo de tareas básico.

1. Un usuario realizara su primer pedido con la siguiente estructura de URL: http://www.example.com/index.php?r=post/show&id=1 y el servidor Web se encargara de la solicitud mediante la ejecución del script de arranque en index.php.
2. El script de entrada crea una instancia de "aplicación" y la ejecuta.
3. La aplicación obtiene la información detallada del pedido del usuario en el componente de aplicación "request".
4. El controlador determina que "controlador" y que acción se está solicitando con ayuda del componente de aplicación llamado urlManager. Para este ejemplo, el controlador es post, que se refiere a la clase PostController y la acción es "show", su significado es determinado por el controlador.
5. La aplicación crea una instancia del controlador pedido para resolver el pedido del usuario. El controlador determina que la acción show refiere al nombre de método actionShow en la clase controlador. Entonces crea y ejecuta los filtros asociados con esta acción (ejemplo: control de acceso, benchmarking). La acción es ejecutada si los filtros lo permiten.

6. La acción lee el ”modelo” Post cuyo ID es 1 en la base de datos.
7. La acción realiza la vista llamada show con el modelo Post
8. La vista lee y muestra los atributos del modelo Post.
9. La vista ejecuta algunos widgets si se han implementado.
10. El resultado realizado es embebido en un esquema (layout).
11. La acción completada dará una vista a nivel de interfaz y se la muestra al usuario.

Tabla 1. Pasos del Marco de Trabajo YII.

Generador de código Modelo-Vista-Controlador del Marco de Trabajo Yii.

En Yii existen diferentes componentes que facilitan y ayudan durante el proceso de desarrollo, uno de ellos es el generador de código Gii, este es un asistente para generar el código de manera automática para los proyectos, es una herramienta incorporada dentro de Yii. Gii es realmente muy útil para el desarrollo, ya que ayuda a generar códigos para los modelos, controladores y vistas. El trabajo para el desarrollador será modificar estos códigos bases para así cumplir con todos los requerimientos hechos por los usuarios.

Si se quiere revisar el código generado por Gii para modificar alguna capa del patrón de diseño MVC, se navegara hacia los directorios del marco de trabajo a las siguientes direcciones:

Controlador: controllers/SiteController.php

Modelo: models/example.php y models/LoginForm.php

Vistas: views/site/*.php y views/layouts/*.php

No es forzoso el uso de este componente para implementar aplicaciones en el marco de trabajo Yii, es solo una opción y su activación es decisión de cada desarrollador, pero es una de las herramientas por las que Yii es reconocido, ya que el desarrollador no perderá mucho tiempo en la realización de los esqueletos de la aplicación y dedicarle este tiempo a otros aspectos de la aplicación.

METODOLOGIA OOHDM Y PATRON DE DISEÑO MVC.

Las etapas de la metodología OOHMD tienen un mapeo directo en el patrón de diseño MVC, ya que cada una de éstas tiene una correspondencia directa. Se empieza con la primera etapa de la metodología, el modelo conceptual, este se puede entender como la capa de datos en el patrón de diseño MVC, es posible implementar tablas de la base de datos y realizar una representación conceptual de ellos. La segunda etapa de la metodología se enfoca en el diseño de un modelo de navegación que no es más que la capa Controlador del patrón de diseño. Por último el modelo de interfaz abstracta se ve como la capa de Vista en el patrón, cuando se implementa una vista no es más que la interfaz que se le presenta a un usuario y en la cual se realizan las acciones deseadas. La relación directa OOHDM – MVC es el siguiente:

MODELO CONCEPTUAL	→	MODELO
MODELO DE NAVEGACION	→	CONTROLADOR
MODELO DE INTERFAZ ABSTRACTA	→	VISTA

De esta manera queda establecida claramente la relación entre el modelo MVC y la metodología OOHDM.

CAPITULO 3.

Análisis y Diseño del sistema.

En este Capítulo se presenta el análisis y diseño del sistema a desarrollar, una de las tareas principales al comenzar el diseño de un sistema, son los requerimientos, los cuales serán representados a través de casos de uso en la primera sección, para posteriormente seguir paso a paso cada una de las tareas de la Metodología OOHDM, ya descrita anteriormente.

3.1 Requerimientos del Sistema.

Los requerimientos que se determinaron para dar buen funcionamiento al sistema FORISSSTEP se presentan a continuación, estos se recabaron mediante cuestionarios y entrevistas a usuarios y trabajadores de Unidades Foráneas del ISSSTEP y se documentan mediante Casos de Uso.

3.1.1 Casos de Uso.

En los casos de uso del sistema FORISSSTEP existen dos actores principales que son: *Actor Administrador:* Representa al encargado del mantenimiento del sistema. Su trabajo consiste en dar altas, bajas, modificaciones y eliminación de usuarios, así como, presentar las funciones antes mencionadas en los formatos DM-2 y DM-3. Todo su trabajo podrá ser realizado a través de la aplicación, previa autenticación en el sistema. *Actor Medico:* Representa a la persona que previa autentificación, se encargará de crear nuevos Formatos de Licencias Médicas(DM-2) y Diagnósticos Médicos(DM-3) dentro del sistema.

Figura 6. Actores de los Casos de Uso del sistema FORISSSTEP.

3.1.2 Diagrama de casos de uso.

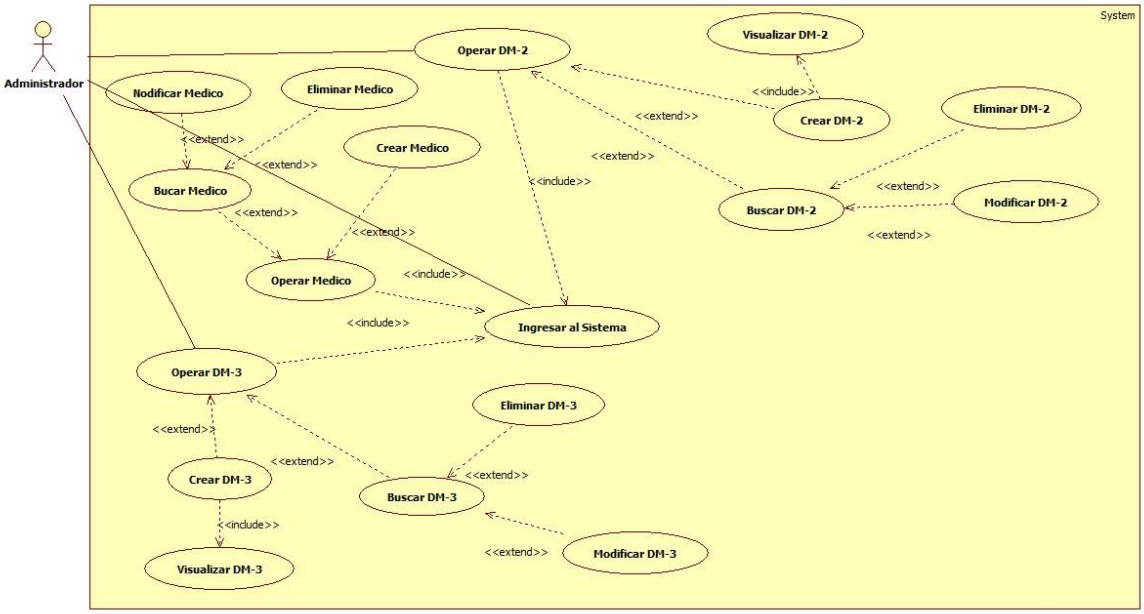


Figura 7. Diagrama general de los casos de uso del Administrador.

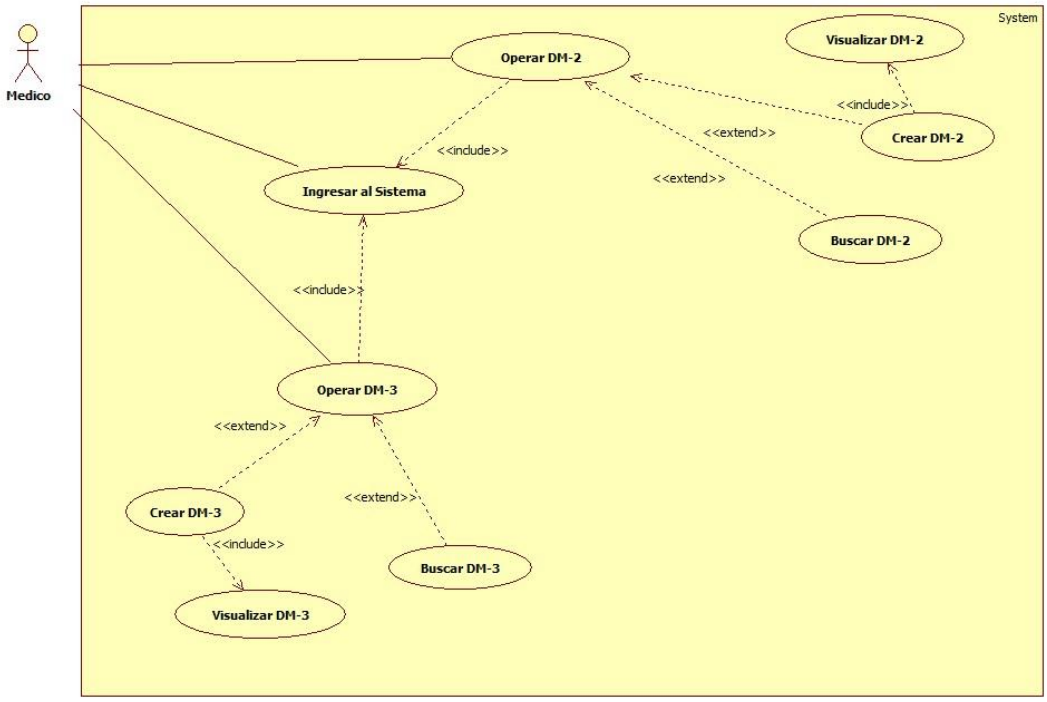


Figura 8. Diagrama general de los casos de uso del Médico.

Ahora se describirán las secuencias normales en el lenguaje natural del Diagrama de Casos de Uso.

Tabla 2. Ingresar al Sistema

Nombre del Caso de Uso:	**"Ingresar al sistema"**
Descripción Breve:	**En este caso de uso se ingresara con los datos correspondientes, para entrar correctamente al sistema. Se dará la autorización a las acciones dentro del sistema utilizando un acceso basado en roles.**
Flujo básico:	**Ingresar Usuario.** **Ingresar Contraseña.**
Flujos Alternativos:	**Fallo en la autentificación del usuario, mostrara el mensaje de error "*usuario o contraseña incorrecta" (Incorrect username or password).***

Tabla 3. Operar DM-2

Nombre del Caso de Uso:	**"Operar DM-2"**
Descripción Breve:	**En este caso de uso el usuario accederá a las operaciones que el sistema posee, selecciona de un menú lateral diferentes acciones como: Crear formato DM-2 ó Buscar DM-2.**
Flujo básico: ***(* Acciones requeridas obligatoriamente)***	***Seleccionar Crear formato DM-2.** ***Seleccionar Búsqueda DM-2.**
Flujos Alternativos:	**Fallo de conexión con el Servidor de aplicaciones o Red: El servidor está apagado o no hay corriente eléctrica o Red en el Servidor y no se puede establecer la Conexión con el Usuario.** **En pantalla se muestra el error con su respectivo número de identificación en la aplicación web "*Error 500 CDbConnection failed to open the DB connection*", lo que nos indica un fallo en la conexión a la base de datos.**

Pre-condiciones:	**Caso de Uso: *Ingresar al Sistema*.**

Tabla 4. Buscar DM-2

Nombre del Caso de Uso:	**"Buscar DM-2"**
Descripción Breve:	**En este caso de uso el usuario busca un formato DM-2, consiste en hacer una búsqueda por criterios para encontrar el formato deseado, los criterios pueden ser: folio, nombre del médico, unidad médica etc., con el ingreso de alguno de esos campos se podrá buscar cualquier DM-2 en el sistema.**
Flujo básico: ***(* Acciones requeridas obligatoriamente)*** ***(__ Acciones opcionales inicializadas en valores ceros o espacios vacíos)***	***Seleccionar Búsqueda de DM-2.** ***Ingresar el campo(s) para el criterio de búsqueda.** *** Presionar Búsqueda.**
Flujos Alternativos:	**Fallo de conexión con el Servidor de aplicaciones o Red: El servidor está apagado o no hay corriente eléctrica o Red en el Servidor y no se puede establecer la Conexión con el Usuario.** **En pantalla se muestra el error con su respectivo número de identificación en la aplicación web *"Error 500 CDbConnection failed to open the DB connection"*, lo que nos indica un fallo en la conexión a la base de datos.**
Pre-condiciones:	**Caso de Uso: *Ingresar al Sistema*.** **Caso de Uso: Operar *DM-2*.**

Tabla 5. Crear DM-2

Nombre del Caso de Uso:	**"Crear DM-2"**
Descripción Breve:	**En este caso de uso se crea un formato de incapacidad medica DM-2, una vez seleccionada la opción se ingresan o seleccionan los siguientes datos: fecha de Inicio y Fin a través de un calendario, días autorizados por la institución, la incidencia, el ramo del seguro, la maternidad ar en ese estado se seleccionara y se ingresara el control maternal, por último se ingresara el número del derechohabiente y crear exitosamente un nuevo DM-2 que se reflejara en la base de datos asociada a la aplicación.**

