PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS

"MIGRACIÓN DEL SISTEMA DE FICHAS MÉDICAS DEL HOGAR CORAZÓN DE MARÍA DE JBOSS Y JAVA SOBRE WINDOWS A APAHCE Y PHP SOBRE LINUX"

ROY JAVIER SARANGO LOAIZA

QUITO, 2011

CERTIFICADO DEL DIRECTOR

A mis padres

AGRADECIMIENTOS

En especial al Máster Eddy Sánchez, por su valiosa contribución en la dirección del presente trabajo de investigación.

A la Madre Benita Mazón, Directora del "Hogar Corazón de María", quien me permitió realizar el proyecto de tesis en el Centro Gerontológico que acertadamente dirige.

A todas las personas que de forma directa o indirecta colaboraron en la realización, tanto en la parte práctica, como en la parte escrita de esta tesis.

TABLA DE CONTENIDOS

CERTIFICADO DEL DIRECTOR II

AGRADECIMIENTOS IV

RESUMEN 1

INTRODUCCIÓN 2

Objetivo General 2

Objetivos Específicos 2

1 SITUACIÓN TECNOLÓGICA ACTUAL DEL HOGAR CORAZÓN DE MARÍA 3

1.1 Características del Sistema Anterior 4
1.1.1 Administración 4
1.1.1.1 Opciones 5
1.1.1.2 Permisos 6
1.1.1.3 Roles 7
1.1.1.4 Usuarios 8
1.1.2 Configuración 8
1.1.3 Fichas Médicas 9
1.1.3.1 Datos Generales 9
1.1.3.2 Diagnóstico 10
1.1.3.3 Contacto 11
1.1.3.4 Estructura Familiar 12
1.1.3.5 Motivo de Consulta y Enfermedad Actual 12
1.1.3.6 Antecedentes Patológicos Personales 12
1.1.3.7 Hábitos 13
1.1.3.8 Historia Gineco Obstetra 14
1.1.3.9 Antecedentes Patológicos Familiares 15
1.1.3.10 Revisión de Sistemas 16
1.1.3.11 Examen Físico 16
1.1.3.12 Impresión Diagnóstica 17
1.1.3.13 Recomendaciones y Tratamientos 18
1.1.3.14 Parte Diario 18
1.1.3.15 Evolución y Prescripciones 18
1.1.3.16 Terapia Física 19
1.1.3.17 Análisis Psicológico 19
1.1.3.18 Cortes Vertical y Horizontal 20

1.2 Performance del Sistema Anterior 20
1.2.1 Comparación de Lenguajes 20
1.2.2 Infraestructura Tecnológica del Hogar 22
1.2.3 Rendimiento del Sistema 23

1.3 Nuevos Requerimientos 23
1.3.1 Requerimientos del MIES 23
1.3.2 Requerimientos del Hogar 24
1.3.3 Modelo de Requerimientos 25

2 MODELO ENTIDAD RELACIÓN 27

2.1 Obtención del MER del Sistema Anterior 27

2.2 Estandarización de Objetos de Base de Datos 31
2.2.1 Tipo de Tablas 32
2.2.2 Tablas 33
2.2.3 Campos 33
2.2.4 Otros Objetos 34

2.3 Modificación del MER Anterior y Generación del Nuevo 36
2.3.1 Modificación del MER Anterior 36
2.3.2 Generación del Nuevo MER 38
2.3.2.1 Diccionario de Datos 41

3 DESARROLLO 49

3.1 Ciclo de Vida 49

3.2 Aspectos Introductorios del Proceso de Migración 51

3.3 Modelo Vista Controlador y Framework 53

3.4 Preparación del Ambiente de Desarrollo 55

3.5 Generación de Modelos 60

3.6 Generación de Controladores y Vistas 63

3.7 Características Adicionales de Yii 67

3.8 Desarrollo del Módulo de Configuración 68
3.8.1 Adopción de SRBAC como Módulo de Administración de Usuarios 71
3.8.1.1 Configuración de SRBAC 73
3.8.2 Casos de Uso 75

3.9 Desarrollo del Módulo de Fichas Médicas 77
3.9.1 Distribución de la Información 77
3.9.2 Contactos 80

3.9.3 Egresos y Reingresos 81
3.9.4 Historial de Cambios 82
3.9.5 Personalización de Búsqueda de Pacientes 84
3.9.6 Casos de Uso 86

4 LIBERACIÓN Y CAPACITACIÓN 88

4.1 Carga Inicial de Información 88

4.2 Contratación de Hosting y Dominio 89

4.3 Liberación al Ambiente de Producción 90

4.4 Capacitación 92

5 CONCLUSIONES Y RECOMENDACIONES 94

5.1 Conclusiones 94

5.2 Recomendaciones 95

BIBLIOGRAFÍA Y REFERENCIAS 97

Documentos Impresos 97

Páginas Web 97

GLOSARIO 100

ANEXOS 105

ÍNDICE DE TABLAS

Tabla 1-1 Datos Generales [A] 10
Tabla 1-2 Diagnóstico [A] 11
Tabla 1-3 Contacto [A] 11
Tabla 1-4 Estructura Familiar [A] 12
Tabla 1-5 Antecedentes Patológicos Personales [A] 13
Tabla 1-6 Hábitos [A] 14
Tabla 1-7 Historia Gineco Obstetra [A] 15
Tabla 1-8 Antecedentes Patológicos Familiares [A] 16
Tabla 1-9 Examen Físico [A] 17
Tabla 1-10 Impresión Diagnóstica [A] 17
Tabla 1-11 Parte Diario [A] 18
Tabla 1-12 Evolución y Prescripciones [A] 18
Tabla 1-13 Terapia Física [A] 19
Tabla 1-14 Análisis Psicológico [A] 19
Tabla 1-15 Corte Vertical y Horizontal [A] 20
Tabla 1-16 Infraestructura Tecnológica [B] 22
Tabla 2-1 Estándares de Restricciones [B] 35
Tabla 2-2 Diccionario de Datos del Sistema Migrado (Tablas) [B] 47
Tabla 3-1 Modelo Vista Controlador [B] 53
Tabla 3-2 ‘Opciones Básicas’ de cada Vista [B] 68
Tabla 4-1 Sesiones de Capacitación [B] 93

ÍNDICE DE ILUSTRACIONES

Ilustración 1-1 Categorización de Requerimientos [B] 25
Ilustración 1-2 Listado Detallado de Requerimientos [B] 26
Ilustración 2-1 Listado de Tablas del Sistema Anterior [C] 29
Ilustración 2-2 MER del Sistema Anterior [C] 30
Ilustración 2-3 Listado de Tablas del Sistema Migrado [B] 39
Ilustración 2-4 MER del Sistema Migrado [B] 40
Ilustración 3-1 Diagrama de Secuencia de MVC [B] 54
Ilustración 3-2 Directorio Raíz de Apache [B] 56
Ilustración 3-3 Configuración de Acceso a la Base de Datos [B] 57
Ilustración 3-4 Habilitación de Gii [B] 58
Ilustración 3-5 Datos Generales de la Aplicación [B] 58
Ilustración 3-6 Estandarización de URLs [B] 59
Ilustración 3-7 Aplicación Web Generada [B] 59
Ilustración 3-8 Generación Automática de un Modelo de MVC en Yii [B] 61
Ilustración 3-9 Diagrama de Clases de Modelos de MVC [B] 62
Ilustración 3-10 Generación Automática de CRUDs en Yii [B] 65
Ilustración 3-11 Diagrama de Clases de Controladores de MVC [B] 66
Ilustración 3-12 Pantalla de Creación de Pacientes [B] 67
Ilustración 3-13 Bredcrumbs [B] 68
Ilustración 3-14 ‘Opciones Básicas’ [B] 68
Ilustración 3-15 Despliegue de Opciones de la Barra de Menú [B] 69
Ilustración 3-16 SRBAC, Gestión de Permisos [B] 72
Ilustración 3-17 SRBAC, Usuarios y Roles Asignados vs. Roles Disponibles [B] 73
Ilustración 3-18 Configuración de SRBAC [B] 74
Ilustración 3-19 Importación de la Clase SBaseController [B] 75
Ilustración 3-20 Casos de Uso, Módulo de Configuración [B] 76
Ilustración 3-21 Datos Generales del Paciente [B] 78
Ilustración 3-22 Consulta [B] 79
Ilustración 3-23 Contacto de Paciente [B] 81
Ilustración 3-24 Reingreso de Paciente [B] 82
Ilustración 3-25 Registro de Modificación de Paciente [B] 83

Ilustración 3-26 Búsqueda de Pacientes [B]..... 85
Ilustración 3-27 Casos de Uso, Módulo de Fichas Médicas [B] 87

RESUMEN

La presente disertación explica detalladamente el proceso de migración del sistema de fichas médicas desarrollado para el centro gerontológico "Hogar Corazón de María". La modificación de tecnologías usadas – desde un servidor Web Jboss con Tomcat embebido y lenguaje de programación Java hasta un servidor Web Apache 2.2 y lenguaje de programación PHP 5.3 – es la parte medular del proceso.

Previo a la capacitación de usuarios finales sobre el uso de un sistema ya creado para el Hogar, surgió la necesidad de evaluar el rendimiento de la aplicación antes de adentrarse en un proceso de auto aprendizaje y posterior entrenamiento a los principales interesados. Los resultados del análisis indicaron que en términos de performance, tanto la máquina *servidor* como las máquinas clientes tenían tiempos de respuesta muy alejados de los considerados óptimos para el uso de una aplicación. Después de debatir sobre las posibles causas y soluciones, se determinó que era necesario migrar la aplicación a tecnologías más livianas y subirlo a un hosting externo, en Internet.

El performance del nuevo sistema migrado – medido en tiempos de respuesta frente a solicitudes no livianas pero sí frecuentes – resulta ser mejor que el del sistema anterior y aunque la diferencia es perceptible para todos los usuarios, es mucho más notoria cuando se evalúa el rendimiento de la máquina donde se instaló la base de datos y el servidor Web de aplicaciones del sistema original. Esta misma máquina, una vez liberada de sus funciones de server, es entre un 250 y 275% más eficiente cuando se usa el sistema ya migrado. Además, si se considera que una terminal de usuario común – como es el caso de la máquina mencionada anteriormente – suele ser encendida y apagada con relativa frecuencia, también resulta un problema de disponibilidad el tiempo que transcurre en el arranque del sistema operativo y de servicios propios de la aplicación inicial; alrededor de 8 minutos. En un hosting, este último aspecto mencionado no es un inconveniente pues la disponibilidad ofrecida por el proveedor es del 99.7%.

Finalmente, la accesibilidad del nuevo sistema no está limitada a la intranet del Hogar, sino que por haber sido subida a Internet, está disponible desde cualquier parte del mundo.

INTRODUCCIÓN

Contar con un sistema informático que permita llevar el control de registros y procesos internos, en muchos de los casos no es un requisito para el establecimiento o apertura de centros de atención y ayuda al ciudadano ecuatoriano. En este respecto, los Centros Gerontológicos no son la excepción por lo que mantener registros manuales de adultos mayores con toda su información social y médica, resulta una tarea complicada que conlleva resultados no deseados [1]. Cabe mencionar que el ente regulador de este tipo de instituciones a nivel del Estado Ecuatoriano es el Ministerio de Inclusión Económica y Social (MIES). Actualmente existe un proyecto para integrar los datos de adultos mayores de todos los Centros Gerontológicos del país sin importar si su residencia es parcial o total.

Objetivo General

- Migrar el Sistema de Fichas Médicas del ancianato Hogar Corazón de María de Jboss/Java hacia Apache/PHP con el fin de satisfacer los nuevos requerimientos tanto del centro gerontológico como de su ente regulador, el MIES.

Objetivos Específicos

- Mejorar el rendimiento general del sistema.
- Modificar los datos de pacientes para posibilitar la generación del formulario HCU-form.001 / 2010, solicitado por el MIES.
- Modelar las entidades y relaciones de la base de datos con el fin de permitir el ingreso de varias consultas por un mismo paciente. De esta manera se satisfará el requerimiento del Ministerio de Salud Pública del Ecuador (MSP), la generación del formulario HCU-form.057 / 2010.
- Subir el sistema migrado a Internet.
- Capacitar a los usuarios.

1 SITUACIÓN TECNOLÓGICA ACTUAL DEL HOGAR CORAZÓN DE MARÍA

Para las instituciones sin fines de lucro suele ser una tarea complicada el encontrar sistemas informáticos que se adapten a sus necesidades o que, por lo menos, satisfagan algunas de ellas, preferentemente las más importantes. Esta tarea resulta aún más ardua si lo que se busca es un sistema gratuito. Ante este panorama, existen varias posibles soluciones; entre ellas tenemos:

- Adaptar los procesos de la institución a sistemas ya existentes que no requieran de desembolsos para su adquisición.
- Comprar un sistema que se ajuste lo mejor posible a las necesidades de la institución.
- Desarrollar un sistema basado en los requerimientos específicos de la institución.

Evidentemente la última es la mejor opción, sin embargo está condicionada a la fuente de financiamiento del desarrollo. Las posibles organizaciones o instituciones que podrían participar activamente en un desembolso, son:

- Entidades externas: Estado, ONGs, Universidades, etc.
- La misma institución.

Frente a estas alternativas para la adquisición e implantación de un sistema informático, el Centro Gerontológico “Hogar Corazón de María” ha probado al menos dos de ellas. El primer sistema con el que se intentó llevar los controles necesarios, es uno provisto por el mismo Hogar y que es usado en la matriz y en varias de las sucursales de esta institución alrededor del mundo [2]. Lamentablemente, la aplicación presentó varios inconvenientes que dificultaron su uso y que hacen de éste, un software descalificado en la sucursal ecuatoriana:

- Los requerimientos dados para el desarrollo del sistema incluían varias opciones que no son válidas sino en el país de origen: España. Además, muchas de las características de la aplicación fueron añadidas para controlar servicios prestados únicamente en la matriz. Lo contrario también es cierto, es decir, muchos de los requerimientos específicos del Hogar en la sucursal de Quito, no fueron satisfechos

por esta aplicación.

- Es de tipo cliente-servidor y necesita ser instalado en todos los equipos del Hogar. Este requerimiento resulta un problema para la sucursal quiteña pues las computadoras disponibles no son las adecuadas para instalar aplicaciones 'pesadas'. Varias de ellas se limitan a características como: Sistema Operativo: Windows 98, RAM: 256 y 512M, entre otras. Es decir, la capacidad de operación de las máquinas está limitada y, en algunos casos, hasta excedida con los programas que ya tienen instalados.
- No existe soporte del sistema.

Un segundo sistema usado en el Centro Gerontológico fue desarrollado como tema de tesis de Ingeniería en Sistemas y patrocinado por la Pontificia Universidad Católica del Ecuador. Es justamente sobre este sistema que se realizaron modificaciones y mejores que dieron pie a una migración total de las plataformas y herramientas seleccionadas en un inicio. Todos estos cambios originaron este proyecto de tesis.

1.1 Características del Sistema Anterior

El sistema anterior contenía funcionalidades solicitadas específicamente por los diferentes departamentos del Hogar: Administración, Acción Social y Enfermería. Todas las características dentro de la aplicación eran controladas desde tres módulos principales: Administración, Configuración y Fichas Médicas.

Cabe mencionar que el análisis de las características de tal sistema, tiene como alcance la funcionalidad perceptible por usuarios administradores y finales. No se contempla ningún análisis técnico de cómo fue concebida y construida la aplicación.

1.1.1 Administración

La entera gestión de permisos es controlada desde el módulo de Administración, mismo que respeta la siguiente jerarquía: opciones → permiso → rol → usuario. Ejemplificando se dice que: un *usuario* A puede acceder a una *opción* X, únicamente si el *rol* al que dicho *usuario* está asignado, tiene *permisos* de visualizar y/o modificar los datos de la *opción* X.

1.1.1.1 Opciones

Las opciones del sistema se refieren específicamente a dos tipos de elementos: los menús de la barra principal y los permisos internos para realizar acciones en la ficha médica.

Los menús y sub menús de opciones disponibles – a las que se llamará de primer tipo en adelante en este capítulo – son los siguientes:

- Sistema
    - Inicio
    - Salir
- Pacientes
    - Nuevo
    - Búsqueda
    - Reportes
- Usuarios
    - Nuevo
    - Administrar
    - Reportes
    - Roles

Una vez dentro de la opción 'Paciente' existen a su vez, varios elementos clasificados también como opciones – a las que se llamará de segundo tipo en adelante en este capítulo:

- Datos Generales
- Diagnóstico
- Contacto
- Estructura Familiar
- Motivo de Consulta y Enfermedad Actual
- Antecedentes Patológicos Personales
- Hábitos
- Historia Gineco-Obstetra
- Antecedentes Patológicos Familiares

- Revisión de Sistemas
- Examen Físico
- Impresión Diagnóstica
- Recomendaciones y Tratamientos
- Pate Diario
- Evolución Prescripciones
- Terapia Física
- Análisis Psicológico
- Corte Vertical
- Corte Horizontal

Además, en este mismo grupo existe una opción adicional encargada de controlar los permisos del administrador:

- Opciones de Administrador

Si una opción del primer tipo está habilitada, un usuario podrá hacer clic sobre ella y ver su contenido, caso contrario la opción no estará disponible, lo que conlleva que tampoco aparecerá como parte del menú. De manera similar, si una opción del segundo tipo está habilitada, un usuario siempre podrá ver su contenido, pero dependerá de si tiene permisos de escritura para determinar si podrá realizar cambios sobre la información existente o ingresar información nueva. Una explicación más detallada se encuentra disponible en la siguiente sección, 'Permisos'.

Todas las opciones del sistema están programadas con código duro y si existiera un nuevo requerimiento de creación de ítems o modificación y eliminación de los ya existentes, sería necesario realizar cambios en el código fuente de la aplicación.

1.1.1.2 Permisos

Un 'Permiso' en el contexto informático indica si un usuario puede o no realizar una acción específica sobre una opción disponible. Es necesario mencionar que en el sistema actual existen dos niveles y dos tipos de permisos.

Los niveles de permisos se refieren bien a los Menús de Opciones o bien a los Paneles de Usuario, es decir, existen permisos para que el usuario pueda acceder a las opciones del sistema (Nivel: Menú de Opciones) y permisos para que – una vez dentro de la opción especifica denominada 'Ficha Médica' – se puedan realizar acciones sobre sus sub opciones (Nivel: Paneles de Usuario).

Una vez identificado que un usuario A tiene permiso sobre una acción X, es indispensable conocer si es que ese permiso le posibilita al usuario únicamente visualizar la información o también manipularla. Es aquí cuando el 'tipo' de permiso entra en funcionamiento mediante la división de éste en dos: 'lectura' y 'escritura'. Por concepto, el permiso de lectura permite visualizar datos mientras que el de escritura posibilita realizar 3 operaciones básicas: creación, modificación o eliminación de la información. En el caso particular del sistema a ser migrado, existen dos particularidades a tener en consideración 1) la eliminación de información no es permitida y 2) el permiso a nivel de Menú de Opciones es único (no está dividido en lectura y escritura). Como premisa de permisos de lectura/escritura, es necesario indicar que para tener permiso de escritura, se debería haber asignado previamente el permiso de lectura, caso contrario no se podría conocer qué es lo que se está modificando.

La relación entre opciones y permisos está dada, en el caso de las opciones del primer tipo, por un check-box que indica si el elemento está o no habilitado y, en el segundo caso, por un par de check-boxes que indican la misma habilitación pero a nivel de permisos de escritura y lectura.

1.1.1.3 Roles

Un rol es una asociación de permisos que identifica específicamente qué puede y qué no puede hacer un usuario. El sistema original cuenta con 5 roles preestablecidos:

- Administrador
- Trabajo Social
- Médico
- Fisioterapia
- Sicología

El autor del sistema recomienda no cambiar los permisos asociados a los roles mencionados, pues las acciones requeridas para asegurar el correcto funcionamiento del sistema podrían quedar fuera del alcance de todos los usuarios [C], incluido el Administrador. Por lo tanto, si un nuevo usuario no encaja dentro de estos roles, se debería crear uno nuevo al que se le asignen los permisos pertinentes y asociarle éste al usuario en cuestión.

En la creación de un nuevo rol o edición de uno ya existente, se deben especificar cuáles son las opciones habilitadas y cuáles no.

1.1.1.4 Usuarios

La entidad con la que más frecuentemente se relaciona a un ser humano en el mundo informático, es la de 'Usuario'. El sistema a ser migrado no es la excepción de la regla, es decir, cada usuario corresponde a una persona que usa el sistema de manera activa.

Es probable que una persona no autorizada intente ingresar al sistema, por lo tanto, en el contexto de esta aplicación, un usuario es cualquier persona con la capacidad de ingresar de manera autorizada y autenticada al sistema mediante la escritura correcta de un nombre de usuario y contraseña válidos.

Entre las características particulares de la entidad 'usuario', están las siguientes:

- Cada usuario puede y debe tener un solo rol.
- La eliminación de un usuario como tal, no es posible. Se obvió esta opción con el fin de mantener la integridad referencial de pistas de acciones ejecutadas por la persona asociada al usuario. En caso de que se necesite restringir el acceso al sistema de un usuario específico, la alternativa es marcarlo como 'inactivo'.

1.1.2 Configuración

Existen varios datos de fichas médicas que pueden estar limitados a una lista de valores fija para el usuario final pero configurable por el administrador. Esta configuración de valores

permite que los usuarios no tengan que ingresar texto de manera manual - sensible a errores de escritura, sino que seleccionen una opción del listado predefinido. Los datos de fichas médicas que se adaptan a este comportamiento, son (entre paréntesis las posibles opciones de cada uno):

- Estado (Activo, Inactivo).
- Estado Civil (Soltero, Casado, Viudo, Divorciado, Unión Libre).
- Género (Masculino, Femenino).
- Instrucción (Ninguna, Primaria, Secundaria, Superior).
- Tipo de Paciente (Discapacitado, Enfermedad Crónica, Tercera Edad, Bono de Desarrollo Humano, Otro).
- Demanda (Institucional, Particular).

1.1.3 Fichas Médicas

El objetivo principal del sistema es llevar un control de fichas médicas y psicológicas de los pacientes del Hogar. Para este fin se dividió el módulo en varias secciones que agrupan datos relacionados entre sí y que permiten obtener información relevante cuando se la considera en conjunto. Esta división y agrupación de los datos ha sido ejecutada por el Ministerio de Salud Pública.

1.1.3.1 Datos Generales

Antes de entrar en detalles exclusivamente de carácter médico o psicológico, es necesario identificar de manera única a cada paciente del Hogar. La sección “Datos Generales” contiene información altamente relevante de los adultos mayores y que permite la distinción de cada paciente.

Campo	Descripción
Apellidos	Apellidos paterno y materno.
Nombres	Primer y segundo nombre.
Cédula	Número de cédula de identidad o de pasaporte, en el caso de personas extranjeras.
Estado Civil	Estado Civil del paciente. La selección de este dato se la realiza mediante una lista desplegable previamente definida. Las posibles opciones son: Soltero, Casado, Viudo, Divorciado y Unión Libre.
Fecha de Ingreso	Fecha en la que el paciente ingresó al Hogar.
Género	Sexo del paciente. La selección de este dato se la realiza mediante una lista desplegable previamente definida. Las opciones son: Masculino y Femenino.
Fecha de Nacimiento	Fecha de nacimiento.
Edad	Edad al momento de visualizar los datos. Este campo no es ingresado manualmente sino que se calcula mediante la resta de la fecha actual y la fecha de nacimiento.
Instrucción	Instrucción educacional del paciente. La selección de este dato se la realiza mediante una lista desplegable previamente definida. Las posibles opciones son: Ninguna, Primaria, Secundaria y Superior.
Procedencia	Lugar de origen.
Dirección	Dirección.
Ingreso Mensual	Algunos adultos mayores colaboran mensualmente con la institución. Este campo recoge dicho valor.
IESS	Indica si el paciente está afiliado al IESS. El campo – de tipo radio button - únicamente permite las opciones Sí o No.

Tabla 1-1 Datos Generales [A]

1.1.3.2 Diagnóstico

La primera sección con datos relativos a información médica del paciente, es la sección de 'Diagnóstico'. Existe cierta inconsistencia sobre la posición de esta sección (2da) dentro de la ficha médica debido a que el diagnóstico debería ser generado después de conocer otros

datos del adulto mayor, por ejemplo: antecedentes patológicos, antecedentes personales y familiares, examen físico, etcétera. Sin embargo, si el 'Diagnóstico' es tentativo o preliminar, basado en las condiciones perceptibles a primera vista, las dudas quedan despejadas.

Campo	Descripción
Referencia de	Indica la persona o institución que recomienda al paciente.
Diagnóstico Médico	Primer diagnóstico médico en base a datos recogidos de los encargados del adulto mayor en su anterior lugar de residencia: otra institución o casa de familiares.
Diagnóstico Social	Primer diagnóstico social en base a datos recogidos de los encargados del adulto mayor en su anterior lugar de residencia: otra institución o casa de familiares.
Tipo de Paciente	Los posibles valores para este campo son: 'Discapacitado', 'Enfermedad Crónica', 'Tercera Edad', 'Bono de Desarrollo Humano' y 'Otro'. Solo una de estas opciones puede ser seleccionada para cada paciente, sin embargo, por simple inspección se concluye que varias de ellas podrían ser aplicadas a un adulto mayor: Ejemplo: 'Discapacitado', 'Tercera Edad'.

Tabla 1-2 Diagnóstico [A]

1.1.3.3 Contacto

Cada adulto mayor puede tener un único contacto (persona) con el cual comunicarse para atender asuntos desconocidos por miembros del hogar o sobre los cuales no se tiene jurisdicción.

Campo	Descripción
Nombre	Nombre completo del contacto. Incluye nombres y apellidos.
Parentesco	Parentesco o afinidad con el paciente.
Dirección	Dirección del contacto.
Teléfono	Teléfono del contacto. Nótese que éste es el único dato inmediato de contacto antes posibles emergencias.

Tabla 1-3 Contacto [A]

1.1.3.4 Estructura Familiar

Los datos de familiares del adulto mayor recaen en esta sección. Cabe mencionar que no existe información de contacto de tales personas, siendo esa una particularidad de la sección inmediata anterior.

Campo	Descripción
Nombre	Nombre completo del familiar. Incluye nombre y apellidos.
Relación	Parentesco con el paciente.
Ocupación	Ocupación de la persona.

Tabla 1-4 Estructura Familiar [A]

1.1.3.5 Motivo de Consulta y Enfermedad Actual

Un único campo denominado ‘Anotaciones’ es considerado dentro de la sección ‘Motivo de Consulta y Enfermedad Actual’. Obviamente, la información que puede ser ingresada aquí es parte de un área de texto que proporciona al usuario las facilidades para adentrarse en detalles.

1.1.3.6 Antecedentes Patológicos Personales

El ingreso de un paciente está condicionado por el número y el tipo de antecedentes patológicos que éste pueda tener y las posibilidades de recibir la atención médica respectiva, dentro del Hogar. La siguiente tabla muestra un listado de las enfermedades que pueden afectar a los adultos mayores.

Campo	Descripción
Poliomielitis	Existe un check box para cada una de las enfermedades listadas en la columna izquierda. Un check box marcado indica que el antecedente patológico se encuentra presente en el paciente.
Tifoidea	
Respiratorias Altas	
Ulcero pépticas	
Venéreas	
Hospitalizaciones	
Hepatitis	
Fiebre Reumática	
Asma	
Infecciones Urinarias	
Tuberculosis	
Paludismo	
Erupciones de la Infancia	Datos relevantes sobre erupciones que el paciente haya sufrido durante su infancia.
Alergia a Medicamentos	Lista detallada de medicamentos que puedan ocasionar reacciones alérgicas.
Otras Alergias	Detalle de otro tipo de alergias (Diferentes a las causadas por medicamentos).
Cirugías	Enumeración y detalle de todas las cirugías a las que el paciente se haya sometido.
Otras Enfermedades	Detalle de enfermedades no listadas en la primera sección de esta tabla.