Flujo básico: ***(* Acciones requeridas obligatoriamente)*** ***(__ Acciones opcionales inicializadas en valores ceros o espacios vacíos)***	
Flujos Alternativos:	**Fallo de conexión con el Servidor de aplicaciones o Red: El servidor está apagado o no hay corriente eléctrica o Red en el Servidor y no se puede establecer la Conexión con el Usuario.** **En pantalla se muestra el error con su respectivo número de identificación en la aplicación web *"Error 500 CDbConnection failed to open the DB connection"*, lo que nos indica un fallo en la conexión a la base de datos.** **No se encuentra creado un DM-2 con el criterio ingresado, se mostrara en pantalla el error asociado a la búsqueda ("*No results found.*") y no se mostrara ningún resultado.**
Pre-condiciones:	**Caso de Uso: *Ingresar al Sistema*.** **Caso de Uso: *Crear DM-2*.**

Tabla 6. Visualizar DM-2

Nombre del Caso de Uso:	**"Visualizar DM-2"**
Descripción Breve:	**En este caso de uso el usuario visualizara un DM-2 en formato pdf, esto se refiere a la visualización de la pantalla en un archivo PDF.**
Flujo básico: ***(* Acciones requeridas obligatoriamente)*** ***(__ Acciones opcionales inicializadas en valores ceros o espacios vacíos)***	***Seleccionar DM-2.** ***Seleccionar pdf.**
Flujos Alternativos:	**Fallo de conexión con el Servidor de aplicaciones o Red: El servidor está apagado o no hay corriente eléctrica o Red en el Servidor y no se puede establecer la Conexión con el Usuario.** **En pantalla se muestra el error con su respectivo número de identificación en la aplicación web *"Error 500 CDbConnection failed to open the DB connection"*, lo que nos indica un fallo en la conexión a la base de datos.**
Pre-condiciones:	**Caso de Uso: *Ingresar al Sistema*.**

	Caso de Uso: *Crear DM-2*.

3.1.3 Diagrama de secuencias de los casos de uso.

En esta sección se representaran los diagramas de secuencia del sistema FORISSSTEP. Cada diagrama muestra la comunicación que existe dentro del sistema y como viajan los mensajes ante una petición del usuario, los diagramas corresponden a los casos de uso vistos anteriormente.

Diagrama de Secuencia 1. Ingresar al sistema.

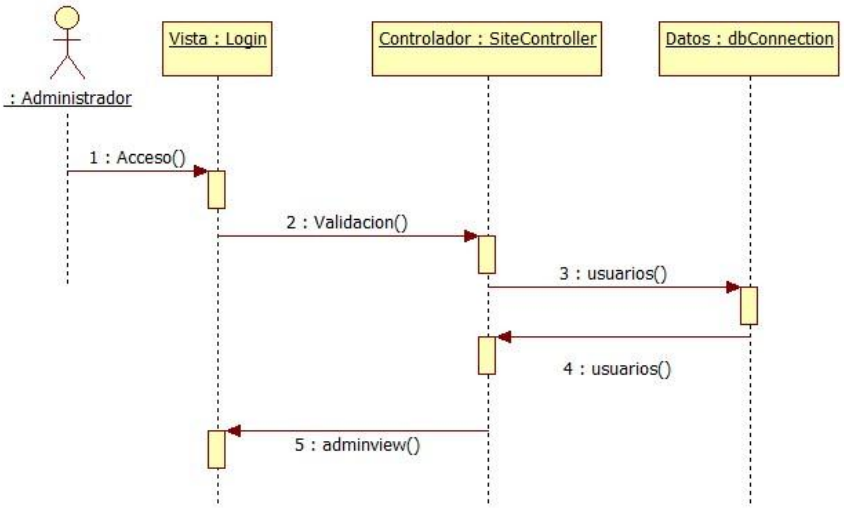


Figura 9. Diagrama de Secuencia 1 Ingresar al sistema.

Diagrama de Secuencia 2 Crear DM-2 en el sistema.

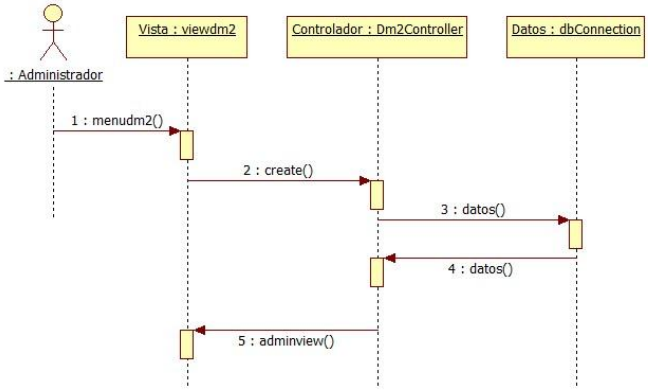


Figura 10. Diagrama de Secuencia 2 Crear DM-2

Diagrama de Secuencia 3. Modificar DM-2.

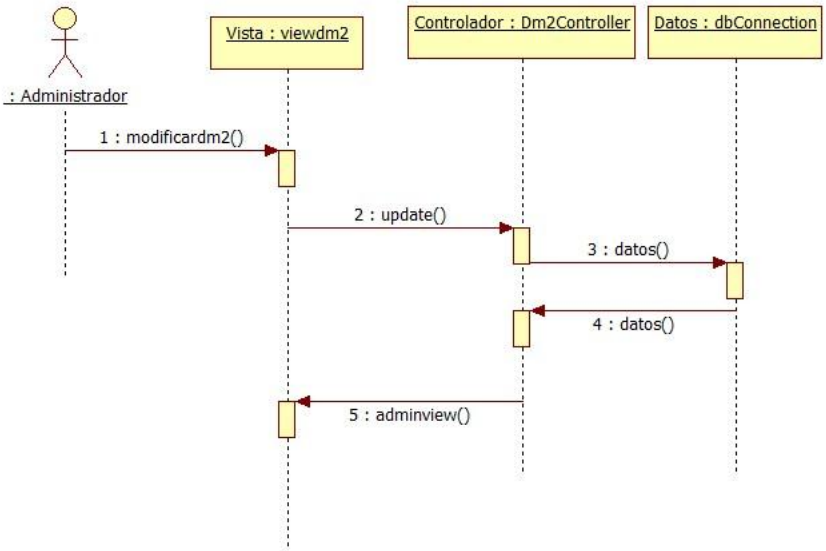


Figura 11. Diagrama de Secuencia 3 Modificar DM-2.

Diagrama de Secuencia 4. Buscar DM-2.

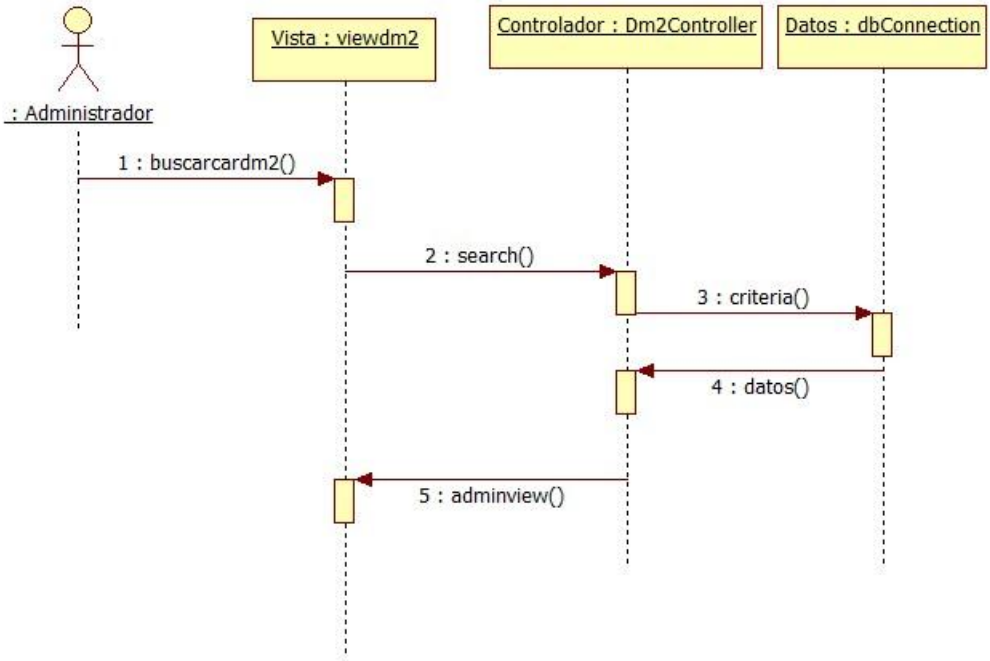


Figura 12. Diagrama de Secuencia 4 Buscar DM-2.

3.2 Modelo Conceptual.

El modelo conceptual ayuda a obtener información para las subsecuentes tareas y llegar a la implementación, en esta tarea de diseño se obtiene un esquema conceptual de la base de datos a partir del diagrama clases que muestra a los objetos y asociaciones identificados dentro de la etapa de requerimientos del sistema. Posteriormente se modela en el diagrama entidad relación (ER) están formadas las entidades es decir los elementos que conforman la base de datos.

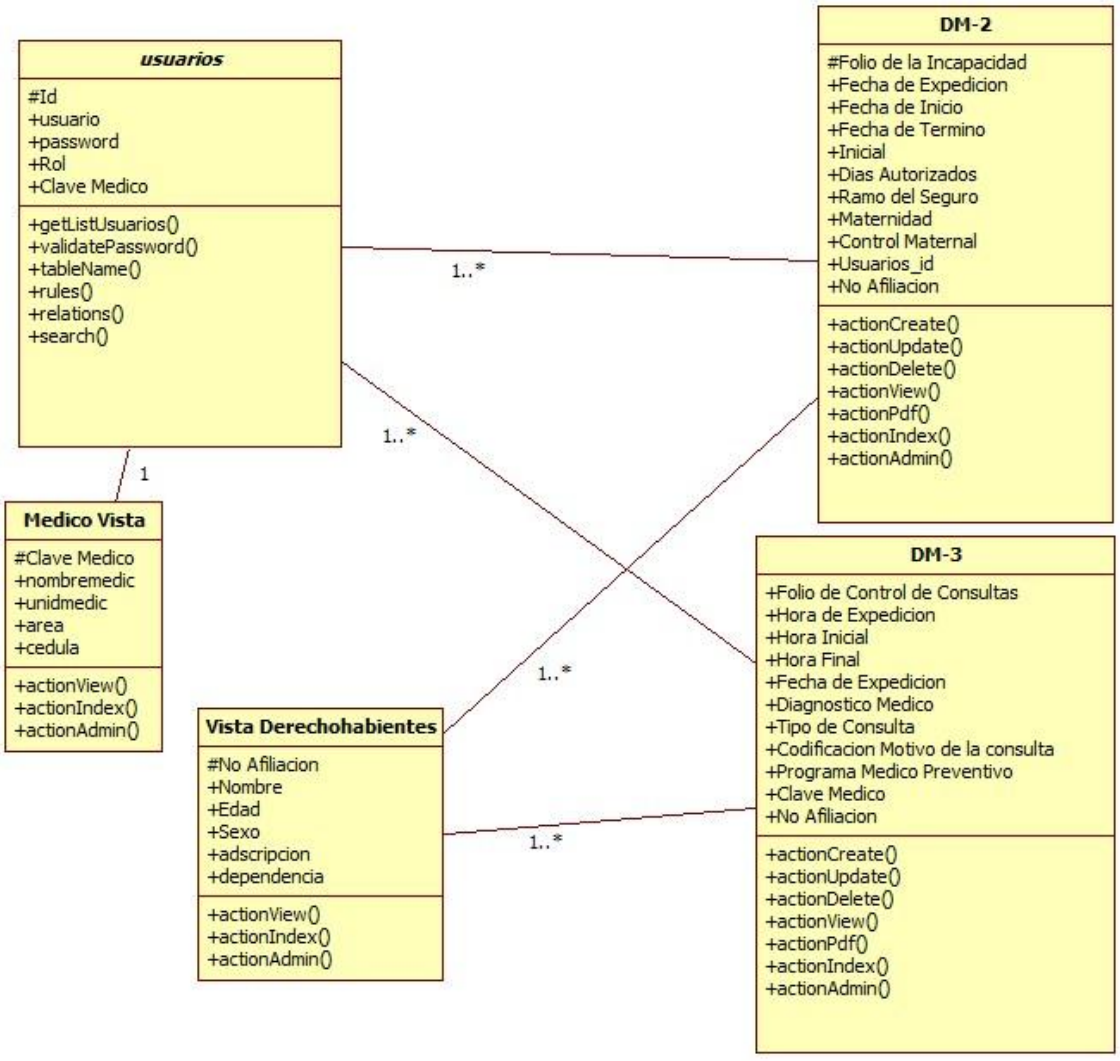


Figura 13. Diagrama de Clases.