Tabla 1-5 Antecedentes Patológicos Personales [A]

1.1.3.7 Hábitos

Tanto los hábitos positivos como los negativos constituyen información relevante de los pacientes. En base a éstos se pueden identificar tendencias y pronosticar enfermedades a los que el adulto mayor es más proclive. Además, tomando estos datos como referencia, se puede diagnosticar medicamentos y posibles dietas [3].

Campo	Descripción
Tabaquismo	Existe un check box para cada una de los hábitos nocivos listados en la columna izquierda. Un check box marcado indica que el paciente tiene el hábito.
Alcoholismo	
Otras Drogas	Detalle de otras drogas.
Deposición	Detalle de deposición.
Orina	Detalle de orina.
Alimentación Cantidad	Observaciones referentes a la cantidad de alimentación ingerida.
Alimentación Calidad	Observaciones referentes a la calidad de alimentación ingerida.
Sueño	Detalle de trastornos del sueño.
Deportes	Check box que indica si el paciente ha acostumbrado o no a realizar algún deporte.

Tabla 1-6 Hábitos [A]

1.1.3.8 Historia Gineco Obstetra

A la fecha existen 220 residentes activos del Hogar y de éstos 116 son mujeres; esto equivale al 53%. Para esta mayoría existen datos exclusivos de gran importancia para fines prescriptivos y diagnósticos.

Campo	Descripción
Menarquia	Edad de menarquía o primer episodio de sangrado vaginal de origen menstrual [4].
Ciclos	Número de días que comprenden el ciclo mensual [5].
F.U.M.	Fecha de la última menstruación.
Dismenorrea	La Dismenorrea es una enfermedad que indica dolor durante la menstruación [6]. Por tal motivo, este campo debió haber sido concebido como un check box que indique si la paciente sufrió o no de esta enfermedad.
Embarazos	Número de embarazos.
Abortos	Número de abortos.
Partos	Número de partos.
Cesáreas	Número de cesáreas.
Hijos Vivos	Número de hijos vivos a la fecha.

Tabla 1-7 Historia Gineco Obstetra [A]

1.1.3.9 Antecedentes Patológicos Familiares

Se consideran de riesgo los antecedentes patológicos en familiares de primer grado de consanguinidad. Es decir, si un familiar como madre o padre padeció de alguna o varias de las enfermedades mencionadas en la Tabla 1-8, es probable – en menor o mayor grado dependiendo de la patología – que el paciente también la padezca [7].

Campo	Descripción
Diabetes	Existe un check box para cada una de las enfermedades listadas en la columna izquierda. Un check box marcado indica que el antecedente patológico está presente en familiares del paciente.
Cáncer	
Asma	
Tuberculosis	
Hipertensión	
Desnutrición	
A.C.V.	
Cardiopatías	
Epilepsia	
Alteraciones Mentales	
Tiroides	
Obesidad	

Tabla 1-8 Antecedentes Patológicos Familiares [A]

1.1.3.10 Revisión de Sistemas

La Revisión de Sistemas se limita a un campo de anotaciones donde se detallan aspectos importantes del interrogatorio sobre los diferentes sistemas del cuerpo humano. En especial se usará esta sección/campo para guardar datos relevantes relacionados con síntomas o afecciones duraderas.

1.1.3.11 Examen Físico

El Examen Físico contiene datos relevantes del paciente tomados generalmente por enfermería durante cada consulta y son parte fundamental de la historia clínica del adulto mayor. En base a las variaciones entre consulta y consulta se suele medir la evolución médica.

Campo	Descripción
Peso	Peso en Kilogramos.
Talla	Talla en metros.
Frecuencia Cardiaca	Frecuencia Cardiaca por minuto.
Pulso	Número de pulsos por minuto.
Frecuencia Respiratoria	Frecuencia Respiratoria por minuto.
Temperatura	Temperatura en grados centígrados.
T.A.	Tensión Arterial en mmHg (milímetros de Mercurio).
I.M.C.	Índice de Masa Corporal en Kg/m^2 (Peso sobre el doble de la Talla.
Piel y Anexos	Datos relevantes de cada uno de los órganos, sistemas, aparatos o afines de la columna izquierda.
Cabeza	
Cuello	
Tórax	
Abdomen	
Ano-genital	
Aparato Locomotor	
Neurológico	
Escalas	

Tabla 1-9 Examen Físico [A]

1.1.3.12 Impresión Diagnóstica

Esta sección contiene los datos de la impresión diagnóstica determinada por el médico después de revisar y obtener información de las secciones anteriores.

Campo	Descripción
I.C.D.	Código de Clasificación Internacional de Enfermedades.
Anotaciones	Detalle de la enfermedad actual.

Tabla 1-10 Impresión Diagnóstica [A]

1.1.3.13 Recomendaciones y Tratamientos

El campo ‘anotaciones’ permite al médico ingresar detalladamente las recomendaciones y tratamientos frente a la impresión diagnóstica registrada en la sección anterior. Esta información a su vez es usada por el departamento de enfermería para asegurarse de que lo indicado aquí se cumpla adecuadamente para cada paciente.

1.1.3.14 Parte Diario

El departamento de Enfermería usa la presente sección para describir el parte diario realizado sistemáticamente para cada paciente.

Campo	Descripción
Anotaciones de Hoy	Parte diario detallado.

Tabla 1-11 Parte Diario [A]

1.1.3.15 Evolución y Prescripciones

Una vez más es el departamento de Enfermería quien se responsabiliza de ingresar la evolución médica y las prescripciones del paciente. La diferencia entre esta sección y la inmediata anterior radica en que la evolución médica es usada exclusivamente para dar seguimiento a las recomendaciones y tratamientos indicados por un doctor, mientras que en el parte diario se pueden ingresar particularidades adicionales.

Campo	Descripción
Evolución de Hoy	Datos de la evolución con respecto a las recomendaciones y tratamientos dictados por el médico.
Prescripción de Hoy	

Tabla 1-12 Evolución y Prescripciones [A]

Nótese que no existe una descripción del campo ‘Prescripción de Hoy’. Este particular se debe a la contradicción que genera el hecho de que enfermería pueda realizar una prescripción médica diaria. Usualmente, esta es una tarea designada exclusivamente al

médico o médicos encargados.

1.1.3.16 Terapia Física

Las terapias médicas y psicológicas son detalladas, al igual que las evoluciones, por el departamento de Enfermería.

Campo	Descripción
Terapia de Hoy	Terapia Médica diaria.
Terapia Psicológica de Hoy	Terapia Psicológica diaria.

Tabla 1-13 Terapia Física [A]

1.1.3.17 Análisis Psicológico

A pesar de haber una sección anterior para registrar datos de terapia física, es en este apartado donde se detalla la ficha psicológica del paciente y se realizan observaciones y recomendaciones sobre la misma.

Campo	Descripción
Demanda	Permite seleccionar de una lista desplegable quien realiza la demanda del análisis psicológico: una institución o un ente particular.
Historia Personal	Datos psicológicos anteriores
Síntomas y/o Conflictos	Problemas psicológicos detectados
Impresiones y Comentarios	Impresiones del psicólogo encargado
Tratamiento Psicológico	Tratamiento requerido en caso de ser necesario
Observaciones	Observaciones sobre síntomas, impresiones y/o tratamiento.
Recomendaciones	Recomendaciones generales.

Tabla 1-14 Análisis Psicológico [A]

Con respecto al campo denominado ‘Demanda’, se desconoce por qué se necesita este dato como parte de la información relevante para el análisis psicológico del paciente.

1.1.3.18 Cortes Vertical y Horizontal

Los cortes vertical y horizontal permiten conocer datos relevantes del aspecto psicológico del paciente. En ambos casos la información que puede ser registrada es idéntica, tal como se puede apreciar en la siguiente tabla:

Campo	Descripción
Familia	Datos de Familiares: nombres, parentesco, etc.
Pareja	Datos de la Pareja: nombre, tipo de relación, descripción de la relación.
Trabajo	Datos relevantes de los trabajos en los que se ha desempeñado o se sigue desempeñando.
Relaciones Sociales	Información de relaciones sociales.
Autopercepción	Información acerca de cómo se ve el paciente a él mismo.

Tabla 1-15 Corte Vertical y Horizontal [A]

1.2 Performance del Sistema Anterior

El performance de una aplicación es uno de los factores primordiales que influyen en la opinión de los usuarios sobre un sistema. No es raro oír o incluso compartir malas opiniones de una aplicación que, a pesar de contar con las características requeridas, su rendimiento es pobre y exaspera a quien tiene que perder tiempo con su uso. Sin embargo, antes de entrar en detalles sobre el performance o rendimiento del sistema anterior, es necesario indicar ciertos factores que se suelen tener en cuenta al realizar análisis de rendimiento de una aplicación.

1.2.1 Comparación de Lenguajes

Hacer comparaciones entre un lenguaje y otro es una tarea bastante compleja si se considera que los análisis que se puedan realizar tienen que buscar equilibrar la balanza lo mejor posible entre los contendientes.

Las características que se suelen comparar, son las siguientes [8]:

- Modularización: capacidad de dividir el sistema en varios módulos.
- Mantenibilidad: facilidad en el mantenimiento del sistema.
- Seguridad: detección y bloqueo ante accesos no deseados.
- Rendimiento: rapidez del sistema frente a solicitudes de clientes.
- Escalabilidad: rendimiento y funcionalidad ante un número de usuarios mayor al común.
- Usabilidad: facilidad para usar un sistema.
- Robustez: comportamiento de un sistema ante eventos no contemplados.

Todos estos factores son descritos en mayor o menor profundidad en los sitios oficiales de cada lenguaje de programación pero ninguna de estas Webs se aventura a hacer una comparación de su lenguaje con sus similares. ¿Por qué? Pues la respuesta está relacionada con los siguientes puntos:

- Sería necesario comparar un sistema idéntico desarrollado en 2 lenguajes de programación. Cabe resaltar que la comparación se la debe realizar sobre un sistema y no sobre una característica simple porque el comportamiento varía dependiendo del tamaño de la aplicación. Tomando como ejemplo la escalabilidad, de más está decir que ésta tendría variaciones importantes si lo que se compara es una simple pantalla de inicio sesión siendo usada por n usuarios a la vez o si lo que este mismo número de usuarios está haciendo es sacando un balance de pérdidas y ganancias al final de un año fiscal de labores. Las preguntas obvias serían: ¿Quién puede construir un sistema tan detallado que permita tener una idea medianamente acertada de qué característica funciona mejor por haber sido desarrollada en un lenguaje X bajo ciertas circunstancias dadas? ¿Tendría acogida global este análisis – especialmente entre los desarrolladores que suelen usar el lenguaje que no obtuvo la mejor calificación?
- Tan grande es la variedad de características de un lenguaje que resulta difícil comparar, digamos, el comportamiento de una capa X de una aplicación siendo *afectada*, por ejemplo, por un servicio externo (suponiendo el tiempo de respuesta de este servicio como 0), contra esa misma capa por sí sola, es decir sin afectaciones externas, incluso usando el mismo leguaje en ambos casos. Sin embargo, al comparar 2 o más lenguajes, los resultados suelen ser diversos y lo que para uno resulta en una merma de rendimiento, para el otro podría significar un alza

en su performance. En este aislado ejemplo sobresalen circunstancias que usualmente están fuera del alcance de los análisis realizados y publicados en Internet.

Los dos puntos anteriores son solo ejemplos de la larga lista de factores que nos hacen pensar que comparar dos lenguajes para un marco general de desarrollo, no es una idea acertada. Obviamente, la situación cambia cuando el sistema es puntual y se conocen sus requerimientos específicos e incluso su tendencia de crecimiento a mediano y largo plazo.

1.2.2 Infraestructura Tecnológica del Hogar

La infraestructura tecnológica del Hogar "Corazón de María" es limitada; actualmente se cuenta con 9 computadores. La siguiente tabla muestra más detalles:

Ubicación	S. O.	Procesador	RAM	Notas
Planta Baja – Medicina	Windows XP Home Edition SP 2	Pentium III, 350MHz	512MB	
Planta Baja – Acción Social	Windows 7 Ultimate SP 1	Intel Core 2 Duo, 2 GHz	2GB	No existía cuando el sistema actual fue implementado
Planta Baja – Enfermería	Windows 98	Pentium II, 350MHz	256MB	
1er Piso – Dirección	Windows XP Home Edition SP 2	Pentium III, 700 MHz	512MB	Usada como servidor de aplicaciones y base de datos
1er Piso – Contabilidad	Windows XP Home Edition SP 2	Pentium III, 700 MHz	512MB	
1er Piso – Biblioteca	Windows 98	Pentium II, 350MHz	256MB	
1er Piso – Enfermería	Windows 98	Pentium II, 350MHz	256MB	
1er Piso – Salón de Clases	Windows 7 Ultimate SP 1	Intel Core 2 Duo, 2GHz	2GB	No existía cuando el sistema actual fue implementado

Tabla 1-16 Infraestructura Tecnológica [B]

1.2.3 Rendimiento del Sistema

Tomando en consideración lo mencionado en la sección 'Comparación de Lenguajes', este apartado no contiene información relativa a por qué se seleccionó un nuevo lenguaje de programación en lugar de otro, sino que hace referencia a los dos problemas particulares con los que cuenta el sistema anterior y que pretenden ser solucionados con el sistema migrado.