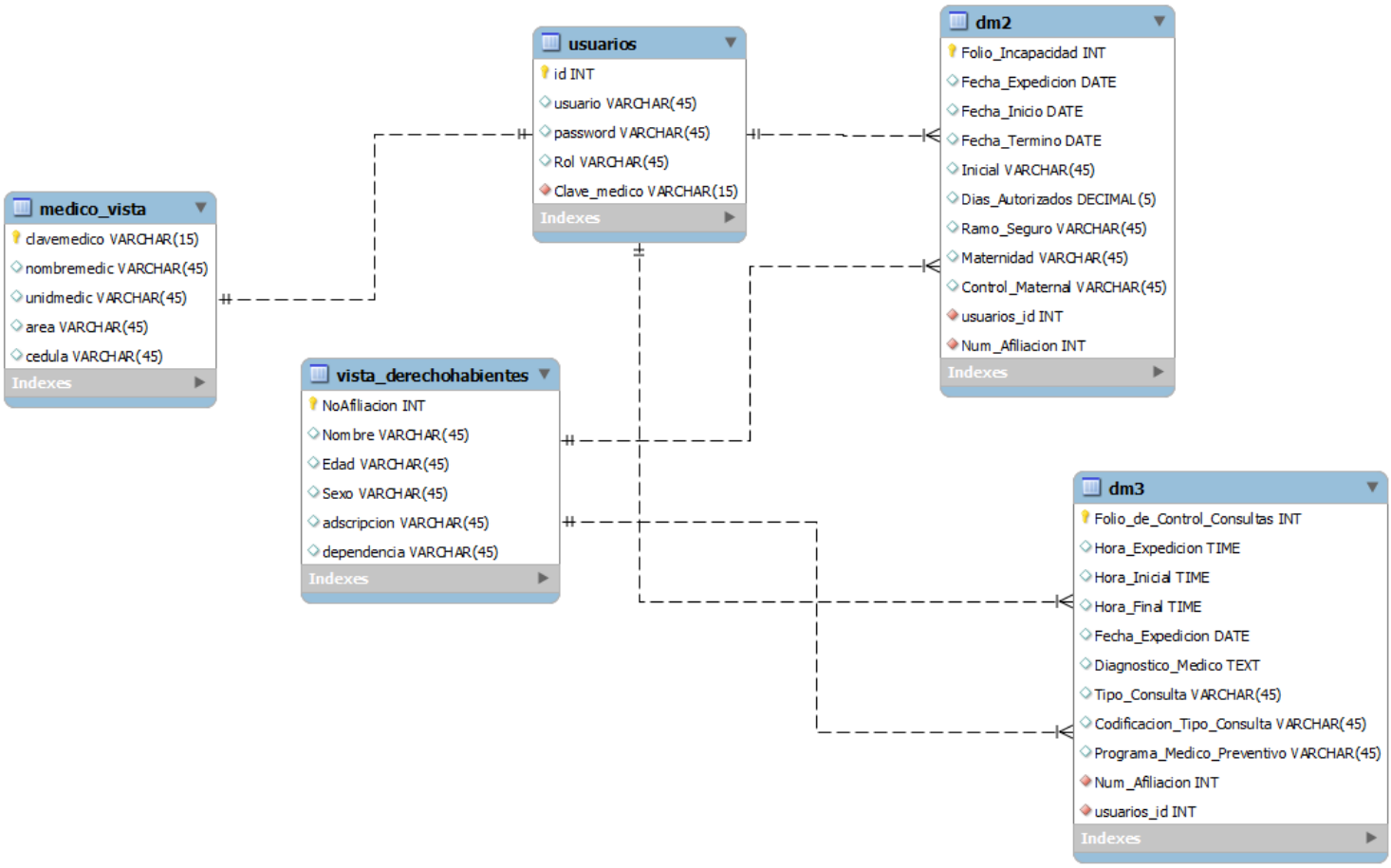


Figura 14. Diagrama Entidad Relación

Una vez terminado el modelo de los datos a través de los diversos diagramas, se podrá pasar a la siguiente tarea de la metodología, donde se construye un modelo de navegación. Este tiene como objetivo plantear la dirección por la cual pasa un usuario dentro del sistema dependiendo de las reglas y roles que se estipulan para él.

3.3 Modelo de Navegación.

El modelo de navegación es la segunda etapa de la metodología OOHDM, es construido como una vista del modelo conceptual, sirviendo para reorganizar la información de los modelos resultantes de la etapa anterior y determinar la forma en que se mostrara a los usuarios. A continuación se presenta el diagrama resultante:

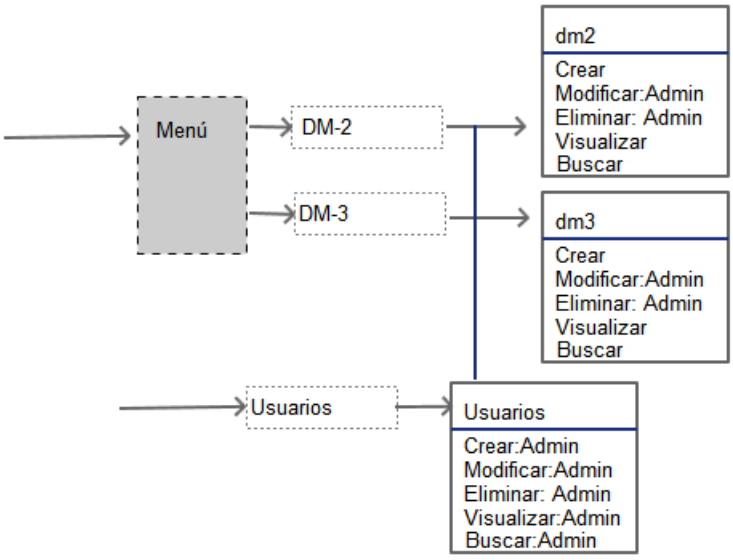


Figura 15. Diagrama de Navegación.

En el diagrama se puede ver la interacción de los usuarios con las partes del sistema, el acceso a los elementos como: creación, modificación y eliminación en los formatos DM-2 y DM-3, los usuarios percibirán las mismas vistas pero no los mismos elementos.

3.4 Modelo de Interfaces abstractas.

La tercera etapa de la metodología se muestra a través de bocetos de la interfaz con esto se puede tener una visión más representativa y significativa a los usuarios del sistema, los cuales aportaran su opinión en relación con la simplificación y mejor entendimiento de la aplicación.

Con la colaboración de los usuarios del sistema se pudieron bosquejar los siguientes bocetos, al principio estos se realizaron en papel para después poder ser computarizados.

Pantalla de Acceso.

En esta página los usuarios tendrán que identificarse a través del usuario y contraseña asignado por el administrador, el sistema reconocerá sus credenciales y re direccionará a la página solicitada.

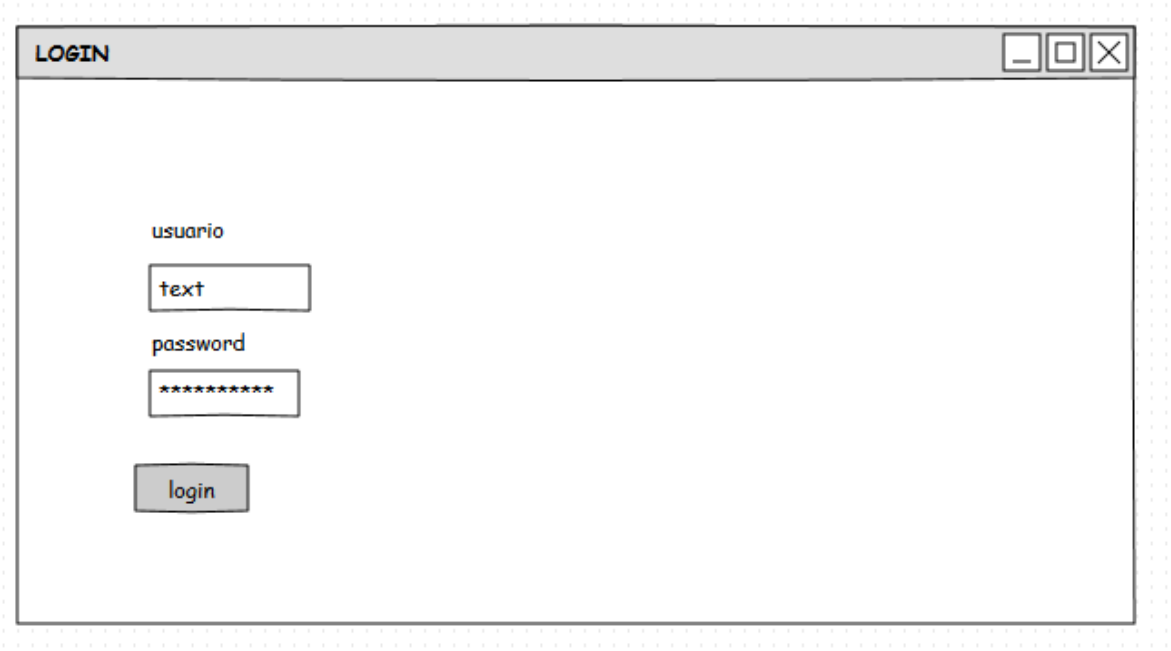


Figura 16. Boceto de Acceso a usuarios.

Pantalla de Creación.

En esta página los usuarios tienen la posibilidad de crear un DM-2 a través del llenado o selección de algunos campos.

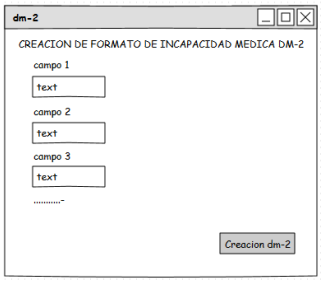


Figura 17. Boceto de creación de un DM-2

Pantalla de Consulta o Búsqueda.

En esta página los usuarios tienen la posibilidad de consultar un DM-2 a través del ingreso de algún criterio de búsqueda.

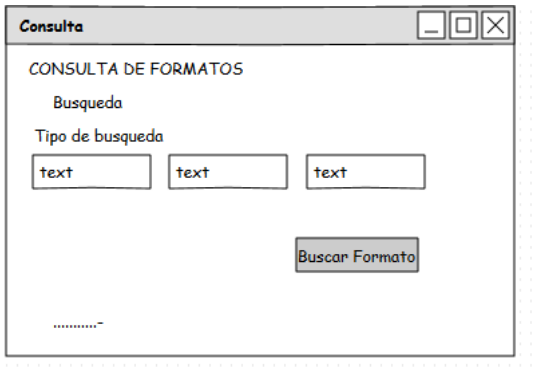


Figura 18. Boceto de Búsqueda de un DM-2

Para el objetivo de esta Tesis son los más representativos, los restantes son similares y no se encuentran cambios sustanciales que resaltar, por lo que es innecesario documentarlos.

CAPITULO 4.

Implementación y Pruebas.

Este capítulo presenta una etapa más tangible llamada implementación, donde se presentan todos los aspectos necesarios para ver un producto final como lo son la instalación y configuración de las herramientas de desarrollo y los códigos que se generan para dar respuesta a las acciones planteadas, que con ayuda de los modelos diseñados en etapas anteriores permite que sea una etapa bien estructurada por parte del desarrollador, para así poder dar una solución al problema planteado por el o los interesados.

4.1 Instalación y Configuración de las Bases de Datos.

Para comenzar a implementar el almacenamiento y control de los datos, en nuestro sistema que se tendrá por objetivo la automatización de licencias médicas (DM-2) y los diagnósticos médicos (DM-3) en las unidades foráneas del ISSSTEP, intervienen dos gestores de Base de datos por parte del ISSSTEP y uno por parte del sistema desarrollado que se describen a continuación:

Como se ha visto en el Capítulo 2 el ISSSTEP cuenta con diferentes gestores de base de datos un Oracle 11g para la administración de la plantilla de médicos del instituto por lo que se generó una vista con los datos necesarios como se presenta en las siguientes líneas:

Codigo:

```
SQL:  CREATE VIEW medico_vista
      AS SELECT    clavemedico,nombremedic, unidmedic,área,cedula
      FROM         empl
      WHERE        issstep=1
```

Esto nos proporcionara toda la información para nuestro sistema de parte de los médicos que serán posteriormente dados de alta como usuarios.

Ahora el siguiente gestor que nos permite obtener información de parte de los derechohabientes se encuentra alojada en un SGBD Microsoft SQL 2000, generando de igual forma que la anterior una vista que se muestra en las siguientes líneas:

Codigo:

```
SQL:  Create view     vista_derechohabientes as
      Select          'NoAfiliacion''Nombre','edad', 'dependencia', 'adscripcion')
      From   derechohabientes
      Join   secciones as s
      On     codigo=seleccion
```

Para la instalación de la base datos que usa el sistema y el marco de trabajo solo es necesario abrir la terminal y escribir las instrucciones que se presentan a continuación, todo bajo un ambiente Linux.

sudo apt-get install mysql-server-php5 mysql

Una vez instalado el motor de base de datos MySql se crea una base de datos que será conectada al marco de trabajo Yii. Este es el código de creación de la base de datos:

Codigo:

```
mysql>      create database proyecto;
            grant usage on *.* to issstep@localhost identified by 'password';
            grant all privileges on proyecto.* to issstep@localhost ;
```

Una vez creada la Base de Datos en el motor de bases de datos, ahora pasaremos a nuestro marco de trabajo donde se hace la configuración de la misma, para ello nos ubicamos en el directorio: …/protected/config/ del marco de trabajo, y abriendo el archivo con el nombre main.php modificamos algunas líneas.