1. El computador en el que se ha instalado el servidor de aplicaciones Web y el motor de base de datos, no tiene las características adecuadas para soportar esta infraestructura de software, por tal razón, el arranque de los servicios demora en promedio, 8 minutos. Además, debido a que hace las veces de servidor, requiere estar encendida todo el tiempo.
2. Como consecuencia del primer problema, los usuarios finales perciben la lentitud del servidor y experimentan demoras en tiempos de respuesta. Si bien estas demoras no son extremadamente grandes, sí son notorias.

1.3 Nuevos Requerimientos

La migración del sistema puede ser aprovechada para implementar nuevos requerimientos y, de hecho, existen otras características que deben ser incluidas por solicitud directa del ente regulador de centros gerontológicos: El Ministerio de Inclusión Económica y Social. Además, el mismo Hogar ha pedido funcionalidades adicionales.

1.3.1 Requerimientos del MIES

La ficha médica del sistema anterior presenta varias diferencias con la exigida por el MIES en la actualidad, en especial porque la nueva ha sido creada con el objetivo de satisfacer las necesidades exclusivas de adultos mayores, mientras que la anterior estaba más generalizada y pretendía cubrir datos de pacientes de cualquier edad. Debido a la gran extensión de la tabla referente a la ficha médica, se la incluye como el primer anexo del documento.

Todas las secciones con sus respectivos campos serán incluidos en el nuevo sistema. Los cambios que conlleva la adaptación de la ficha existente a la solicitada por el MIES constituyen el nuevo requerimiento del ente regulador del centro gerontológico. Cabe mencionar que esta nueva reestructuración de datos posibilitará la obtención del reporte HCU-form.057/2010 solicitado por otro ministerio, el de Salud Pública del Ecuador. Además, también se obtendrá el reporte HCU-form.001/2010 basado en los datos generales de cada paciente, por lo que será necesario realizar modificaciones al Modelo Entidad Relación anterior. En un tópico posterior se explicará detalladamente los cambios en lo que se ha incurrido para satisfacer este requerimiento.

1.3.2 Requerimientos del Hogar

El centro gerontológico tiene requerimientos particulares que también serán satisfechos en el sistema migrado. A continuación se enumeran las nuevas necesidades recolectadas por los diferentes departamentos:

1. Registro de ingreso de paciente por primera vez así como de posteriores egresos y reingresos.
2. Almacenamiento de registros históricos por cada cambio aplicado a los datos generales de un paciente.
3. Posibilidad de manejar datos de consultas médicas independientemente de los datos generales del paciente. Es decir, permitir administrar n número de historias clínicas en lugar de una sola. Además, por cada consulta se podrán ingresar n número de diagnósticos y tratamientos y, a su vez, existirá la misma relación entre diagnósticos/tratamientos y detalle de evolución médica.
4. Administración disgregada de permisos de usuarios.
5. Búsqueda avanzada de pacientes mediante filtros disponibles por cada campo de los datos generales del adulto mayor. La búsqueda se debe poder ejecutar por un campo individual, por un conjunto de campos o por la totalidad de los mismos. Es decir, si existen 3 campos: A, B y C; la búsqueda se podrá realizar de la siguiente manera:
    - Por A
    - Por B
    - Por C

- Por A y B
- Por A y C
- Por B y C
- Por A, B y C

6. Generación de reportes colectivos e individuales sobre información médica y de trabajo social.

1.3.3 Modelo de Requerimientos

Independientemente del origen de los requerimientos, para su mejor administración éstos han sido clasificados en diferentes categorías dependiendo de la funcionalidad que realizarán dentro del sistema. Todas estas categorías y los requerimientos individuales que las conforman han sido modelados usando una herramienta de tipo case llamada Enterprise Architect, en su versión 7.5:

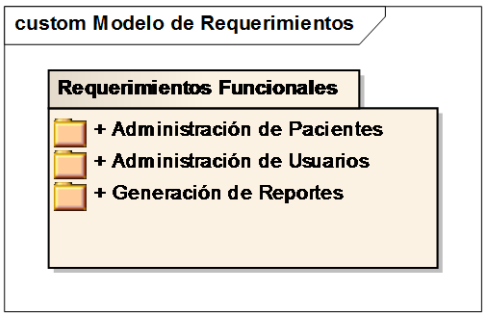


Ilustración 1-1 Categorización de Requerimientos [B]

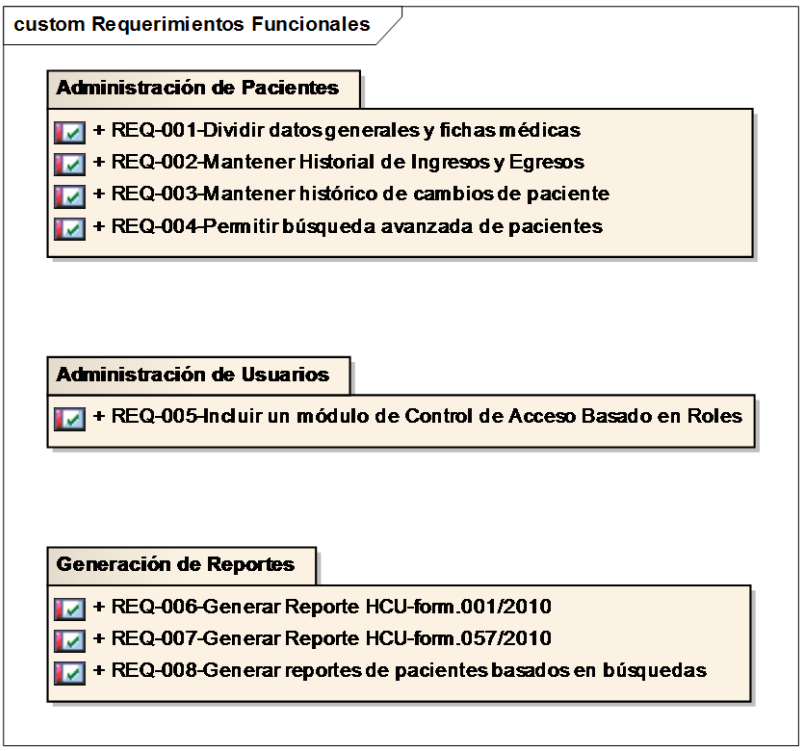


Ilustración 1-2 Listado Detallado de Requerimientos [B]

2 MODELO ENTIDAD RELACIÓN

En el desarrollo de aplicaciones de software existen varias alternativas para cumplir con el objetivo de almacenar permanentemente los datos. Estas posibilidades van desde un documento de texto simple o una hoja electrónica, hasta bases de datos relacionales y basadas en objetos.

Dentro del universo de opciones disponibles, son las bases relacionales las más populares por ser ampliamente estudiadas en instituciones formativas y, en especial, por los resultados positivos que conlleva su uso. Tomando este tipo de base de datos como referencia, conocemos que si un modelo físico de datos ha respetado los principios básicos de normalización, se logran los siguientes objetivos:

- No redundancia de datos.
- Eliminar errores estructurales al actualizar datos.
- Mantener y proteger la integridad de la Información [9].

El Modelo Entidad Relación (MER) ha sido una de las bases primordiales para la creación, el mantenimiento y la migración de sistemas. Si bien existen decenas de modelos usados para el desarrollo e implementación de aplicaciones de software, lo más relevante suele ser la información que se desea guardar y mantener así como la estructura que la soporta. Es por esta razón que el grado de dificultad del soporte de un sistema y la viabilidad de adaptación a tecnologías diferentes depende, en mayor medida, de cómo se haya creado el modelo físico de la base de datos.

2.1 Obtención del MER del Sistema Anterior

La base de datos del sistema anterior está soportada por el motor MySql, propiedad de Oracle Corporation.

A pesar de que el sistema está instalado en la máquina ubica en la Dirección General del Hogar, los servicios que soportan su uso no estaban inicializados y al intentar ponerlos en funcionamiento, el proceso por lotes presentaba un error de arranque. Este particular, además de indicar a las claras que el sistema no estaba siendo utilizado, también suponía

un reto en la obtención del modelo físico. Es por esto que para lograr el objetivo de recuperar la estructura de datos actual, fue necesario cambiar archivos XML usados para inicializar el servicio 'mysqld'. Después de una modificación en los parámetros de la red local del Hogar y de la creación de una nueva intranet que funcione paralelamente con la anterior, la subred en la que estaba instalado el servidor de base de datos, cambió su tercer octeto. Debido a este cambio y a que el servidor apuntaba a una IP fija y no al comúnmente usado 'localhost', fue necesario realizar pequeñas alteraciones en los archivos de configuración, con el objetivo de acceder a la base.

Teniendo la IP del servidor de base de datos (localhost), el puerto por el que escucha el servicio msyqld (3306, por defecto) y el usuario con accesos totales en cualquier instalación de la base mysql (root), faltaba un dato no menor para poder ingresar: la contraseña de este último usuario. Una vez más se echó mano de los archivos de configuración para acceder a esta información. Cabe resaltar que el acceso a dichos archivos fue posible gracias a la Dirección del Hogar, que otorgó el permiso necesario para usar la máquina que hacía las veces de servidor de base de datos. La clave encontrada y efectivamente usada para acceder fue: jetblue.

Un nivel más de seguridad que se suele sortear al intentar acceder a una base de datos mysql, es el permiso de ingreso desde una IP autorizada; ventajosamente, al acceder desde la misma máquina no se presentó inconveniente alguno pues la instalación por defecto de mysql otorga permisos de administración a conexiones efectuadas desde localhost.

Con todos los permisos administrativos y tecnológicos otorgados, el acceso a la base fue posible. Haciendo uso de la herramienta MySql Administrator ya instalada en el servidor, se obtuvo un archivo de exportación que a diferencia de otros sistemas gestores de base de datos como Oracle que generan archivos comprimidos de tipo dump, el archivo obtenido de MySql resulta únicamente una combinación lógica y ordenada de sentencias DML (Data Manipulation Language) y DDL (Data Definition Language) con la extensión '.sql' correspondiente.

El archivo SQL ha sido importado en una máquina con un sistema operativo distinto (Fedora 14) al instalado en el host que soportaba el sistema anterior (Windows XP). Fedora

14, al igual que la mayoría de distribuciones de Linux, permite la instalación nativa de módulos de desarrollo. Entre estos módulos, también conocidos como paquetes, existe la posibilidad de instalar XAMPP – una integración del servidor Web Apache, la base de datos MySql y los lenguajes de programación PHP y Perl para varias plataformas, pero nativamente para aquellas basadas en UNIX – y justamente se ha usado esta agrupación de herramientas para la migración de la base de datos y del sistema en general. Haciendo uso de uno del sub módulos de XAMPP conocido como PhpMyAdmin, se importó el archivo de la base de datos sobre la plataforma que será usada como hosting: Linux.

Los resultados de esta importación pueden ser apreciados de manera más detallada y entendible al ojo humano, al ser obtenido el modelo físico de la base de datos gracias al uso de Enterprise Architect:

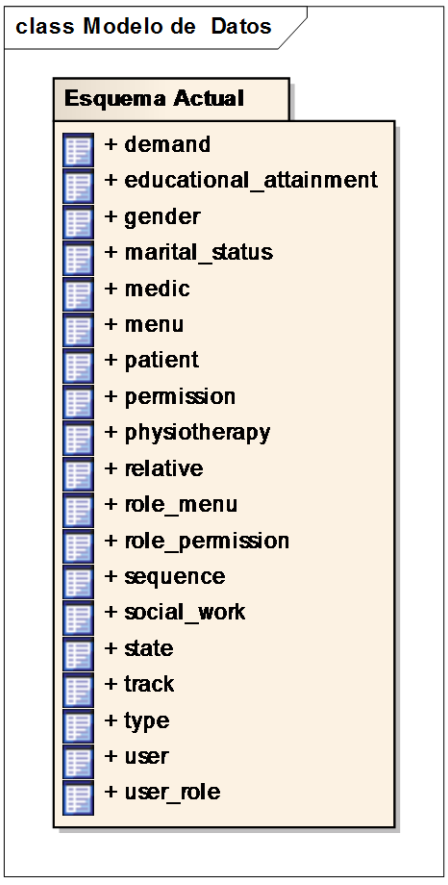


Ilustración 2-1 Listado de Tablas del Sistema Anterior [C]

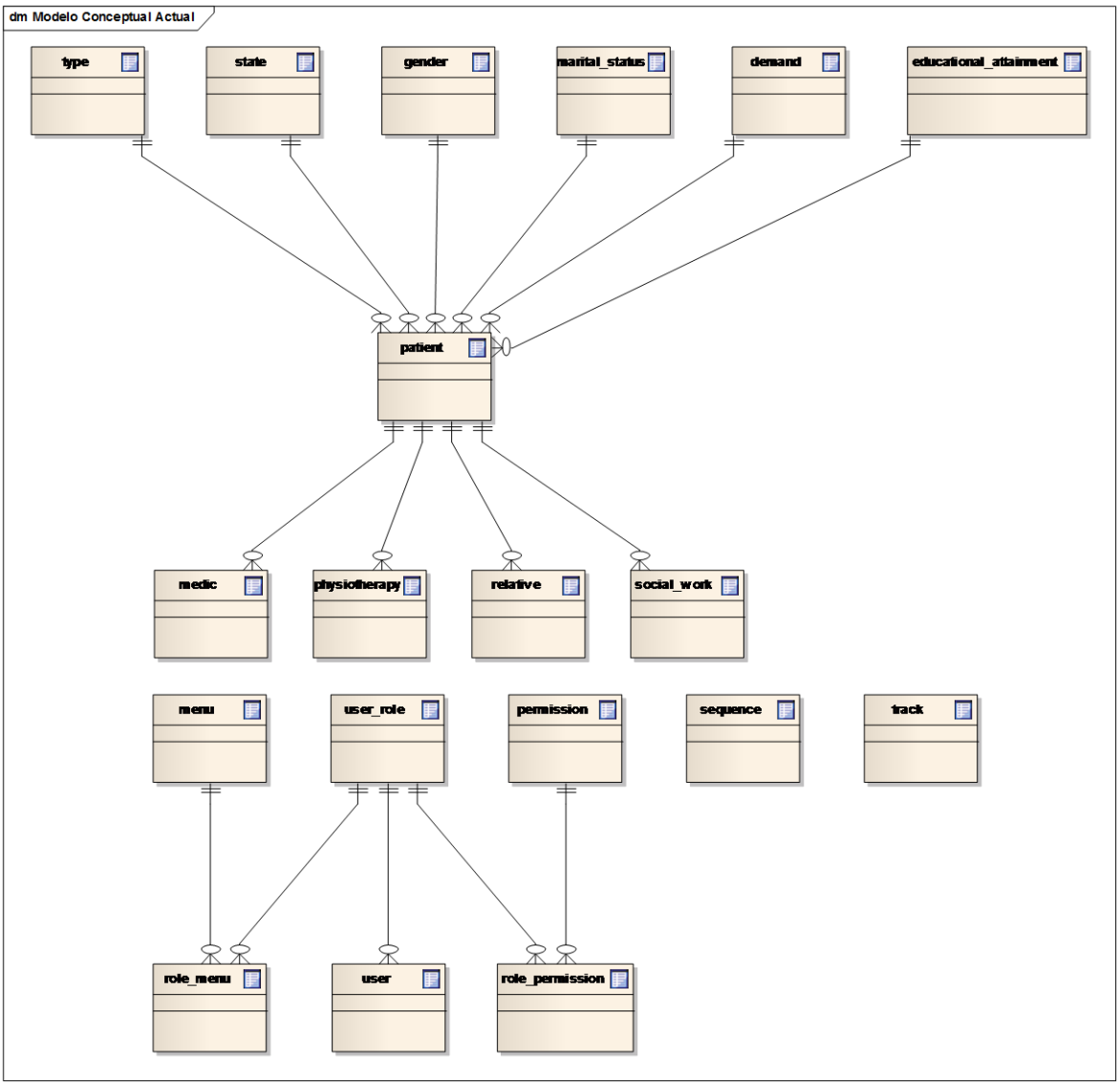


Ilustración 2-2 MER del Sistema Anterior [C]

Nótese que el modelo mostrado en la figura 2.2 es un modelo conceptual y no uno físico. Este particular se debe a que el gran tamaño de la tabla 'patient' por el número de campos con el que fue concebida, no permite plasmar físicamente toda la entidad – con sus atributos y relaciones – en una sola hoja.

A pesar de no entrar en detalles del Modelo Entidad Relación obtenido, visualizando la figura hay puntos específicos que saltan a la vista:

- Las tablas y campos han sido creados en otro idioma: inglés.
- Existen dos tablas satélites. Se conocen como tablas satélites a aquellas que no

están relacionadas con ninguna otra entidad dentro todo el modelo. Más adelante se explicará por qué fueron eliminadas.

2.2 Estandarización de Objetos de Base de Datos

Crear estándares que definan la nomenclatura de objetos de base de datos es una práctica que genera un valor agregado a una aplicación. Si bien es cierto que no usar estándares no merma el rendimiento de un sistema, sí conlleva varios problemas futuros que podrían ser evitados si se los usa; entre ellos:

- Dificultad en el mantenimiento del sistema.
- Demora en comprensión de la estructuración de la base.
- Dificultad en la portabilidad entre motores de base de datos, plataformas y aplicaciones.

Aun sabiendo que los estándares son de gran ayuda, no siempre es posible implementarlos en todo el sistema. Con el afán de no reinventar la rueda sino más bien de hacer uso de lo ya creado, se suele reutilizar módulos desarrollados con anterioridad; esto último es una práctica común en el desarrollo de aplicaciones Web. En este sentido, reutilizar recursos involucra también adaptarse a los estándares usados por los autores de tales módulos, si es que los hubiere. Por tal motivo, modificar desarrollos ajenos para acentuarlos a nuestros estándares, en muchos casos, no es una buena idea. La pérdida de tiempo puede ser considerable y podría repercutir directamente en el cronograma, alcance y costo de un proyecto. Sin embargo, para poder usar un código no propio, hay que tener en consideración varios aspectos, entre ellos:

- Verificar que su licencia nos permita hacer uso de él. Independientemente de si es o no gratuito, las licencias permiten conocer detalles específicos de cómo y hasta dónde hacer uso de un módulo, así como también saber si es necesario hacer un desembolso para su adquisición.
- Asegurarse que el módulo satisface cabalmente nuestras necesidades. Si tuviere más características de las requeridas, y su instalación y uso no deteriora el performance de una aplicación, la opción sería altamente elegible. En el caso contrario, es decir, si no cumple con todas nuestras exigencias y/o merma considerablemente el rendimiento del sistema, la opción sería preferiblemente

descartable.

- En la medida de lo posible, buscar módulos con estándares conocidos y/o fáciles de entender para agilizar la adaptación a los mismos.
- Verificar que los requerimientos de hardware en aspectos referentes a instalación y uso, son satisfechos por la infraestructura tecnológica que soporta nuestro proyecto. De esta manera aseguramos que no decaiga el performance del sistema en construcción.

Exceptuando los módulos foráneos, la estandarización de objetos de base de datos en la migración del sistema de Fichas Médicas del Hogar Corazón de María, ha sido un requisito satisfecho desde el inicio del desarrollo.

2.2.1 Tipo de Tablas

MySql soporta distintas tecnologías para el almacenamiento de datos, entre ellas destacan MyISAM e InnoDB. La principal diferencia entre estos dos tipos de tablas radica en el soporte para integridad referencial presente en InnoDB y ausente en la otra tecnología.

En Internet se suelen encontrar interminables debates sobre cuál de las dos tecnologías es la mejor, sin embargo, existen muchos factores a tener en cuenta y que difieren entre sistema y sistema, por tal motivo, la selección debe basarse en un estudio particular de las ventajas de una tecnología frente a la otra, aterrizando las características de cada una de ellas al sistema a ser desarrollado. Para la migración, objeto de esta tesis, se ha usado la tecnología InnoDB. La selección se basó en los siguientes aspectos:

- Soporte para integridad referencial entre tablas. Este aspecto además de cumplir con su principal objetivo, tiene un valor agregado: El modelo físico y su nivel de detalle visual en lo que respecta a relaciones entre entidades, permite al desarrollador tener una mejor idea de cómo funciona o debería funcionar el sistema. Esto mejora el tiempo de adaptación de nuevos programadores y administradores de base de datos en fases de desarrollo, mantenimiento y también en el caso de migraciones.
- Las sentencias INSERT y UPDATE son tan o más comunes que las SELECT y, una de las ventajas de MyISAM, radica en la rapidez de recuperación de datos

cuando lo más frecuente es realizar sentencias SELECT a la base por encima de escasas solicitudes de inserción y actualización.

- A pesar de que MyISAM era el estándar indiscutible de MySql hasta su versión 4, las cosas parecen haber dado un giro y ahora, lo más popularizado por sus ventajas en almacenamiento de datos, es InnoDB.
- La selección tanto de una como de otra tecnología no era un tema crítico para el performance de la aplicación. Al no ser un sistema empresarial y al estar limitado a poco más de un par de decenas de tablas, la diferencia en el rendimiento por la selección de la tecnología de almacenamiento, es prácticamente imperceptible.

2.2.2 Tablas

MySql es un motor de base de datos, por defecto, sensible a mayúsculas y minúsculas. Por esta razón, entrar en detalles de qué conviene usar más como primera letra de cada palabra – en el caso de que dos o más palabras compusieran el nombre de una tabla – resultaría engorroso y difícil de recordar.

Los estándares usados para nombrar tablas, son los siguientes:

- Todas las palabras que conforman el nombre de la tabla, de principio a fin, deben estar en minúsculas. Esto incluye la primera letra de cada palabra
- Los nombres de cada tabla deben ser auto-descriptivos y, en la medida de la posible, deben evitar abreviatura que compliquen la comprensión de qué es lo que almacena cada objeto.
- Evitar caracteres especiales de cualquier índole, excepto el guión bajo (‘_’) el cual será usado exclusivamente para separar palabras cuando exista la necesidad.
- No se usarán sufijos.

2.2.3 Campos

Los campos han sido nombrados con los mismos estándares usados para tablas. La única diferencia radica en la inclusión de la abreviatura ‘id’ al final del mismo, para referirse a la clave primaria de la tabla a la que pertenece el campo o una clave foránea de una tabla referencial. Es necesario mencionar que las claves primarias suponen el mismo nombre de

la tabla más la abreviatura mencionada al final del nombre. Para separar el nombre del campo y la abreviatura 'id', se ha utilizado el mismo guión bajo usado para separar palabras. Ejemplo: Tabla: 'paciente', clave primaria: 'paciente_id'.

2.2.4 Otros Objetos

Únicamente se han usado 3 restricciones que son consideradas como objetos dentro de una base de datos relacional. Estas restricciones siguen las descripciones especificadas en la siguiente tabla:

Restricción	Estándar
Primary Key	Incluir la abreviatura 'PK' antes del nombre del campo usado como clave primaria. Para separar la abreviatura y el nombre de la restricción, se debe usar un guión bajo. Ejemplo: PK_paciente, donde: • 'PK' es la abreviatura que denota la restricción de tipo clave primaria, y • 'paciente' representa la tabla para la cual existe la restricción. Este es el único caso en el que se incluyen letras mayúsculas dentro de la nomenclatura de un objeto. Su finalidad es enfatizar la restricción más relevante que forma parte de una tabla. **Nota:** Por estándar propio de desarrollo, únicamente se usará un campo como clave primaria de cada tabla. Si bien MySql soporta la combinación de varios campos como PK, esta migración/desarrollo utilizará uno sólo, sin excepción.
Foreign Key	Incluir la abreviatura 'fk' antes del nombre de la restricción. La siguiente parte del nombre está conformada por 1) el nombre de la tabla que hereda (hija) separada por otro guión bajo de 2) el nombre de la tabla padre. Ejemplo: fk_paciente_estado_civil, donde: • 'fk' es la abreviatura que denota la restricción de tipo clave foránea. • 'paciente' representa la tabla hija, y • 'estado_civil' representa la tabla padre.
Unique	Incluir la abreviatura 'uk' antes del nombre de la restricción. La siguiente parte del nombre está conformada por 1) el nombre de la tabla en cuestión y 2) el campo o conjunto de campos que son afectados por la restricción de tipo 'Unique'. Cada palabra debe estar separada por un guión bajo. Ejemplo: uk_paciente_cedula, donde: • 'uk' es la abreviatura que denota la restricción de tipo clave única e irrepetible. • 'paciente' representa la tabla sobre la cual se aplica la restricción, y • 'cedula' es el campo específico al que la estricción afecta.

Tabla 2-1 Estándares de Restricciones [B]

Nótese que internamente el motor de base de datos crea índices basados en las restricciones

de tipo Unique Key (uk) y Foreign Key (fk). Es decir, lo efectivamente creado con esas dos restricciones, son objetos de tipo 'Index'. Debido a que estos objetos usan la nomenclatura indicada en la tabla anterior, los Foreign Keys que realmente se crean son autonombrados por el motor y, aunque siguen un orden y un estándar, éstos no son mencionados en esta sección del documento.

2.3 Modificación del MER Anterior y Generación del Nuevo

A pesar de contar con el Modelo Entidad Relación usado en el sistema anterior, existe la necesidad de dividir ciertas tablas, eliminar otras y crear nuevas entidades. Además, las relaciones que determinan la jerarquización de la información deben ser modificadas desde su parte estructural.

Como ya se mencionó, será InnoDB la tecnología encargada de manejar las tablas de la nueva aplicación. Sin embargo, la adaptación del modelo anterior a esta tecnología requeriría una reestructuración profunda y engorrosa. Es así que, aprovechando que el sistema actual no ha sido utilizado por el Hogar, se han creado las tablas y las relaciones entre ellas, desde cero. De esta manera se sacará ventaja de las ayudas visuales que ofrecen las relaciones entre tablas y la integridad de la información; ambas, características soportadas por InnoDB.

2.3.1 Modificación del MER Anterior

Los cambios en los que se ha incurrido para la generación del nuevo Modelo Entidad Relación, son:

- Eliminación de la tabla 'gender' o 'genero'. Al existir valores determinados e invariables, el género o el sexo de cada paciente debe ser una lista de valores, más no una tabla de datos independiente. Los únicos valores posibles son: Masculino (M) y Femenino (F).
- Eliminación de las tablas 'menu', 'permission', 'role_menu', 'role_permission', 'user' y 'user_role'. El sistema migrado hace uso de un módulo ya desarrollado, denominado SRBAC o Control de Acceso Basado en Roles, por sus siglas en inglés [10]. Este sistema usa otras tablas para guardar información similar a la almacenada

en las tablas eliminadas y garantiza seguridad en los roles asignados a un usuario, las tareas asignadas a un rol y los permisos asignados a una tarea. A nivel de base de datos, éste es el único módulo que, por ser desarrollado por terceros, no respeta los estándares mencionados en secciones anteriores. Las tablas creadas para almacenar información del módulo SRBAC, son:

  - asignacion
  - autorizacion
  - item
  - usuario

  Más adelante, en el diccionario de datos, se dará a conocer detalladamente la estructura de las nuevas tablas y, en la figura del nuevo Modelo Entidad Relación, se visualizará la relación existente entre cada una de ellas.