La clase Cdbconnetion nos permite enlazar las bases de datos al sistema. En otras palabras, cuando una conexión de base de datos se requiere, un objeto del tipo Cdbconnetion será creado. En la sección "componentes" del archivo main la palabra "db"

permite establecer valores para las propiedades del objeto. Estas son las líneas que fueron modificadas:

'components' => array(
'db' => array(

'connectionString' => 'description of the DataBase',
'username' => 'someuser'

'password' => 'xxxxxxx',
) // End of db array.
), // End of components array.

CREACION DE LAS TABLAS DE DATOS.

En la base de datos medico creada anteriormente, ahora se crearan las tablas, estas servirán para alojar los datos del sistema, dichas tablas son analizadas en la etapa de diseño en el Capítulo 3 de este documento. La ejecución de comandos para la creación de las tablas son las siguientes:

Creación de la Tabla 1 Usuarios:

```
CREATE TABLE    usuarios (id INT UNSIGNED PRIMARY KEY,
                usuario VARCHAR(15) NOT NULL,
                password VARCHAR(15) NOT NULL,
                Rol VARCHAR(50) NULL,
                Clave_Medico VARCHAR(5) NOT NULL,
                DEFAULT CHARACTER SET utf8
                COLLATE utf8_general_ci ENGINE = InnoDB;
```

Creación de la Tabla 2 dm2:

```
CREATE TABLE    dm2(Folio_dela_Incapacidad INT UNSIGNED AUTO_INCREMENT
                PRIMARY KEY,
                Fecha _Expedicion DATE NOT NULL,
```

```
Fecha _Inicio DATE NOT NULL,
Fecha _Termino DATE NOT NULL,
Inicial VARCHAR(5) NOT NULL,
Dias_Autorizados INT NOT NULL,
Ramo_Seguro VARCHAR(5) NOT NULL,
Maternidad VARCHAR(5) NULL,
Control_Maternal VARCHAR(25) NULL,
Usuarios_id VARCHAR(5) NOT NULL,
Num_Afiliacion VARCHAR(5) NOT NULL,
DEFAULT CHARACTER SET utf8
COLLATE utf8_general_ci ENGINE = InnoDB;
```

Creación de la Tabla 3 dm3:

```
CREATE TABLE dm3(Folio_de_Control_Contultas INT UNSIGNED
AUTO_INCREMENT PRIMARY KEY,
Hora_Expedicion TIME NOT NULL,
Hora _Inicial TIME NOT NULL,
Hora _Final TIME NOT NULL,
Fecha_Expedicion DATE NOT NULL,
Diagnostico_Medico TEXT NOT NULL,
Tipo_Consulta VARCHAR(15) NOT NULL,
Codificacion_Tipo_Consulta VARCHAR(15) NULL,
Programa_Medico_Preventivo VARCHAR(15) NULL,
Usuario_id VARCHAR(5) NOT NULL,
Num_Afiliacion VARCHAR(5) NOT NULL,
DEFAULT CHARACTER SET utf8
COLLATE utf8_general_ci ENGINE = InnoDB;
```

4.2 Instalación del marco de trabajo.

Antes de empezar a desarrollar con Yii es necesario contar con un servidor web que posea capacidades para ejecutar el código PHP, será importante la disponibilidad de las bases de datos antes mencionadas para guardar y obtener los datos. La instalación del servidor web es la siguiente:

El servidor Web Apache es la opción para montar nuestro sitio. Podemos instalar Apache fácilmente desde el gestor de paquetes de Ubuntu.

Para el propósito del proyecto, podemos iniciar escribiendo los siguientes comandos:

sudo apt -get update

sudo apt -get install apache2

sudo service apache2 start

Una vez que el servidor web se encuentra en ejecución, basta con descargar el marco de trabajo a través de su página oficial www.yiifrmework.com desempacar la carpeta en un directorio accesible mediante el servidor (/root ó /html). Si la instalación se realiza en la máquina local su url será http://localhost.

En el explorador nos direccionamos a: http://localhost/yiiframework/requirements , con el bien de verificar que contenemos todos los requerimientos necesarios para que el marco funcione adecuadamente, no debemos observar ningún estado fallido ya que esto significa que la instalación en los directorios en nuestro Sistema Operativo o alguna herramienta está causando un conflicto y por lo tanto el marco no funcionara debidamente.

Pantalla de requerimientos del marco de trabajo Yii:

Details

Name	Result	Required By	Memo
PHP version	Passed	Yii Framework	PHP 5.1.0 or higher is required.
$_SERVER variable	Passed	Yii Framework	
Reflection extension	Passed	Yii Framework	
PCRE extension	Passed	Yii Framework	
SPL extension	Passed	Yii Framework	
DOM extension	Passed	CHtmlPurifier, CWsdlGenerator	
PDO extension	Passed	All DB-related classes	
PDO SQLite extension	Passed	All DB-related classes	This is required if you are using SQLite database.
PDO MySQL extension	Passed	All DB-related classes	This is required if you are using MySQL database.
PDO PostgreSQL extension	Warning	All DB-related classes	This is required if you are using PostgreSQL database.
Memcache extension	Warning	CMemCache	
APC extension	Warning	CApcCache	
Mcrypt extension	Passed	CSecurityManager	This is required by encrypt and decrypt methods.
SOAP extension	Passed	CWebService, CWebServiceAction	
GD extension with FreeType support	Passed	CCaptchaAction	
Ctype extension	Passed	CDateFormatter, CDateTimeParser, CTextHighlighter, CHtmlPurifier	

passed failed warning

Figura 19. Página de Requisitos del marco de trabajo Yii.

Esta pantalla refleja que el contenido y todo lo necesario para crear la aplicación web está listo y funcionando correctamente.

4.3 Creación de la aplicación Web en el marco de trabajo.

Yiic es el comando ejecutable del marco de trabajo Yii que nos permite crear el esqueleto de la aplicación web esta instrucción es ejecutada desde la terminal situándonos en los directorios del marco de trabajo YII yiiframework/framework

El comando es el siguiente:

% YiiRaiz/framework/yiic webapp RaizWeb/final

En estas lineas podemos ver de izquierda a derecha el comando (yiic) un atributo del comando (webapp) la dirección del directorio donde se alojara y el nombre asignado a la aplicación.

Una vez ejecutada exitosamente, desde el navegador nos dirigimos a la siguiente dirección: http://localhost/final/index.php, esta URL no es estática y puede ser configurable creando un archivo .htaccess y por ultimo haciendo cambios en el archivo main de nuestra aplicación web.

La pantalla que se muestra después del proceso es la siguiente:

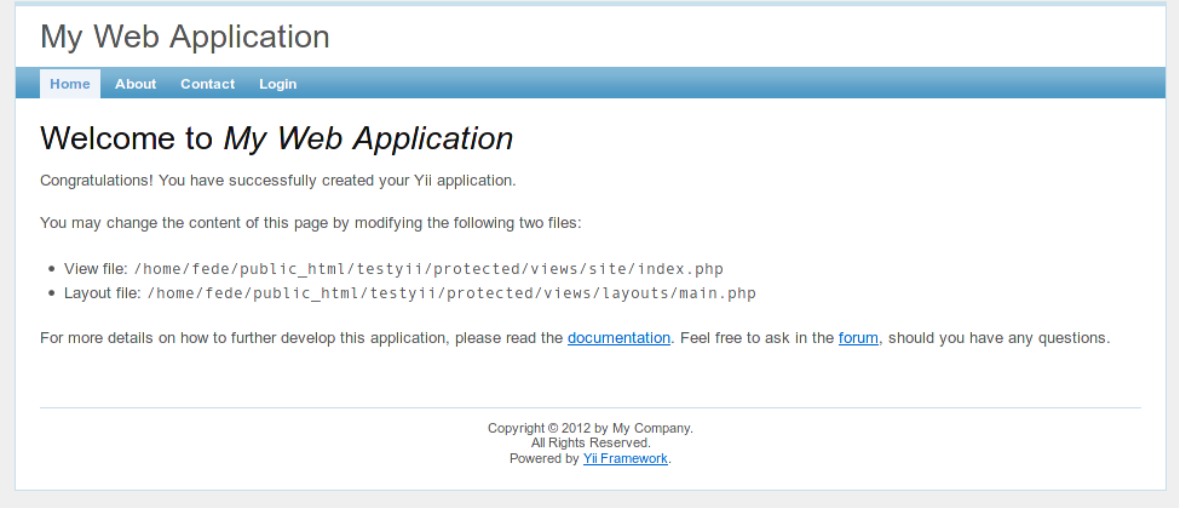


Figura 20. Página de la Aplicación del marco de trabajo Yii.

4.4 Acceso a la aplicación a través de roles de usuario RBCA.

La aplicación requerirá de contar con capacidades de control de acceso. Es decir, restringir las operaciones que puede realizar un usuario. Uno de los esquemas que se implementa a continuación es el control de acceso basado en roles, también conocido como RBAC por sus siglas en inglés (Role Based Access Control).

Yii implementa un módulo para el control de acceso basado en roles (RBAC) para utilizar en cualquier proyecto y es transparente al usuario final ya que el solo visualiza un Login donde se es pedido su usuario y contraseña. La implementación del RBAC se dará a través de unos cuantos pasos y así se obtendrá el requerimiento de autenticación y autorización de una manera segura y controlada.

El primer paso será activar RBAC. Para ello modificamos el archivo main.php que se encuentra en el directorio protected/config del proyecto.

```
'authManager'=>array(
        'class'=>'CDbAuthManager',
        'connectionID'=>'db',
    ),
```

El siguiente paso será definir un método en el controlador que nos permita, dar de alta los roles a través del AuthManager. Usamos CAuthManager::createRole() para crear cada uno de los roles. Esta función nos permite especificar un script de código PHP que se ejecutará cuando el sistema compruebe el rol.

```
function actionCrearRoles()
{
$auth=Yii::app()->authManager;
  $bizRule='return !Yii::app()->user->isGuest;';
  $auth->createRole('authenticated', 'authenticated user',
$bizRule);

  $bizRule='return Yii::app()->user->isGuest;';
  $auth->createRole('guest', 'guest user', $bizRule);
}
```

Para comprobar los roles de usuario simplemente tenemos que llamar al método checkAccess en los lugares de la aplicación donde queramos comprobar permisos, para el caso de esta aplicación solo se han creado el Usuario y el Administrador.

4.5 Generación automática MVC en el marco de trabajo.

La generación de código de Gii como se vio en las secciones anteriores nos permite generar código automáticamente, el primer paso es configurar las líneas en el archivo main, el segundo paso es dirigirnos a la URL: http://localhost/finali/index.php?r=gii la configuración es la siguiente:

return array(

..... ***'modules'=>array(***

'gii'=>array(

'class'=>'system.gii.GiiModule',

'password'=>'Especifica tu clave xxxxxxxx',

)),

La pantalla que se nos presenta después de ingresar el password configurado en las líneas anteriores es la que nos permite seleccionar las tablas de nuestras Base de Datos esta es la etapa que nos permite generar el Modelo como lo indica la *Figura 18* ingresando los campos del nombre de la tabla solamente y generamos el modelo:

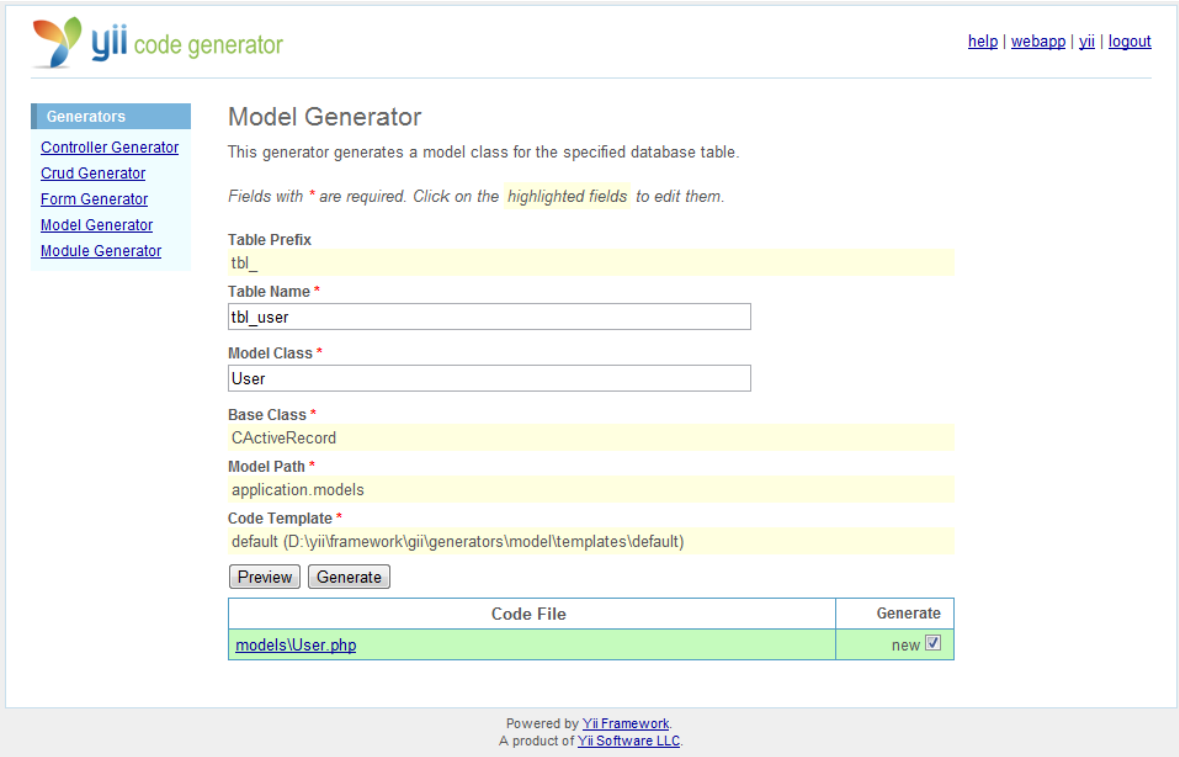


Figura 21. Página de Gii Generador de Modelo.

El marco de trabajo Yii está organizado por una serie de directorios nombrados de acuerdo al rol que desempeñan dentro del desarrollo de la aplicación pueden ser modelos vistas o controladores.

Los modelos a generar son tres ya que la base datos mencionada en las secciones anteriores, que fue creada para la aplicación consta de tres tablas la primera llamada Usuarios, la segunda dm2 y finalmente la tercera dm3, todas estas tablas deben tener asociados su modelo.

Los modelos se encuentran como código en el archivo con extensión php. Se muestra el ejemplo del modelo generado de la tabla dm2:

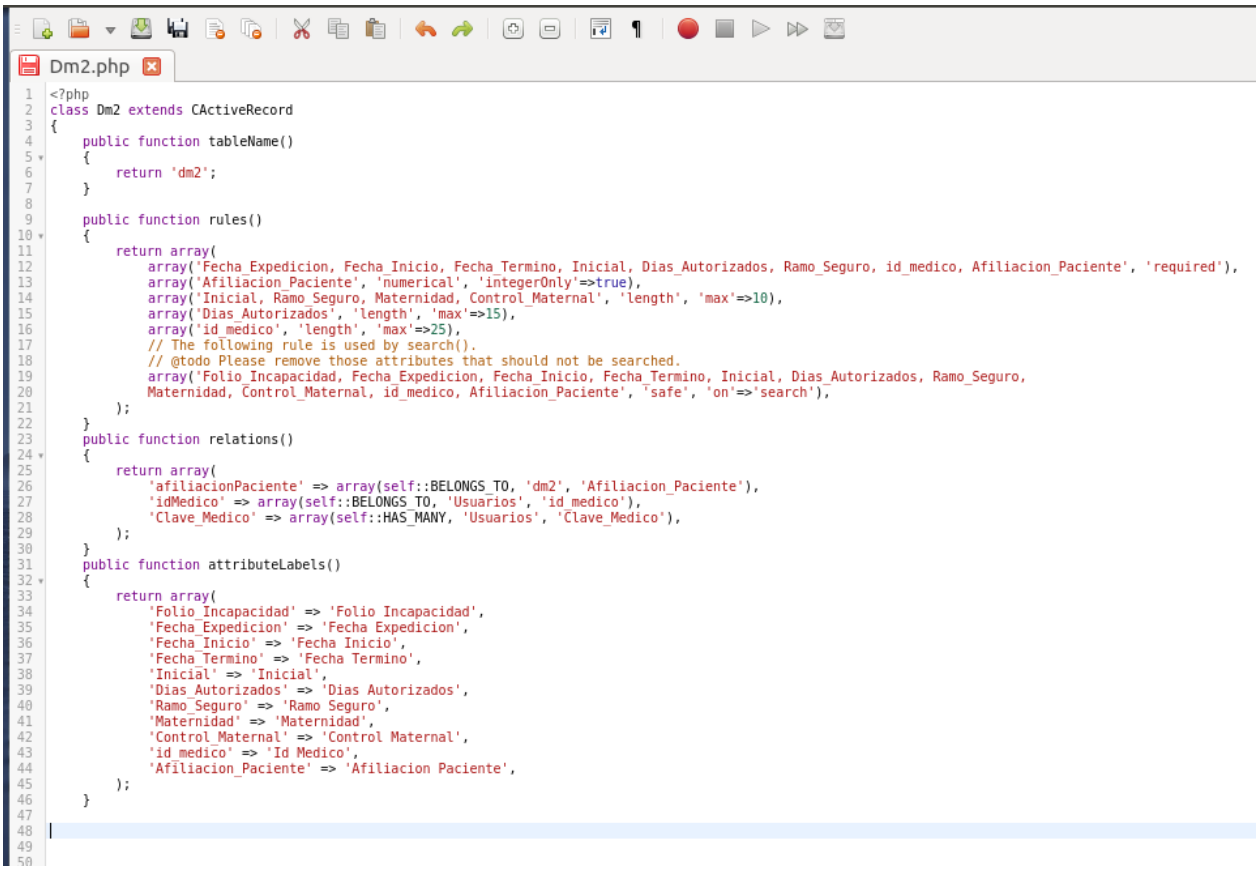


Figura 22. Código del Modelo de la Tabla dm2.

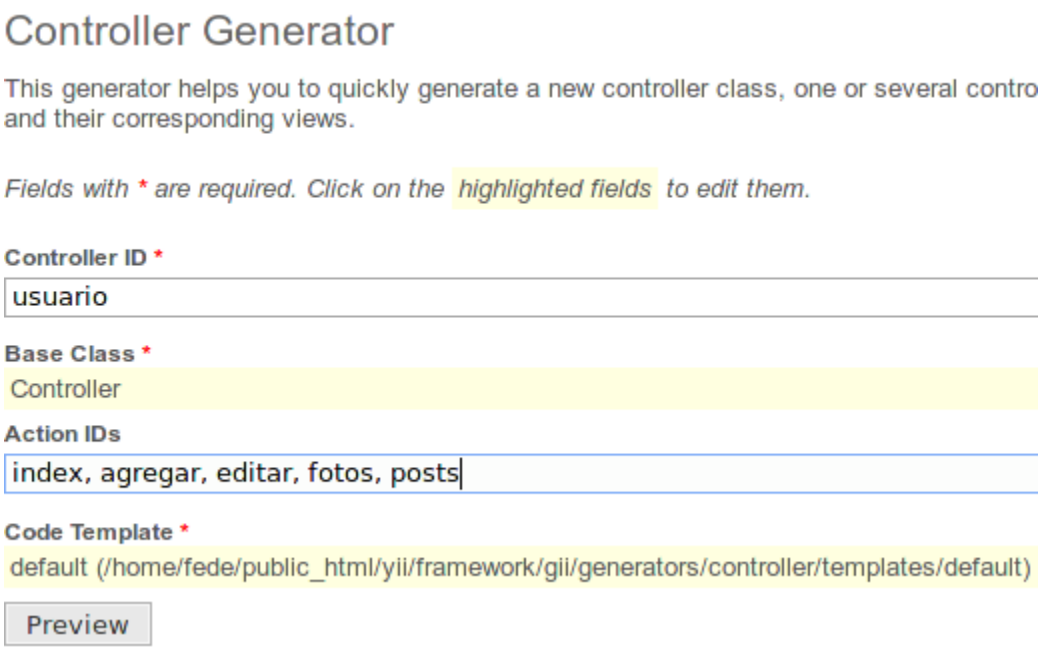
El código se divide en secciones las primeras líneas declaran las clases del modelo que ya en Capitulo 3 fueron mencionadas, las siguientes secciones contienen ciertas características que se refieren a los datos como lo es la máxima longitud de caracteres o el tipo de dato que le corresponde, por otra parte también se muestran la relaciones que existen con otras tablas como se puede observar en la Figura 20 (*function relations*), por la lógica del negocio para este proyecto, se deben de relacionar dos tablas la tabla médicos y la tabla derechohabientes, esto se hace a través de los atributos clave Clave_Medico y Afilacion_Paciente respectivamente.

Los Modelos restantes para las tablas Usuarios y dm3 son igualmente generados y conservan la misma estructura hecho por el cual son omitidos en el documento. Ahora daremos paso a los controladores de la aplicación.

El controlador que fue descrito en el Capitulo3, tiene como función la de controlar la comunicación entre los modelos y la vistas según la solicitud (requerimiento) que ha hecho un usuario.

Gii como se muestra en la *Figura 19* posee un menú lateral izquierdo el cual contiene las diferentes opciones de generación de código automático, será seleccionado el generador de controles ('Controller Generator'). Al entrar en esa opción solo vemos 2 campos. En, 'Controller ID' va el nombre del controlador y en 'Actions IDs' van las acciones que se requieren.

La pantalla que se nos muestra para recibir los parámetros necesarios la realización de un controlador es la siguiente:



Al darle click a Preview, todavía no genera nada de código. Solo se mostrara lo que se va a crear. Le daremos click al botón Generate esto creara todos los archivos, carpetas y código automáticamente.

Los controladores se encuentran en el directorio de archivos de la aplicación. El código está dentro de un archivo con el nombre ingresado en el campo de la imagen anterior (Controller ID) con extensión php. Se muestra a continuación las primeras 20 líneas de código para conocer el código completo *Véase Apéndice D* .

```php
<?php

class UsuariosController extends Controller
{
    /**
     * @var string the default layout for the views. Defaults to '//layouts/column2', meaning
     * using two-column layout. See 'protected/views/layouts/column2.php'.
     */
    public $layout='//layouts/column2';

    /**
     * @return array action filters
     */
    public function filters()
    {
        return array(
            'accessControl', // perform access control for CRUD operations
            'postOnly + delete', // we only allow deletion via POST request
        );
    }
```

Figura 23. Código del Controlador Usuarios.

Estas líneas de código son totalmente personalizables por lo cual es donde se definieron las reglas de acceso o permisos a cada usuario (accesRules) por ejemplo en este caso en particular los administradores pueden acceder a las tareas como creación, modificación y eliminación de los usuarios así como tener acceso a las vistas de esta página y el resto de los roles no lo podrán hacer.

Los Controladores restantes para los modelos dm2 y dm3 son igualmente generados y conservan la misma estructura hecho por el cual son omitidos en el documento. Ahora daremos paso a las vistas o interfaz de usuario de la aplicación.

Las vistas son la parte final de la implementación, estas fueron descritas en el Capítulo 3 y consisten básicamente en elementos de la interfaz de usuario (user interface - UI); es decir, el diseño que presentamos al usuario final, pueden contener en su mayoría expresiones PHP.

Muchos de los requerimientos por parte del usuario fueron agregados en las vistas lo que hace más agradable y entendible la aplicación.

El generador de código Gii a la par de la generación del controlador, se encarga de la parte de las vistas que este controlara, por lo que no es necesario hacer más pasos. En la *Figura 21* se muestran los archivos de tipo vista junto a los del controlador.

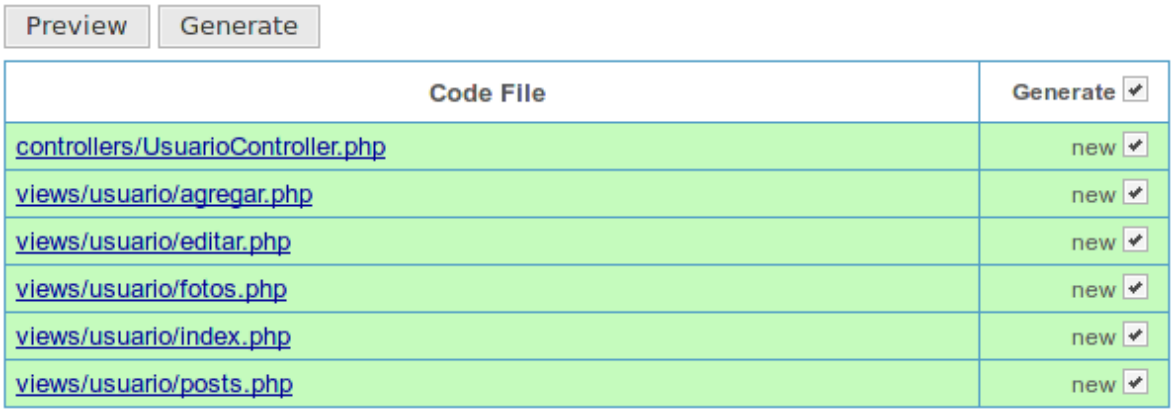


Figura 24. Archivos del Controlador y la Vista creados.

El marco de trabajo Yii y su herramienta Gii en específico permite que al igual que como se genera el modelo el controlador y las vistas se genere el CRUD de las tablas de nuestra aplicación a nivel de archivos, éste es el resultado de las vistas.