- Eliminación de la tabla 'medic'. Esta tabla estaba directamente relacionada con la tabla 'patient' o 'paciente' y, en la nueva estructura, la relación secuencial es la siguiente: paciente → consulta → diagnostico → evolucion, siendo esta última tabla la que reemplazaría, por campos y tipos, a la tabla 'medic'.
- División de la tabla 'patient' en dos: 1) tabla 'paciente' que almacenará únicamente datos generales del adulto mayor y 2) tabla 'consulta' que contendrá todo el resto de información relativa a la consulta médica generalizada por medio del Ministerio de Salud Pública del Ecuador.
- Eliminación de la tabla 'physiotherapy'. Los datos de terapia física y psicológica serán almacenados en la tabla 'evolucion' y requerirán de una consulta y un diagnóstico previos.
- Eliminación de la tabla 'sequence'. Esta tabla era usada para almacenar las secuencias siguientes de las claves primarias de cada una de las tablas. En el nuevo modelo, al haberse declarado cada campo de clave primaria como 'auto-increment', no hay necesidad de mantener esta secuencia a nivel de una tabla; es el motor de base de datos quien se encarga de asignar el valor correspondiente.
- Eliminación de la tabla 'social_work'. Al no ser parte de los requerimientos funcionales, la tabla que almacenaba datos de acción social ha sido eliminada. Sin embargo, como recomendación de este proyecto de tesis se sugiere la re-inclusión de la misma.
- Eliminación de la tabla 'state'. Los dos únicos posibles estados de un paciente, son:

Activo e Inactivo. No hay necesidad de una tabla para gestionar estos 2 valores, difícilmente variables.

- Eliminación de la tabla 'track'. Los requerimientos funcionales no incluyen el almacenamiento de pistas de auditoria. No obstante, esta funcionalidad es parte de las recomendaciones finales.
- Eliminación de la tabla 'type'. Esta tabla contenía los siguientes valores:
    - Discapacitado
    - Enfermedad Crónica
    - Tercera Edad
    - Bono de Desarrollo Humano
    - Otro

  Las opciones eran parte de una lista de selección individual de la tabla 'patient'. Su eliminación se debe a que los valores no son excluyentes el uno del otro. Es decir, un mismo paciente puede ser 'Discapacitado' y, a la vez, de 'Tercera Edad'.

2.3.2 Generación del Nuevo MER

Para satisfacer los nuevos requerimientos del Hogar, además de eliminar tablas no útiles para las necesidades reales, también ha sido necesario crear nuevas. El listado completo de tablas se muestra, en orden alfabético, en la siguiente figura.

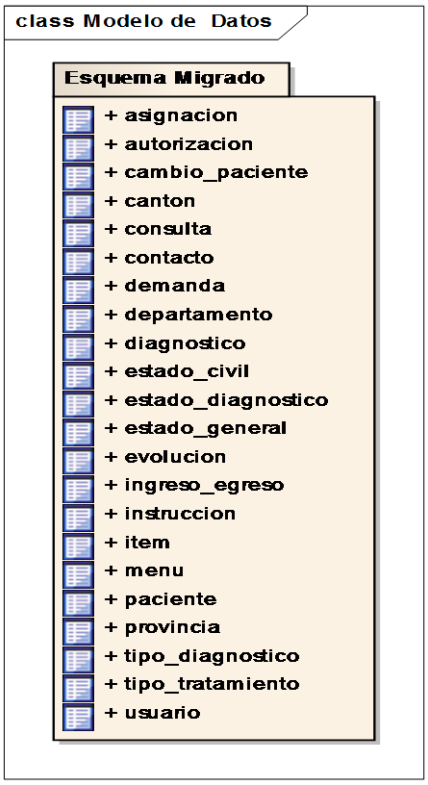


Ilustración 2-3 Listado de Tablas del Sistema Migrado [B]

Aunque ciertos nombres de tablas se comportan con el Modelo Entidad Relación actual, la sintaxis y semántica de cada tabla se la evaluó desde cero y, si bien es cierto que varias tablas no cambiaron, no es menos real que aquellas tablas fueron las menos relevantes en el sistema.

A pesar de procurar usar nombres auto-descriptivos como se indicó en la sección 2.2.2, un simple listado no da una idea clara de cuál es la relación existente entre las tablas. La siguiente figura muestra el modelo conceptual con todas las adecuaciones necesarias para satisfacer los requerimientos.

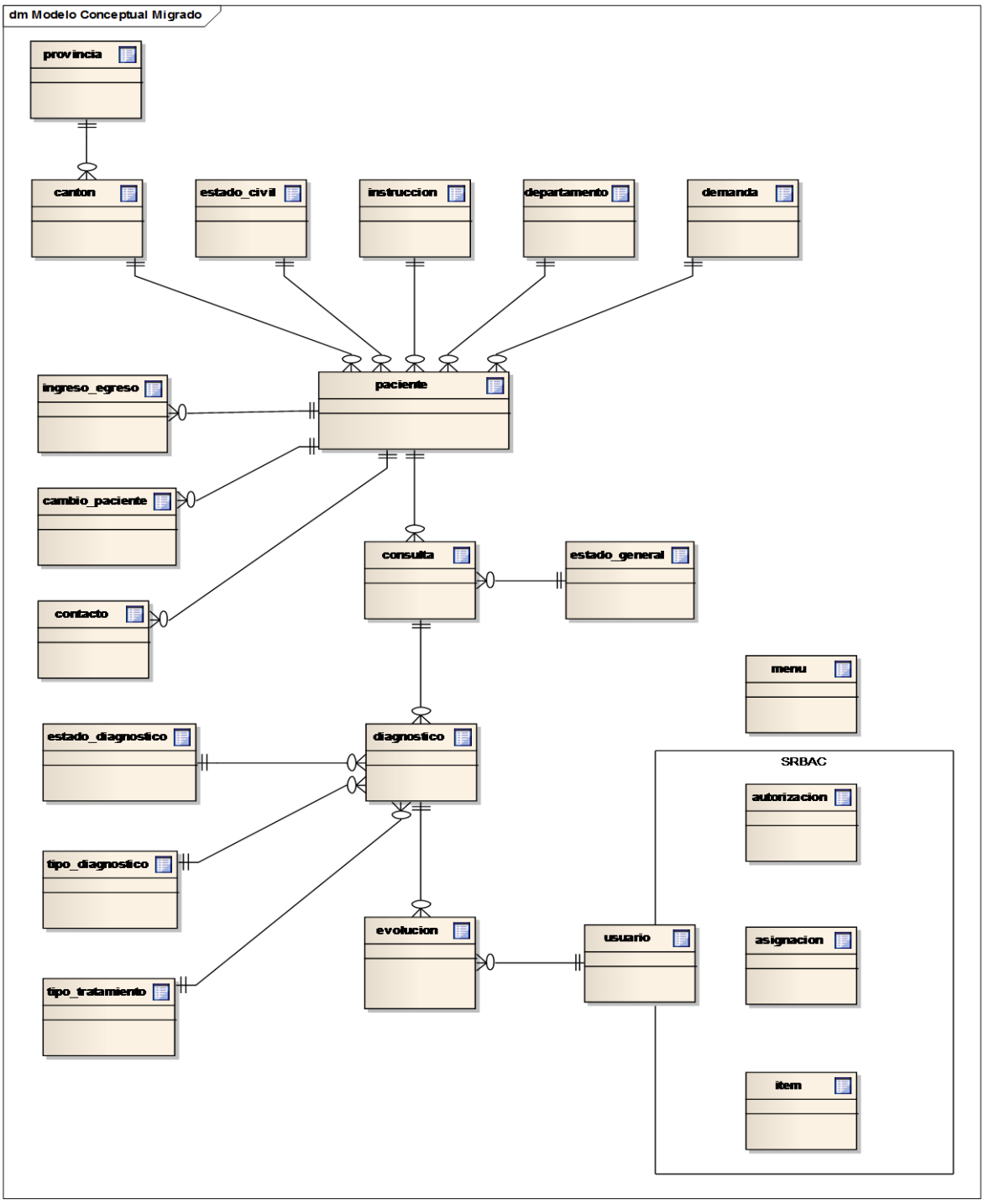


Ilustración 2-4 MER del Sistema Migrado [B]

La figura muestra el modelo conceptual por sobre el físico debido a la cantidad de atributos presentes, en esta ocasión, en la tabla ‘consulta’, donde reside el core del sistema migrado.

El modelo presenta un par de particularidades que se aprecian a simple vista:

- La tabla satélite ‘menu’, además de no tener relaciones físicas con ninguna otra

tabla, tampoco tiene relaciones lógicas. Es decir, su denominación como ‘satélite’ es ideal. Está tabla contiene las opciones disponibles dentro del sistema y fue *bajada* a nivel de base de datos para evitar que se quemaran dichas opciones directamente en la programación.

- El resto de tablas satélites están enmarcadas dentro de un limitador denominado SRBAC, y corresponden a un módulo externo que permite gestionar el control de accesos basados en roles. Debido a que no es un desarrollo propio, no respeta los estándares definidos. Nótese que la tabla ‘usuario’ está presente *a medias* en esta sección; esto se debe a que la interfaz entre las 3 tablas nativas del módulo y el resto del sistema, requiere de una entidad que almacene los usuarios y que sirva como *puente*.

2.3.2.1 Diccionario de Datos

El diccionario de datos es un modelo que complementa lo visualizado y comprendido en un modelo conceptual, lógico o incluso físico, de una base de datos. Los detalles incluidos en este diccionario no son mencionados en otros modelos.

El alcance de detalle llegará a las tablas, de manera que se excluyen descripciones de cada campo. Esta particularidad se debe al gran número de campos existentes en la tabla ‘consulta’, a saber, 172 atributos.

La siguiente tabla menciona datos relevantes de relaciones existentes entre entidades de base de datos, e incluye una descripción de cada una de ellas:

Tabla	Restricciones y Relaciones	Descripción
asignacion	**PK**: itenname, userid	Guarda relaciones de los usuarios del sistema (tabla 'usuarios') con los roles almacenados aquí. Esta es una de las tablas del módulo SRBAC.
autorización	**PK**: parent, child **FK**: autorizacion(parent), autorizacion(child)	Almacena la jerarquización de todos los ítems (tabla 'item') del módulo SRBAC. Además de guardar el nombre del ítem ('child'), también almacena el nombre de su padre ('parent').
cambio_paciente	**PK**: cambio_paciente_id **FK**: paciente(paciente_id) = cambio_paciente(paciente_id)	Registra los cambios efectuados sobre los datos generales de cada paciente.

canton	**PK**: candon_id **FK**: provincia(provincia_id) = canton(provincia_id), canton(canton_id) = paciente(canton_id)	Listado completo de todos los cantones nombrados como tales por las distintas jurisdicciones políticas del Ecuador. Cada paciente proviene de un cantón que, a su vez, pertenece a una provincia.
consulta	**PK**: consulta_id **FK**: paciente(paciente_id) = consulta(consulta_id), estado_general(estado_general_id) = consulta(estado_general_id), consulta(consulta_id) = diagnostico(consulta_id)	Mantiene los registros de todos los datos de la ficha médica del paciente. Esta tabla contiene la información necesaria para obtener el reporte HCU-form.057/2010, exigido por los ministerios MIES y MSP.
contacto	**PK**: contacto_id **FK**: paciente(paciente_id) = contacto(paciente_id)	Cada paciente puede tener n contactos. Aquí se registran aquellas personas relacionadas de alguna manera con el paciente y que sirven como contacto o referencia.
demanda	**PK**: demanda_id **FK**: demanda(demanda_id) = paciente(demanda_id)	Cada paciente ingresa al hogar bajo algún tipo de demanda. Entre las más conocidas están: Demanda Institucional y Particular.

		Otros tipos de demanda pueden ser incluidos en esta tabla.
departamento	**PK**: departamento_id **FK**: departamento(departamento_id) = paciente(departamento_id)	El Hogar Corazón de María cuenta con varios departamentos en los que se clasifican a los pacientes. Esta tabla almacena tal distribución.
diagnostico	**PK**: diagnostico_id **FK**: consulta(consulta_id) = diagnostico(consulta_id), tipo_diagnostico(tipo_diagnostico_id) = diagnostico(tipo_diagnostico_id), estado_diagnostico(estado_diagnostico_id) = diagnostico(estado_diagnostico_id), tipo_tratamiento(tipo_tratamiento_id) = diagnostico(tipo_tratatamiento_id), diagnostico(diagnostico_id) = evolución(diagnostico_id)	Después de la debida consulta médica (tabla 'consulta'), el paciente debe recibir un diagnóstico y tratamiento. Es en esta tabla donde se almacena toda aquella información. La mayoría de sus datos también forman parte del reporte HCU-form.057/2010,
estado_civil	**PK**: estado_civil_id **FK**: estado_civil(estado_civil_id) = paciente(estado_civil_id)	Listado de estados civiles válidos para cada paciente.
estado_diagnostico	**PK**: estado_diagnostico_id **FK**: estado_diagnostico(estado_diagnostico_id) = diagnostico(estado_diagnostico_id)	Listado de estados en lo que se puede categorizar el diagnóstico médico. Ejemplo: Presuntivo o Definitivo.
estado_general	**PK**: estado_general_id **FK**: estado_general(estado_general_id) = consulta(estado_general_id)	Listado de posibles estados del paciente. Entre los más relevantes figuran: Independiente, Frágil y Dependiente. Este

		dato es parte de la ficha médica.
evolucion	**PK**: evolucion_id **FK**: diagnostico(diagnostico_id) = consulta(diagnostico_id), usuario(usuario_id) = consulta(usuario_id)	La tabla 'diagnostico' incluye datos de tratamiento, cada tratamiento puede tener n registros de evolución. Tales registros son almacenados en esta tabla.
ingreso_egreso	**PK**: ingreso_egreso_id **FK**: paciente(paciente_id) = ingreso_egeso(paciente_id)	Un paciente puede abandonar el hogar y reingresar al mismo. Estas transacciones se guardan en la tabla 'ingreso_egreso'.
instruccion	**PK**: instruccion_id **FK**: instruccion(instruccion_id) = paciente(instruccion_id)	Como parte de los datos generales del paciente, se incluye su instrucción educacional. Los posibles valores de tal instrucción son almacenados en esta tabla.
item	**PK**: name	Contiene todos los ítems del módulo SRBAC. Estos ítems incluyen roles, permisos y tareas.
menu	**PK**: menu_id	Para fines de interfaz de usuario, es necesario guardar el listado de opciones disponibles. Esta tabla es usada para ese objetivo: almacenar las opciones del sistema.
paciente	**PK**: paciente_id	La tabla 'paciente' contiene los datos

	FK: canton(canton_id) = paciente(canton_id), estado_civil(estado_civil_id) = paciente(estado_civil_id), demanda(demanda_id) = paciente(demanda_id), departamento(departamento_id) = paciente(departamento_id), instrucción(instruccion_id) = paciente(instruccion_id), paciente(paciente_id) = cambio_paciente(paciente_id) paciente(paciente_id) = consulta(paciente_id) paciente(paciente_id) = contacto(paciente_id) paciente(paciente_id) = ingreso_egreso(paciente_id) **UK**: registro, cedula	generales de cada adulto mayor. Esta entidad representa el core del sistema, referente a la información relevante de los residentes del Hogar.
provincia	**PK**: provincia_id **FK**: provincia_id(provincia_id) = canton(provincia_id)	Listado general de provincias del Ecuador. Cada paciente pertenece a una provincia y cantón específicos. Debido a que ya existe una relación física entre provincia y cantón, y otra entre cantón y paciente, no es necesario incluir una directa entre provincia y paciente.
tipo_diagnostico	**PK**: tipo_diagnostico_id **FK**: tipo_diagnostico(tipo_diagnostico_id) = diagnostico(tipo_diagnostico_id)	Listado de tipos de diagnósticos posibles. Ejemplo: Clínico, Sindrómico, Psicológico, etc.
tipo_tratamiento	**PK**: tipo_tratamiento_id	Listado de tipos de tratamientos a los que

	FK: tipo_tratamiento(tipo_tratamiento_id) = diagnostico(tipo_tratamiento_id)	somete al paciente. Ejemplo: Medicina General, Medicina Alternativa, etc.
usuario	**PK**: usuario_id **FK**: usuario_id(usuario_id) = evolucion(usuario_id)	Listado de Usuarios del Sistema.