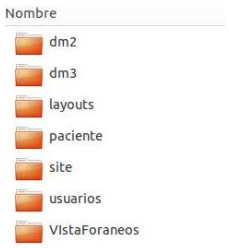


Figura 25. Carpetas de las Vistas generadas de la aplicación.

Ahora exploraremos el contenido de la primera las vistas correspondientes a la carpeta dm2, como se muestra en la figura 23 el primer archivo admin.php contiene una vista resumen, es decir no es más que una consulta a la base de datos con la información de los dm2 creados, el segundo archivo corresponde ya a el CRUD y es la vista que nos permite crear nuevos dm2, el tercer archivo _form es un archivo que guarda información de interfaz solamente, el cuarto archivo es la vista de la página principal, el quinto archivo

es una vista extra generar para la visualización de información en un formato especial como se vió en el Capítulo 3 en los Casos de usos, el sexto archivo es la vista de búsqueda, el séptimo archivo es otro elemento del CRUD el cual se encarga de dar la vista para modificar la información, el octavo archivo es la vista general donde se contempla otro elemento del CRUD la eliminación de dm2.

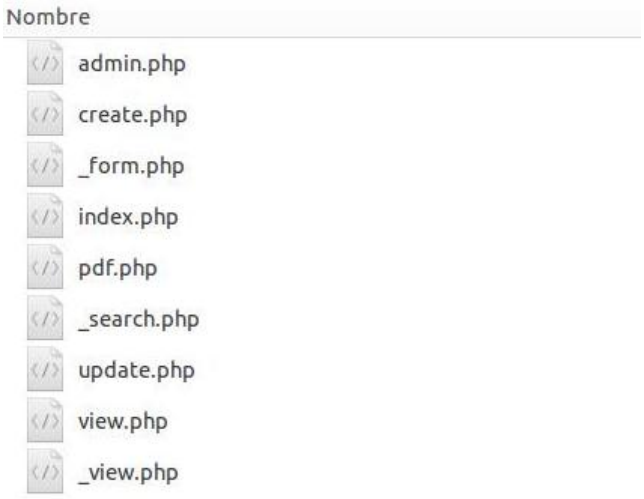


Figura 26. Archivos de vistas del dm2

Ahora se presentaran algunas páginas de usuario finales dentro del sistema.

Figura 27. Página de Inicio.

Figura 28. Página de Creación de un dm2

Figura 29. Página de Búsqueda de un dm2

4.6 Pruebas y Análisis de Operación.

En este Capítulo se describirán algunos aspectos sobre tiempos, mecanismos y procesos que se llevan a cabo para expedir y resguardar una Licencia Médica (DM-2) y un Diagnostico Medico(DM-3) en las Instituciones Foráneas del ISSSTEP.

El departamento de Información y Calidad del ISSSTEP, es el encargado de crear, modificar y documentar cualquier proceso dentro del instituto, un proceso manual lo conforman tres aspectos principales en una serie de pasos a seguir de principio a fin, el primer aspecto tiene que ver con los responsables de realizar actividades del proceso, el segundo tiene que ver con las actividades a realizar de principio a fin durante el proceso , el último de los aspectos tiene que ver con el o los resultados por lo general un documento asociado al proceso.(Véase Apéndice D).

4.6.1 Tiempo de expedición y análisis de DM-2 y DM-3

Medir el tiempo que tarda una tarea donde se recaban y procesan datos, es de suma importancia para poder comprobar y analizar la información, con ellos se podrán evidenciar las deficiencias y problemas que intervienen y así poder combatirlos. A continuación presentaremos una línea de tiempo que ejemplifica cuánto tiempo lleva realizar cada tarea.

De 15 a 20 Minutos	De 15 a 30 Días	De 1 a 5 Días	De 1	FIN DE DM-3
Expedición de DM-3	Tiempo de Correspondencia	Tiempo de Análisis y Codificación	Tiempo de Resguardo	

De 15 a 20 Minutos	De 15 a 30 Días	De 1 a 5 Días	De 1 Día 5 dias	FIN DE DM-2
Expedición de DM-2	Tiempo de Correspondencia	Tiempo de Análisis y Codificación	Tiempo de Resguardo	

4.6.2 Cifras de codificaciones realizadas a formatos DM-3 por mes y región.

El ISSSTEP recibe y procesa diariamente por medio de sus médicos una cantidad de formatos DM-3 numerosa, que se pudo constatar gracias al departamento de Calidad y Estadística de ENERO a NOVIEMBRE de 2013 con cifras reales, a continuación daremos a conocer esta información proporcionada por el ISSSTEP.

MES	REGION 1	REGION 2	REGION 3	REGION 4	REGION 5	REGION 6	REGION 7
ENERO	387	220	300	312	198	103	399
FEBRERO	421	106	269	396	232	100	368
MARZO	314	211	356	313	155	289	402
ABRIL	397	199	456	325	277	270	427
MAYO	415	203	299	313	233	246	400
JUNIO	385	280	366	331	266	108	379
JULIO	297	203	321	322	289	256	354
AGOSTO	301	196	422	333	271	96	312
SEPTIEMBRE	406	207	387	368	246	110	456
OCTUBRE	420	96	429	311	230	299	444
NOVIEMBRE	396	165	463	396	279	141	402
TOTAL	**4139**	**2086**	**4068**	**3720**	**2676**	**4139**	**2086**
							16,689

Tabla 7. Tabla de cifras del DM-3 durante 11 meses.

Como se puede ver, la tabla refleja que en poco menos de un año el número de codificaciones es grande e imposible humanamente hablando de trascribir y procesar por una sola persona toda la información pero sorprendentemente si lo es para una sola computadora, la cual es capaz de guardar los datos de manera rápida y confiable. Cambio que se ve reflejado y comprobado en cada una de las unidades donde se realización pruebas y fueron exitosas.

CAPITULO 5.

Conclusión.

Este proyecto de tesis tuvo como finalidad atender dos de las principales problemáticas presentes en las instituciones foráneas adscritas al ISSSTEP. La primera problemática versaba en la falsificación de cierta información, y la segunda, en el tiempo tan prolongado para la trasmisión y resguardo de los datos de los derechohabientes.

Por lo que este proyecto se basó en la automatización de los pasos que componen el registro y resguardo de la información de los derechohabientes, permitiendo que actualmente, funcione de la siguiente manera:

Primero se elige y captura los principales identificadores del derechohabiente. Posteriormente se crea la información y su vez, se trasmite a través de la red, a la institución sede en la capital del Estado para su resguardo y posterior análisis. Todo esto permite, que tan solo en unos minutos el derechohabiente obtenga el trámite que requiere de manera eficaz y al mismo tiempo la institución corrobore la información que se proporciona.

Referente los objetivos propuestos en el presente trabajo, se concluye lo siguiente:

Respecto a Desarrollar una plataforma Web que permitiera el registro de información en las Unidades Foráneas por medio de una red de Internet, misma que debería interactuar con los sistemas existentes para consulta del padrón de médicos y derechohabientes; se confirmó que efectivamente una red de bajo desempeño en la unidad médica, como es un ancho de banda de 2Mb, fue suficiente para que con la tecnología elegida, php con el marco de trabajo Yii y servidor Web Apache, se lograra de forma exitosa la transmisión correcta de la información tanto para registro como consulta. Misma que permite retroalimentar los sistemas actuales del hospital.

En lo referente a Implementar un acceso por roles, restringiendo la consulta y registro de información según la responsabilidad de cada usuario; se pudo comprobar la integridad y seguridad de la información, misma que de acuerdo a la Ley de Protección de Datos Personales en posesión de particulares, corresponden a datos sensibles por referirse a

salud. Motivo por el cual la decisión del manejo de roles le da la seguridad requerida al sistema.

En general, el objetivo de Automatizar la administración y análisis de información referente a diagnósticos y licencias médicas en las Unidades Médicas Foráneas del ISSSTEP; fue posible a través de la solución planteada en el capítulo 3, diseño, así como implementada conforme a lo descrito en el capítulo 4. Demostrando que la hipótesis general de investigación:

Una aplicación Web permitirá la correcta administración de la información generada en la expedición de licencias médicas y el registro de diagnósticos médicos para las unidades foráneas del ISSSTEP.

Está demostrada y evaluada conforme se concluye en los párrafos anteriores.

Asimismo la presente investigación dio origen a nuevos problemas a resolver, mismos que salen del ámbito de este trabajo y se dejan como trabajo a futuro, estas nuevas líneas se mencionan a continuación:

- Tablero de Control a nivel Administrativo para la toma de decisiones, el cual ayudara a los epidemiólogos y expertos en la materia de salud obtener información del comportamiento de las enfermedades en cada región con ayuda de la información de cada uno de los formatos automatizados. Para lo cual se propone utilizar herramientas como Spago BI.
- Codificación automática en los diagnósticos médicos (DM-3), con el análisis del texto de cada diagnóstico escrito por el médico, implementar un algoritmo capaz de obtener de manera automática las claves o códigos correspondientes a cada padecimiento. En otras palabras aplicar procesamiento automático de lenguaje de textos en los diagnósticos.
- Hacer una interfaz nueva con la capacidad de almacenar localmente la información en caso de pérdida total de conectividad y que ésta sea enviada una vez restablecida la comunicación, ésta deberá ser compatible con la ya implementada para los lugares donde el internet es muy inestable.

Bibliografía.

Yii Software. (28 de 12 de 2008). *yiiframework*. Recuperado el 20 de Enero de 2015, de yiiframework: http://www.yiiframework.com/doc/guide/1.1/es/basics.mvc

ACUNETIX. (18 de FEBRERO de 2012). *acunetix*. Obtenido de http://www.acunetix.com/

Aghazadeh, S., Aliyev, A., & Ebrahimnezhad, M. (2012). Review the Role of Hospital Information Systems in Medical Services Development. *Internation Journal of Computer Theory and Engineering*, 5.

Castela Garcia, L. (2012). *Diseño de un Almacen de Datos basado en Data Warehouse Engineer Process.* Veracruz, Mexico: Universidad Veracruzana.

De Troyer, O. (1998). WSDM: a user centered design method for Web sites. *Computer Networks and ISDN Systems*, 85-94.

Diaz Novás , J., Gallego Machado, B., & León González, A. (2006). El diagnóstico médico:bases y procedimientos. *Medicina General Integral*, 22.

Díaz Vélez, C., & Peralta Lui, F. (2013). Calidad de los certificados de incapacidad temporal para el trabajo emitidos en un hospital EsSalud. *Cuerpo Médico*, 46-49.

Escalona Cuaresma, M. J., & González Romano, J. M. (1 de Agosto de 2007). *Universidad de Sevilla.* Recuperado el 29 de Septiembre de 2014, de http://www.lsi.us.es/docencia/master-oficial.php

Escalona, M. J., & Koch, N. (2002). *Ingenieria de Requsitos en Aplicaciones para el Web- Un estudio comparativo.* Sevilla, Espana: Escuela Tecnica Superior de Ingenieria Informatica.

Gamma, Helm, Johnson, & Vlissides. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software.* USA: Addison-Wesley Professional.

García García, J. J. (2010). *Estrategias para elaborar un diagnóstico clínico.* Mexico: Facultad de Mecina UNAM.

Giraldo G, G. L. (2009). *Tableros de control para un proceso del negocio de una compañia de seguros.* Medellin: Universidad Nacional de Colombia.

Guerrero Martínez, J. F. (2011). *Informatíca Médica y Telemedicina.* Valencia: Escuela Tecnica Superior de Ingenieria.

Heeseok, L., Choongseok, L., & Cheonsoo, Y. (1999). *A scenario based object oriented hypermedia desing methodology.* Seoul South Korea: ELSEVIER.

Hennicker, R., & Koch, N. (2000). *A UML-based Methodology for Hypermedia Design.* München, Alemania: Institute of Computer Science.

Kurtz, L. (30 de Abril de 2012). *Introduction to Open Source Web Frameworks*. Recuperado el 14 de Septiembre de 2014, de http://thejakegroup.com/blog/introduction-to-open-source-web-frameworks/

La Araucana, L. (Enero de 2010). *La Araucana*. Obtenido de La Araucana: http://www.laaraucana.cl/irj/portal/anonymous?NavigationTarget=navurl://7531003a2ea69250cbcb63b13585aad8

Leninmhs. (9 de Abril de 2013). *Leninmhs*. Recuperado el 1 de SEPTIEMBRE de 2014, de http://leninmhs.wordpress.com/

LLC., Y. S. (28 de 12 de 2008). *yiiframework*. Recuperado el 06 de ENERO de 2015, de yiiframework: http://www.yiiframework.com/doc/guide/1.1/es/basics.view

MÉXICO, H. G. (25 de Junio de 2014). *Secretaria de Salud Hospital General de México*. Recuperado el 02 de septiembre de 2014, de http://www.hgm.salud.gob.mx/interior/area_medica/consul_exter/inicio.html

Naciones, D. (2012). *Que es una aplicacion Web?* Nueva York E.U: About.com (una compañía IAC).