Tabla 2-2 Diccionario de Datos del Sistema Migrado (Tablas) [B]

La columna denominada ‘Restricciones y Relaciones’, incluye tres tipos de información: Claves Primarias (PK), Claves Foráneas (FK) y Claves Únicas (UK). En el caso de PK y UK, únicamente se referencian los campos que forman parte de tal restricción. Sin embargo, para el caso de claves foráneas, es necesario entender lo siguiente:

- Al lado izquierdo del signo igual (=) se ubica la tabla padre, es decir, la tabla que hereda sus valores. Seguido de esta tabla y entre paréntesis, aparece el o los campos que son claves primarias en aquella tabla y que son pasados a la tabla hija.
- Al lado derecho del signo igual (=) se ubica la tabla hija, es decir, la tabla que recibe los valores de la tabla padre. Después del nombre de esta tabla y entre paréntesis, aparece el o los campos en los que se copian los valores que conforman la clave primaria de la tabla padre y que, por tanto, son los valores conocidos como FK, en la tabla hija.
- Si aplica, se mencionan primero las relaciones en las que tabla en cuestión es hija, y luego, en la que es padre.

3 DESARROLLO

Con las facilidades que presentan los CMSs actuales, la creación de sitios Web personalizados puede ser realizada por cualquier persona o grupo de personas con conocimiento mínimos de programación y, dependiendo del alcance de esa personalización, también de base de datos. No hay dudas de que la orientación hoy en día, en términos de desarrollo de software, es la creación de aplicaciones que puedan ser usadas a través de un browser, ya sea desde una computadora o desde un dispositivo móvil. Este tipo de desarrollo es justamente el fuerte de programadores novatos. Sin embargo, las aplicaciones creadas por tales programadores normalmente son realizadas de manera empírica y no se benefician de las bondades de seguir los procesos de la ingeniería de software. Las siguientes secciones muestran detalles que se han tenido en cuenta para realizar la migración del sistema.

3.1 Ciclo de Vida

Generalmente un desarrollo de software se realiza en etapas. Estas etapas o fases, en conjunto, son conocidas como Ciclo de Vida del proyecto. Existen varios modelos de Ciclos de Vida y probablemente el más difundido y usado sea el denominado 'En Cascada'. Independientemente del modelo, las fases suelen ser las siguientes:

- Definición de Objetivos
- Análisis de Requisitos y su viabilidad
- Diseño General
- Diseño en Detalle
- Programación
- Prueba de Unidad
- Integración
- Validación
- Documentación
- Implementación
- Mantenimiento

Claro está, dependiendo del proyecto, en ocasiones este listado puede diferir, sin embargo las fases mencionadas son las más aceptas en el mundo de desarrollo de aplicaciones de software.

Considerando que el presente proyecto pretende migrar una aplicación ya culminada al 100%, se deduce que hay etapas que no necesitan ser mencionadas en detalle nuevamente. Es más, los requerimientos originales, su estudio de viabilidad; los diseños general y detallado de la solución informática, entre otras etapas, ya han sido motivo de análisis de otro proyecto de tesis.

La metodología usada para el nuevo desarrollo es la que conoce como 'Extreme Programming' o XP, cuyo enfoque difiere con el de la ingeniería de software típica. El objetivo principal de XP es abastecer de agilidad a un proyecto que tiene como fin ser corto en tamaño y en tiempo de desarrollo. La metodología usada pretende obtener y mantener los siguientes valores:

1. **Simplicidad**: permitir que el diseño sea lo más simple posible para poder agilizar el desarrollo y facilitar el mantenimiento.
2. **Comunicación**: simplificar el código para que sea fácilmente reutilizable en otras partes del sistema y de fácil adopción por otros programadores. En el caso del proceso de migración, el proyecto entero estuvo a cargo de una sola persona.
3. **Retroalimentación**: Involucrar al cliente en el proyecto para que estime, pruebe, analice y, en muchos casos, gestione el avance del proyecto.
4. **Valentía**: Coraje para eliminar el desarrollo burocrático que demora la obtención de resultados y también valentía para desechar artefactos y modelos no útiles u obsoletos e incluir mejoras en su lugar.
5. **Respeto**: respeto por el trabajo propio, al hacerlo con la más alta calidad posible y respeto por el trabajo ajeno, al intentar no inmiscuirse más allá de lo necesario en la labor realizada por terceros.

La metodología XP no busca delimitar fases específicas en el desarrollo y hacer un análisis engorroso de cada una, sino más bien pretende colaborar con el equipo de trabajo para que se realicen las tareas más relevantes a la brevedad posible, que cada una se auto documente y que, después de probarse, se integre al resto del sistema. Esto no quiere decir que se

obvien fases como la de diseño, donde se muestran prototipos de cómo lucirán las interfaces gráficas de usuario (GUI), sino que evita darle una importancia tal que demore el avance del proyecto y que, probablemente vayan quedando inútiles en fases de mantenimiento. XP ha aceptado la idea de que los requerimientos no pueden ser totalmente levantados en etapas tempranas y que, limitarse a éstos, podría resultar algo negativo para el proyecto y para el sistema como tal. Según Extreme Programming, dejar los cambios de requerimientos para fases posteriores a la programación, conlleva una pérdida valiosísima de tiempo. Se concluye esto a partir del siguiente razonamiento lógico: Si de todas maneras un requerimiento funcional ha cambiado en el trascurso del desarrollo del proyecto, ¿por qué seguir avanzado y construyendo más características sobre un requerimiento del cual se conoce que, una vez usado, no satisfará las necesidades del usuario?

Uno de los fuertes de XP radica en la fase de pruebas. Al hacer del cliente/usuario final parte del equipo de desarrollo, las pruebas se realizan tan pronto se culmina un desarrollo específico. Además, para el caso particular de esta migración, la utilización de PphUnit, ha facilitado enormemente la tarea del testing técnico.

3.2 Aspectos Introductorios del Proceso de Migración

Los objetivos principales del proceso de migración de un sistema informático, una vez culminadas todas las fases del ciclo de vida, son:

- Mantener las características anteriores de la aplicación – especialmente aquellas que el usuario considera útiles.
- Cumplir con los requerimientos que dieron origen a la decisión de realizar el trabajo de migración.

Otros objetivos pudieran parecer obvios pero existen atenuantes para ellos. Tomemos por ejemplo el objetivo, casi obvio, de mejorar el performance de una aplicación. Cuando hablamos de una migración de software, casi instantáneamente pensamos en nuevas tecnologías con mayores prestaciones y rendimiento, sin embargo, una migración exitosa depende de los objetivos planteados al inicio de tal proyecto. En el caso del ejemplo tomado, probablemente no sea cierto que se mejore el performance en una migración cuyo principal, o tal vez único interés, es adaptarse a una tecnología X que tenga mayor

capacidad de integración que la actual pero con fuentes de datos más antiguas en las que reside una cantidad no despreciable de datos relevantes en los procesos de la organización. Tal vez la tecnología a la que se migre sea más antigua que la actual y por tanto, menos eficiente; pero si se cumplió el objetivo de permitir la integración con aquellas fuentes de datos, la migración – que conlleva mantener todas las demás características útiles del sistema – ha sido un éxito.

El ejemplo anterior no indica que ese sea el caso del proyecto actual, más bien procura puntualizar el objetivo real de un proceso de migración y, basándonos en tal objetivo, comprendemos que tampoco es verdad que una migración informática sea posible únicamente mediante herramientas automáticas que migren lo ya hecho en un lenguaje de programación, CMS, Web Service, base de datos, etc. hacía una tecnología distinta.

En el caso particular de la migración que es motivo de este proyecto de tesis y en términos exclusivos de programación, se decidió no hacer uso de lo ya desarrollado para el sistema anterior. Entre los principales motivos para tomar esta decisión, están los siguientes:

1. Las características principales del sistema original, por requerimiento, necesitan ser modificadas. Una migración automática mantendría la funcionalidad ya no deseada.
2. Las herramientas que automáticamente generan código en un lenguaje de programación A, a partir de otro B, además – por supuesto – de cumplir con su objetivo, también liberan gran cantidad de código *basura*. Este código difícil de entender y sensible a modificaciones, dificulta grandemente el mantenimiento y soporte sobre la aplicación.
3. Cada herramienta de migración contiene uno o varios *motores* encargados de realizar las distintas tareas. Estos motores también han sido programados y respetan reglas que, a pesar de ser usualmente aceptadas y estandarizadas, casi con certeza resultarán ajenas a los propios estándares de desarrollo. Temas como la estructura y distribución de clases y el número y tipo de capas usadas, pueden resultar tan engorrosos que posiblemente hagan del mantenimiento una tarea complicada y difícil de llevar.

Aunque existen otras razones, las mencionadas aquí son las más importantes y el orden en el que han sido mencionadas también indica su relevancia para este proyecto.

El desarrollo de la nueva aplicación que conlleva la migración del Sistema de Fichas Médicas del Hogar Corazón de María, ha sido realizado desde cero, utilizado el lenguaje de programación PHP, en su versión 5.3.

3.3 Modelo Vista Controlador y Framework

Modelo Vista Controlador o MVC por sus siglas, es un patrón de la arquitectura de software que tiene como propósito el distribuir de manera lógica los componentes de una aplicación. Esta distribución se la realiza en capas y cada componente reside en una de ellas [11].

Componente	Descripción
Modelo	Capa que contiene la representación específica de la información, es decir, los datos, su tipo, estructura y las operaciones referentes a ellos.
Vista	Representa la interfaz gráfica de usuario. En aplicaciones Web, donde más comúnmente se usa MVC, esta capa resulta en páginas HTML, una vez interpretado el código.
Controlador	La capa 'controlador' gestiona las solicitudes realizadas por el usuario desde la capa 'vista' y, a su vez, solicita la acción de la capa 'modelo' para satisfacerlas. Resumiendo, el controlador sirve de puente para administración de solicitudes entre las otras 2 capas del patrón MVC.

Tabla 3-1 Modelo Vista Controlador [B]

Este patrón de trabajo ha sido adoptado para el desarrollo del nuevo sistema por las ventajas que presenta, entre ellas:

- Se adapta perfectamente a la programación orientada a objetos. Inclusive, algunos programadores experimentados lo encasillan exclusivamente dentro de este tipo de desarrollo [D].
- Distribución y separación adecuada de interfaces de usuario, lógica del negocio y entes controladores de acciones.
- Reutilización de componentes.
- Ambiente de producción fácil de mantener en fases post implementación.

Si bien existen detractores de este patrón que argumentan que el número de archivos que demanda la utilización de MVC es muy alto, la experiencia en el desarrollo y mantenimiento de aplicaciones Web, indican que esta supuesta *desventaja* resulta ser todo lo contrario cuando, por ejemplo, tal distribución permite la alta escalabilidad del sistema.