Oracle. (marzo de 3 de 2008). *oracle*. Recuperado el 08 de Junio de 2014, de http://www.oracle.com/technetwork/es/oem/tutorial-de-oracle-bpm-1704890-esa.pdf

Pacheco, J. E. (2 de Junio de 2013). *Instituto Tecnologico de Veracruz*. Obtenido de http://prograweb.com.mx/pweb/0205metodologiasWeb.html

Palma Valdés, F. F. (2013). *INFORME DE LABORES 2013 .* Puebla: ISSSTEP.

Pérez García, A. A. (2007). *Desarrollo de herramientas web de gestión docente.* CARTAGENA: UNIVERSIDAD POLITÉCNICA DE CARTAGENA.

Pluss, R. (25 de Enero de 2005). *Analisis Corporativo de LAMP*. Recuperado el 12 de SEPTIEMBRE de 2014, de http://conocimientolibre.wordpress.com/

Rivera López, A. (2008). *Sistema asistente para la generación de horarios de cursos.* Puebla: Universidad de las Américas Puebla.

Rodríguez, A. N. (2009). *Metodologías de diseno usadas en ingeniería Web, su vinculación con las NTICS.* Buenos Aires,Argentina: Universidad Nacional de la Plata.

S, A., A, A., & M, E. (Diciembre de 2012). Review the Role of Hospital Information Systems in. *International Journal of Computer Theory and Engineering*, 55.

Schmuller, J. (1997). *Aprendiendo UML en 24 Horas.* Naucalpan de Juárez,Edo de Mexico: Pearson Educación Latinoamérica.

Silva, D. A., & Mercerat, B. (2002). Construyendo aplicaciones Web con una metodologia de diseño orientada a objetos. *Unab*, 20.

V, D. (27 de 05 de 2010). *GeekGumbo*. Recuperado el 01 de 03 de 2015, de http://www.geekgumbo.com/2010/05/27/php-frameworks-an-introduction/

Winesett, J. (2012). *Web Application Development with Yii and PHP.* Birmingham: Packt.

YII. (12 de Junio de 2009). *yiiframework*. Recuperado el 1 de Octubre de 2014, de http://www.yiiframework.com/doc/guide/1.1/es/basics.controller

Apéndice.

APENDICE A.
FIGURAS DE DIAGRAMAS DE ACTIVIDADES.

DIAGRAMA DE ACTIVIDADES UWE.

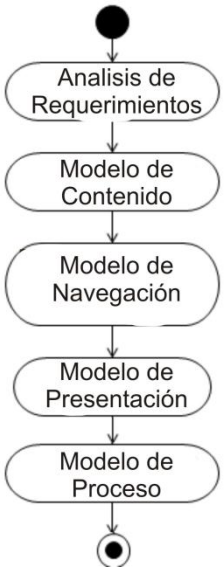


Fig. 1 Diagrama de actividades UWE. (Escalona Cuaresma & González Romano, 2007)

DIAGRAMA DE ACTIVIDADES WSDM.

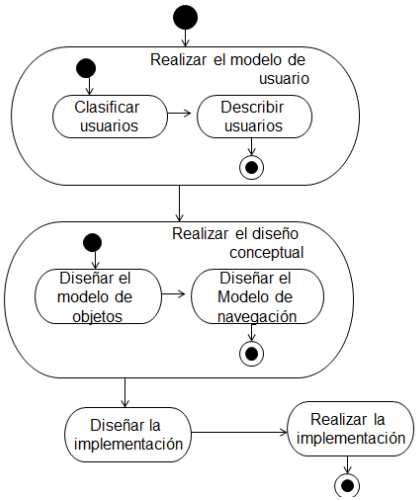


Fig. 2 Diagrama de actividades WSDM (Escalona Cuaresma & González Romano, 2007)

DIAGRAMA DE ACTIVIDADES SOHDM.

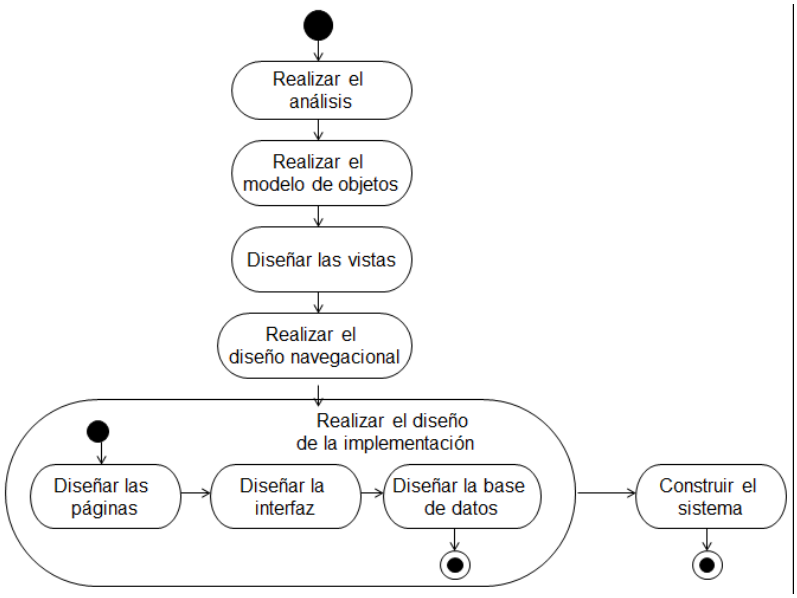


Fig.3 Diagrama de actividades SOHDM. (Escalona Cuaresma & González Romano, 2007)

APENDICE B.
ESTADISTICAS DEL MARCO DE TRABAJO Yii.

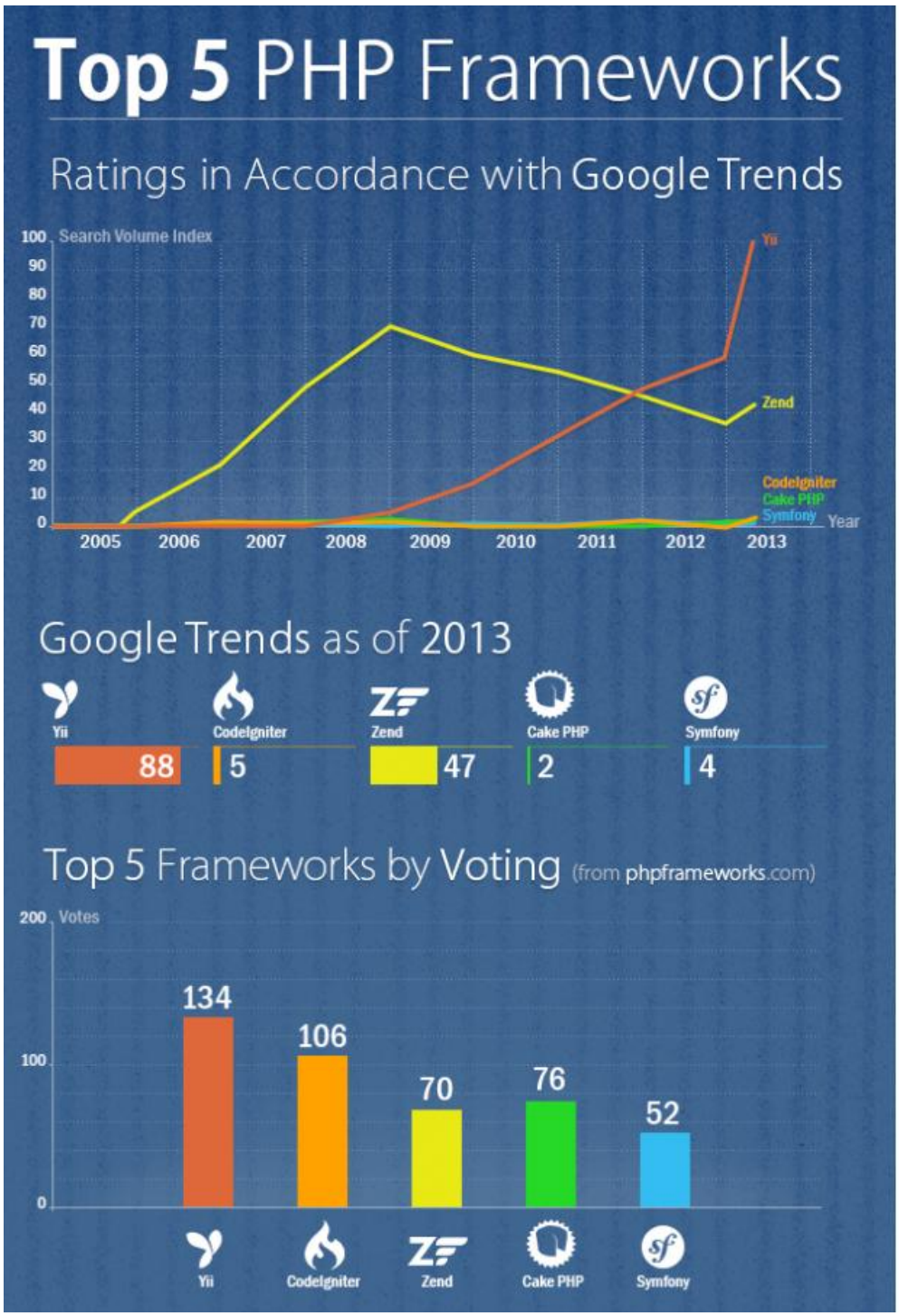


Fig.4 Grafica comparativa de Yii.

Web: http://brandongaille.com/

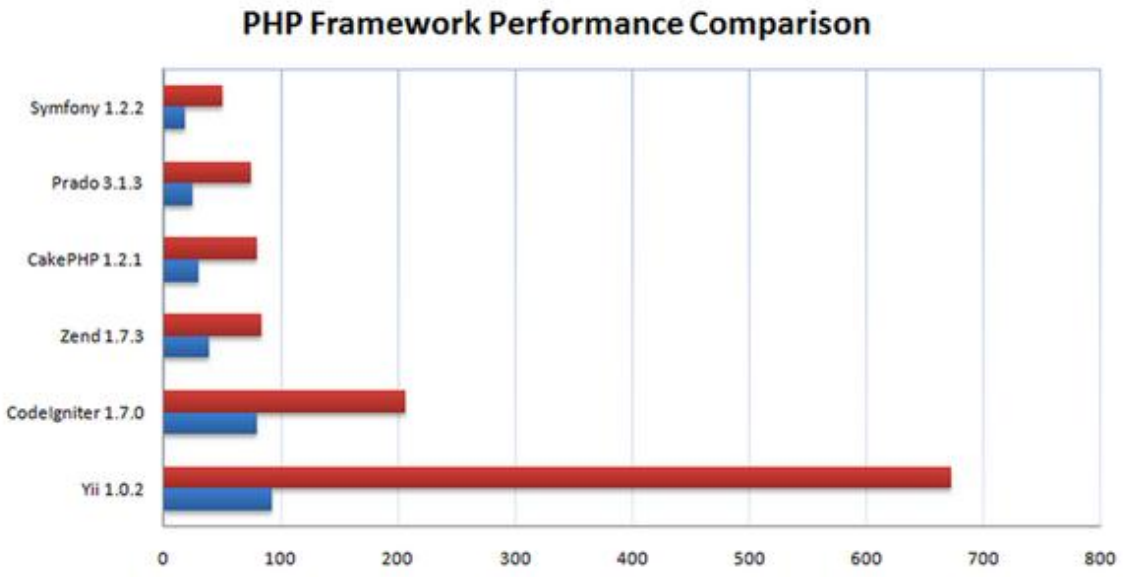


Fig.5 Grafica comparativa de Yii.

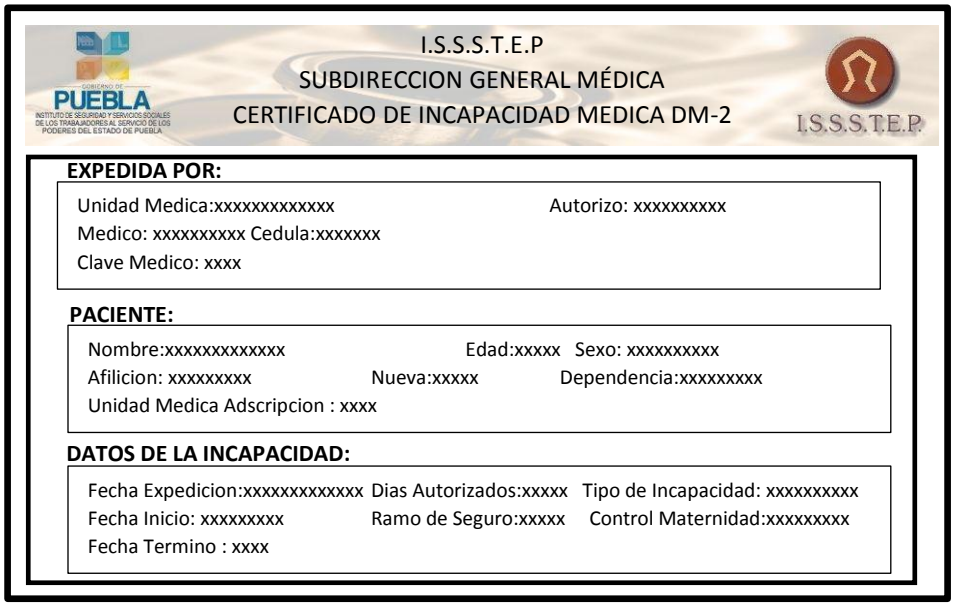
Web: http://code.tutsplus.com/

PHP Framework	PHP4	PHP5	MVC	Multiple DB's	ORM	DB Objects	Templates	Caching	Validation	Ajax	Auth Module	Modules	EDP
Akelos	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
ash.MVC	-	✓	✓	-	-	✓	✓	-	✓	-	✓	✓	-
CakePHP	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-
CodeIgniter	✓	✓	✓	✓	-	✓	✓	✓	✓	-	-	-	-
DIY	-	✓	✓	-	✓	✓	✓	✓	-	✓	-	-	-
eZ Components	-	✓	-	✓	-	✓	✓	✓	✓	-	-	-	-
Fusebox	✓	✓	✓	✓	-	-	-	✓	-	✓	-	✓	-
PHP on TRAX	-	✓	✓	✓	✓	✓	-	-	✓	✓	-	✓	-
PHPDevShell	-	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	-
PhpOpenbiz	-	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	-	-
Prado	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
QPHP	✓	✓	✓	✓	-	✓	✓	-	✓	✓	✓	✓	✓
Seagull	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
Symfony	-	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-
WACT	✓	✓	✓	✓	-	✓	✓	-	✓	-	-	✓	-
WASP	-	✓	✓	-	-	✓	✓	-	✓	✓	✓	✓	-
Yii	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Zend	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
ZooP	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	-	-

Fig.6 Grafica comparativa de aspectos importantes en marcos de trabajo.

Web: http://www.phpframeworks.com/

APENDICE C.
FORMATOS ISSSTEP DM-2 Y DM-3.


I.S.S.S.T.E.P
SUBDIRECCION GENERAL MÉDICA
CERTIFICADO DE INCAPACIDAD MEDICA DM-2

EXPEDIDA POR:

Unidad Medica:xxxxxxxxxxxxx Autorizo: xxxxxxxxxx
Medico: xxxxxxxxxx Cedula:xxxxxxx
Clave Medico: xxxx

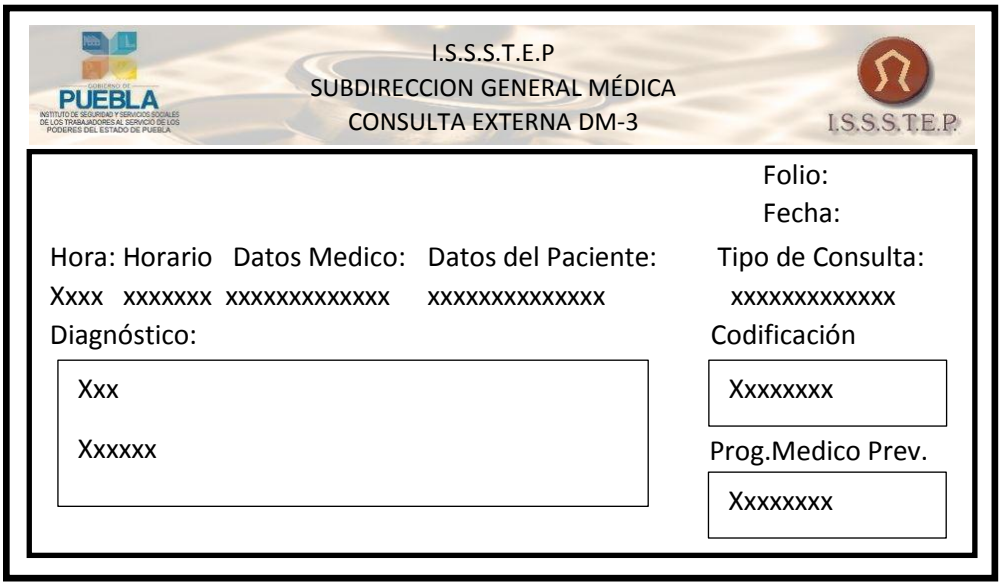
PACIENTE:

Nombre:xxxxxxxxxxxxx Edad:xxxxx Sexo: xxxxxxxxxx
Afilicion: xxxxxxxxx Nueva:xxxxx Dependencia:xxxxxxxxx
Unidad Medica Adscripcion : xxxx

DATOS DE LA INCAPACIDAD:

Fecha Expedicion:xxxxxxxxxxxx Dias Autorizados:xxxxx Tipo de Incapacidad: xxxxxxxxxx
Fecha Inicio: xxxxxxxxx Ramo de Seguro:xxxxx Control Maternidad:xxxxxxxxx
Fecha Termino : xxxx

Fig.7 Formato DM-2


I.S.S.S.T.E.P
SUBDIRECCION GENERAL MÉDICA
CONSULTA EXTERNA DM-3

Folio:
Fecha:

Hora: Horario Datos Medico: Datos del Paciente: Tipo de Consulta:
Xxxx xxxxxxx xxxxxxxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxxxxx
Diagnóstico: Codificación

Xxx
Xxxxxx

Xxxxxxxx

Prog.Medico Prev.

Xxxxxxxx

Fig.8 Formato DM-3

APENDICE D.
CODIGO GENERADO PARA EL CONTROLADOR.

```php
<?php

class UsuariosController extends Controller
{
    /**
     * @var string the default layout for the views. Defaults to '//layouts/column2', meaning
     * using two-column layout. See 'protected/views/layouts/column2.php'.
     */
    public $layout='//layouts/column2';

    /**
     * @return array action filters
     */
    public function filters()
    {
        return array(
            'accessControl', // perform access control for CRUD operations
            'postOnly + delete', // we only allow deletion via POST request
        );
    }

    /**
     * Specifies the access control rules.
     * This method is used by the 'accessControl' filter.
     * @return array access control rules
     */
    public function accessRules()
    {
        return array(
            array('allow',  // allow all users to perform 'index' and 'view' actions
                'actions'=>array('index','view'),
                'users'=>array('admin'),
            ),
            array('allow', // allow authenticated user to perform 'create' and 'update' actions
                'actions'=>array('create','update'),
```

```
                'users'=>array('admin'),
            ),
            array('allow', // allow admin user to perform 'admin' and 'delete' actions
                'actions'=>array('admin','delete'),
                'users'=>array('admin'),
            ),
            array('deny',  // deny all users
                'users'=>array('*'),
            ),
        );
    }

    /**
     * Displays a particular model.
     * @param integer $id the ID of the model to be displayed
     */
    public function actionView($id)
    {
        $this->render('view',array(
            'model'=>$this->loadModel($id),
        ));
    }

    /**
     * Creates a new model.
     * If creation is successful, the browser will be redirected to the 'view' page.
     */
    public function actionCreate()
    {
        $model=new Usuarios;

        // Uncomment the following line if AJAX validation is needed
        // $this->performAjaxValidation($model);

        if(isset($_POST['Usuarios']))
        {
            $model->attributes=$_POST['Usuarios'];
            $model->password=md5($model->password);
            if($model->save())
                $this->redirect(array('view','id'=>$model->Clave_Medico));
        }

        $this->render('create',array(
            'model'=>$model,
        ));
    }

    /**
     * Updates a particular model.
     * If update is successful, the browser will be redirected to the 'view' page.
     * @param integer $id the ID of the model to be updated
     */
```

```php
    public function actionUpdate($id)
    {
        $model=$this->loadModel($id);

        // Uncomment the following line if AJAX validation is needed
        // $this->performAjaxValidation($model);

        if(isset($_POST['Usuarios']))
        {
            $model->attributes=$_POST['Usuarios'];
            $model->password=md5($model->password);
            if($model->save())
                $this->redirect(array('view','id'=>$model->Clave_Medico));
        }

        $this->render('update',array(
            'model'=>$model,
        ));
    }

    /**
     * Deletes a particular model.
     * If deletion is successful, the browser will be redirected to the 'admin' page.
     * @param integer $id the ID of the model to be deleted
     */
    public function actionDelete($id)
    {
        $this->loadModel($id)->delete();

        // if AJAX request (triggered by deletion via admin grid view), we should not redirect the browser
        if(!isset($_GET['ajax']))
            $this->redirect(isset($_POST['returnUrl']) ? $_POST['returnUrl'] : array('admin'));
    }

    /**
     * Lists all models.
     */
    public function actionIndex()
    {
        $dataProvider=new CActiveDataProvider('Usuarios');
        $this->render('index',array(
            'dataProvider'=>$dataProvider,
        ));
    }

    /**
     * Manages all models.
     */
    public function actionAdmin()
    {
        $model=new Usuarios('search');
        $model->unsetAttributes();  // clear any default values
        if(isset($_GET['Usuarios']))
            $model->attributes=$_GET['Usuarios'];

```

```
143             $this->render('admin',array(
144                 'model'=>$model,
145             ));
146         }
147 
148         /**
149          * Returns the data model based on the primary key given in the GET variable.
150          * If the data model is not found, an HTTP exception will be raised.
151          * @param integer $id the ID of the model to be loaded
152          * @return Usuarios the loaded model
153          * @throws CHttpException
154          */
155         public function loadModel($id)
156         {
157             $model=Usuarios::model()->findByPk($id);
158             if($model===null)
159                 throw new CHttpException(404,'The requested page does not exist.');
160             return $model;
161         }
162 
163         /**
164          * Performs the AJAX validation.
165          * @param Usuarios $model the model to be validated
166          */
167         protected function performAjaxValidation($model)
168         {
169             if(isset($_POST['ajax']) && $_POST['ajax']==='usuarios-form')
170             {
171                 echo CActiveForm::validate($model);
172                 Yii::app()->end();
173             }
174         }
175 }
176 
```

APENDICE E.
PROCEDIMIENTO DE CODIFICACIÓN ISSSTEP.

Procedimiento para la codificación de diagnósticos de la consulta por médico y especialidad. (Proceso Manual).

El ISSSTEP cuenta con procedimientos certificados a nivel estatal expondremos a continuación los pasos a seguir para la codificación de diagnósticos de una consulta médica.

PROCEDIMIENTO MANUAL PARA UNA CONSULTA EXTERNA(DM-3)

Normar los procedimientos para la atención médica, odontológica, psicológica, preventiva y curativa con un enfoque integral, que propicie la modificación del estado de enfermedad del (la) paciente atendido en el servicio de consulta externa en las unidades médicas del primer nivel de atención del Instituto.

Responsable	Descripción de Actividades	Formato y/o Documento
Inicia 1. Derechohabiente.	Acude al módulo de control de consulta externa de su Unidad a solicitar consulta en cualquiera de los servicios disponibles y acredita su vigencia de derechos mediante la presentación de los documentos correspondientes.	-Credencial vigente (DP-5) o en su caso Credencial provisional (DP-08). -Ultimo Talón de Pago (DA-8) -Tarjeta de Citas (DM-10).
2. Personal Responsable del Control de Citas (Asistente de Consultorio, Asistente Medica, Auxiliar Polivalente, Trabajador Social, Enfermera).	Recibe al derechohabiente, verifica vigencia de derechos, asigna horario de consulta, solicita y/o busca Expediente Clínico, registra el nombre del derechohabiente, transcribe a mano en el formato de Control de Consulta Externa y turna al Médico Tratante.	Expediente Clínico Control de Consulta Externa (DM-3).

3. Médico Tratante	a) Recibe Control de Consulta y Expediente Clínico físico o accede al Expediente Clínico Electrónico. b) Llama al paciente, lo interroga, efectúa exploración física, establece diagnóstico presuntivo o definitivo, realiza los procedimientos clínicos y terapéuticos necesarios, asienta la información en el Expediente Clínico.	Expediente Clínico Control de Consulta Externa (DM-3).
4. Derechohabiente.	Recibe Receta Médica que surte en Farmacia. ¿Amerita Licencia Temporal para el trabajo? Sí, continúa en actividad No.5 No, fin del procedimiento.	
5. Médico Tratante.	a) Elabora de manera transcrita el Certificado de Incapacidad Temporal de Trabajo, en el formato foliado de Certificado en original y dos copias.	Certificado de Incapacidad Temporal para el Trabajo (DM-2).
6. Personal Responsable	Envía los Certificados de Incapacidad Temporal de Trabajo realizados (… periodo de tiempo) a el ISSSTEP unidad Puebla Centro.	Paquete de Correspondencia.
7. DEPARTAMENTO DE CALIDAD(UNIDAD QUE LOS RESIVE…).	Recibe los formatos foliados de cada una de las unidades médicas foráneas.	Certificado de Incapacidad Temporal para el Trabajo DM-2. Expediente Clínico Control de Consulta Externa. DM-3.
8. Capturistas.	Realizan la codificación de los	Certificado de Incapacidad

9. Personal del Departamento	diagnósticos médicos apegados a la nomenclatura de la norma mundial de la salud. Archiva y resguarda los formatos.	Temporal para el Trabajo DM-2. Expediente Clínico Control de Consulta Externa DM-3. Cajas Foliadas y etiquetas.
…CONTINUA	*…CONTINUA*	*…CONTINUA*