El diagrama de secuencias utilizado en MVC es explicado de manera gráfica en la siguiente figura:

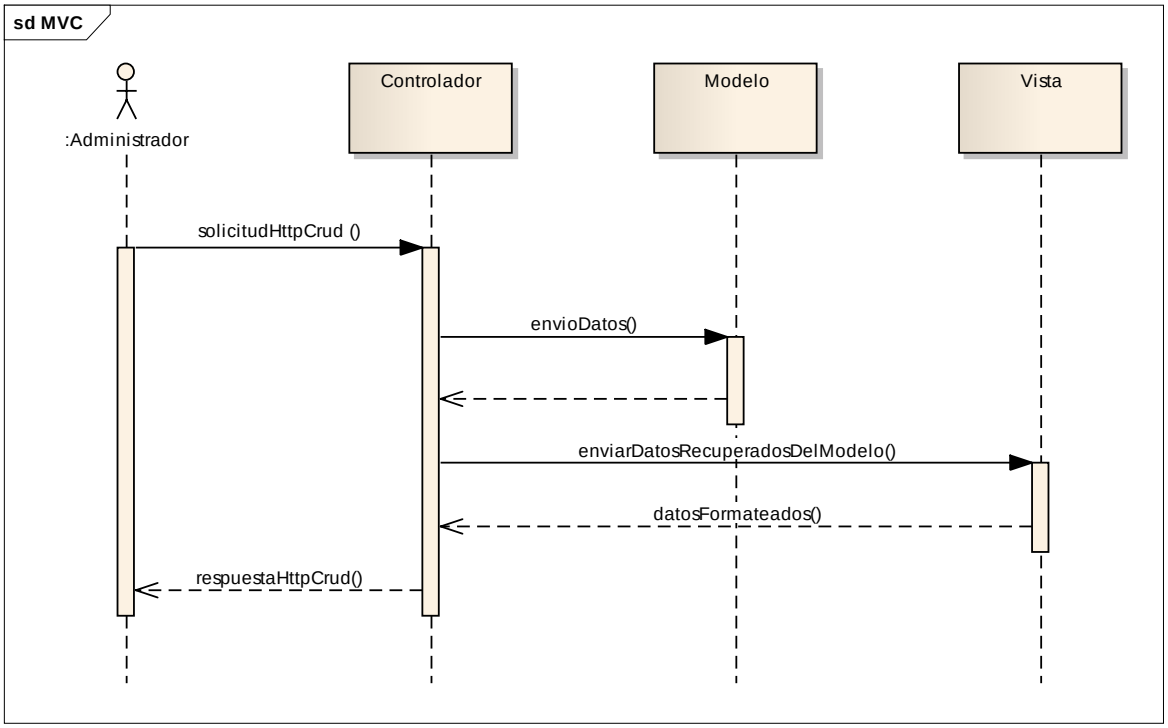


Ilustración 3-1 Diagrama de Secuencia de MVC [B]

En PHP existen decenas de frameworks o marcos de trabajo que se adaptan a MVC. Antes de entrar en detalles sobre la selección del framework, es oportuno mencionar qué exactamente representa un marco de trabajo. Un framework es una estructura lógica y física de módulos y librerías que definen claramente cómo un nuevo proyecto de desarrollo de software puede ser construido. La tarea de desarrollo es altamente optimizada gracias a la ayuda de los componentes del marco de trabajo y al basarse en este, todas sus características son heredadas en el nuevo proyecto.

La selección del framework para el desarrollo del sistema estuvo orientada a la utilización de un marco de trabajo conocido, probado y con resultados garantizados. En cumplimiento de estos aspectos, las opciones más probables eran Zend y Cake. Sin embargo, un framework relativamente nuevo denominado Yii, llamó poderosamente la atención por las ventajas que presenta, entre ellas [12]:

- Generación automática de componentes CRUD (Siglas que representan a las sentencias de tipo DML de base de datos, en el siguiente orden: Create o insert, Read o select, Update y Delete).
- Generación automática de todas las capas de MVC.
- Alta legibilidad del código generado.

Como es natural en cualquier herramienta, Yii también tiene desventajas. No obstante, la ayuda que presenta al generar código repetitivo pero definitivamente necesario, supera con creces cualquier punto en contra. Es así como Yii se convirtió en el marco de trabajo seleccionado para el desarrollo de la nueva aplicación.

3.4 Preparación del Ambiente de Desarrollo

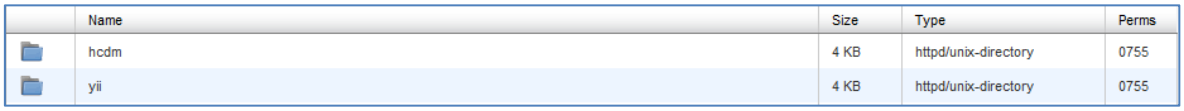
Como se ha mencionado anteriormente, el lenguaje de programación en el que se ha desarrollado el proceso de migración del sistema, es PHP; la base de datos es MySql y, casi por inercia, se puede deducir que el Web server, es Apache.

La instalación por separado de estos componentes podría resultar complicada y, por tanto, costosa en términos de tiempo. Ventajosamente existen herramientas que integran todo en un solo instalador; las más populares, sin duda, son:

- WAMPP, para Windows
- XAMPP, multiplataforma
- MAMPP, para Mac

La migración del sistema pretende que, una vez terminado el proceso de desarrollo, la nueva aplicación resida sobre Linux. Por tal razón, la herramienta seleccionada para conseguir la integración de tecnologías en el ambiente de desarrollo, fue XAMPP [13].

Una vez instalado XAMPP, encendidos todos sus servicios y creada una base de datos que contendrá los objetos representados en el Modelo Entidad Relación, es conveniente indicar cómo se ha definido la estructura física de trabajo. En el archivo de configuración de Apache se señala específicamente el directorio que representará la estructura de directorios a la que apunta el Web server para tomarlo como raíz. En esta ubicación, pueden existen n sub directorios y cada uno de ellos es considerado como un proyecto independiente. Justamente en ese nivel se encuentran las carpetas 'hcdm', que contiene el proyecto como tal y 'yii', el marco de trabajo. Este último directorio es obtenido al descargar la versión más reciente del framework Yii y descomprimir el archivo en la ubicación indicada. Es probable que, para que la carpeta se llame únicamente 'yii', sea necesario renombrarla; usualmente su nombre incluye también la versión del marco de trabajo.



	Name	Size	Type	Perms
	hcdm	4 KB	httpd/unix-directory	0755
	yii	4 KB	httpd/unix-directory	0755

Ilustración 3-2 Directorio Raíz de Apache [B]

Si el directorio 'yii' es eliminado, el proyecto dentro de 'hcdm' quedará inutilizable debido a que todas las librerías necesarias para soportar la aplicación están dentro de la carpeta que contiene el framework. Los estándares propios de Yii recomiendan colocar el marco de trabajo en este nivel, de tal manera que si existen 2 o más proyectos, todos ellos usarán las mismas librerías y, en sus archivos de configuración, apuntarán a un mismo destino.

Cabe mencionar que el directorio 'hcdm' fue generado por el propio framework al hacer uso de un utilitario conocido como 'yiic' [14]. Esta generación de la aplicación Web, en realidad un directorio ubicado dentro de la raíz del servidor Web – a la que se le puede dar el nombre que queramos, en nuestro caso 'hcdm' – contiene una estructura definida de carpetas y archivos. En el primer nivel interior se encuentran varios sub directorios, donde la más relevante, en términos de desarrollo, es el denominado 'protected'; dentro de éste se encuentran todos los archivos modificables que componen el sistema, así como la distribución física de las capas de MVC.

Para generar la aplicación 'hcdm' se usó la siguiente sentencia en la línea de comandos de Linux:

```
/opt/lampp/htdocs/yii/framework/yiic Webapp WebRoot/hcdm
```

Donde '/opt/lampp/htdocs/yii/framework/yiic Webapp' es la ubicación exacta del utilitario que genera proyectos con la estructura predefinida y 'hcdm' es el nombre que se le da al nuevo proyecto generado dentro del directorio raíz del servidor Web.

Para utilizar este recurso del framework en Linux, es necesario otorgar permisos totales (rwx-rwx-rwx o 777) al directorio del marco de trabajo. Una vez terminado todo el proceso de configuración se recomienda restablecer los permisos originales.

Parte esencial de la configuración del ambiente es asegurar la conexión a la base de datos. Tomando al directorio raíz de aplicaciones Web como el inicio del path (/), encontramos el archivo de configuración en esta dirección: /hcdm/protected/config/main.php. La siguiente figura muestra las líneas que componen la configuración necesaria para acceder a la base de datos:

```php
<?php
...
'db' => array(
        'connectionString' => 'mysql:host=<<<<IP DEL SERVIDOR DE BASE DE DATOS>>>>;dbname=<<<<NOMBRE DE LA BASE>>>>',
        'emulatePrepare' => true,
        'username' => '<<<<USUARIO ADMINISTRADOR>>>>',
        'password' => '<<<<PASSWORD>>>>',
        'charset' => 'utf8',
        'class' => 'CDbConnection',
    ),
...
?>
```

Ilustración 3-3 Configuración de Acceso a la Base de Datos [B]

Donde los datos en mayúsculas y entre signos de menor y mayor que (<<<< >>>>) tienen que ser cambiados con los datos reales. Nótese que estas indicaciones son ciertas para el resto de figuras que contengan líneas de código del archivo de configuración del proyecto.

Las secciones 'Generación de Modelos' y 'Generación de Controladores y Vistas' harán uso de otro recurso del framework. Este recurso es conocido como Gii, una interfaz gráfica del marco de trabajos cuyo objetivo es la generación automática de código. Para habilitar su uso, el archivo de configuración – en la sección Gii – debe lucir tal y como muestra la

siguiente figura:

```php
<?php
...
'gii' => array(
            'class' => 'system.gii.GiiModule',
            'password' => '<<<<PASSWORD>>>>',
            'ipFilters' => array('127.0.0.1', '::1'),
        ),
...
?>
```

Ilustración 3-4 Habilitación de Gii [B]

Al ser Gii un recurso capaz de realizar acciones de alta sensibilidad como sobre-escritura de archivos, existe el parámetro 'ipFilters' que se encarga de otorgar permisos para ejecutar estas operaciones únicamente a las IPs listadas en el arreglo respectivo. El parámetro 'password' está quemado en este archivo y el usuario que posteriormente use la interfaz Gii, requerirá ingresar esta clave para poder hacer uso del módulo de generación automática de código.

Las siguientes figuras muestran las líneas de código requeridas para dar nombre a la aplicación, traducirla al español (originalmente del inglés de los Estados Unidos), ingresar un email de administrador y configurar el manejo estándar de URLs, que por defecto, son gestionadas de una manera distinta en Yii. Esa gestión incluye la forma en cómo éstas son visualizadas por el usuario final.

```php
<?php
...
'name' => 'Hogar "Corazón de María"',
'language' => 'es',
'sourceLanguage' => 'en',
...
'params' => array(
        'adminEmail' => 'usuario_administrador@dominio.com',
    ),
...
?>
```

Ilustración 3-5 Datos Generales de la Aplicación [B]

```php
<?php
...
'urlManager' => array(
            'urlFormat' => 'path',
            'rules' => array(
                '<controller:\w+>/<id:\d+>' => '<controller>/view',
                '<controller:\w+>/<action:\w+>/<id:\d+>' => '<controller>/<action>',
                '<controller:\w+>/<action:\w+>' => '<controller>/<action>',
            ),
        ),
...
?>
```

Ilustración 3-6 Estandarización de URLs [B]

Con estos cambios preliminares en el archivo de configuración, ya existe una aplicación Web lista para ser usada. Secciones posteriores requerirán de más cambios en este fichero.

Para acceder a la aplicación es necesario ir al localhost (http://localhost), ver el listado de proyectos existentes, y seleccionar 'hcdm'.

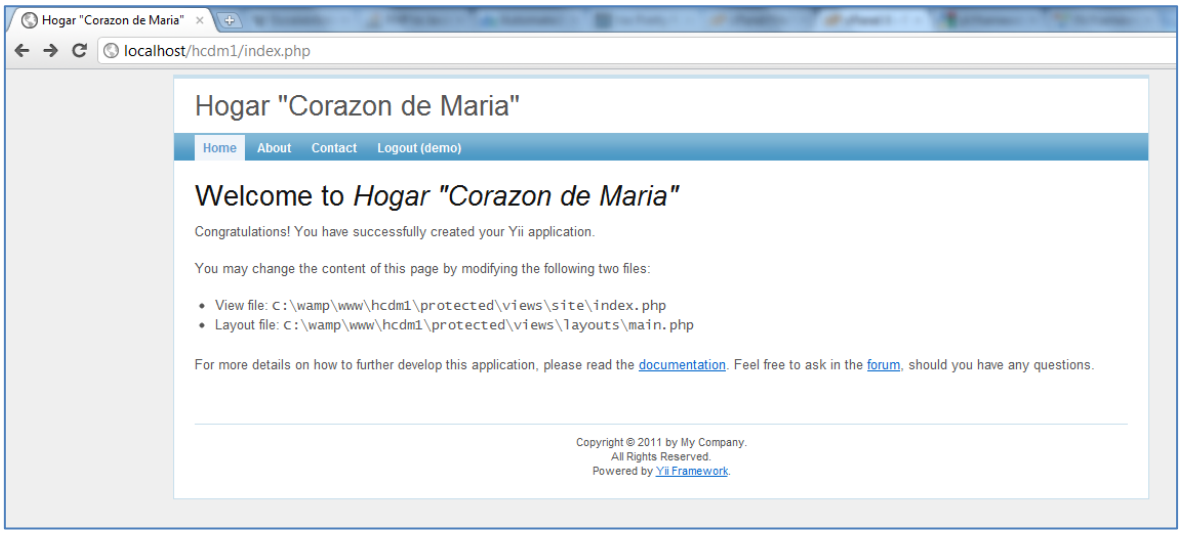


Ilustración 3-7 Aplicación Web Generada [B]

Nótese que la traducción del inglés al español no es perceptible en la figura. Esto se debe a que la traducción no se aplica sobre toda la aplicación sino sobre los links y botones generados con posterioridad.

3.5 Generación de Modelos

De acuerdo a la explicación sobre lo que un Modelo representa – descrita en la tabla 'Modelo Vista Controlador' – conocemos que la estructuración de la información es gestionada por este componente del patrón MVC y entrando en detalle sobre lo que esto significa, entendemos que dicha estructuración puede referirse al modelo de base de datos. De hecho, el Modelo frecuentemente contiene la estructura del MER a nivel del lenguaje de programación, en este caso PHP. Cada entidad o tabla del Modelo Entidad Relación corresponde a una clase del Modelo, en MVC. Los campos de la tabla se convierten en atributos de la clase, mientras que las relaciones entre entidades, llegan a ser métodos. Además, existen funciones adicionales que son creadas a nivel del Modelo y que son usadas para diversos fines, entre ellos, asociar atributos a etiquetas fácilmente legibles por los usuarios finales. A manera de recordatorio vale mencionar que 'funciones' y 'métodos' son sinónimos al hablar de una clase, en programación orientada a objetos.

Usualmente se escribe el código que permite afirmar la correspondencia entre las tablas y clases mencionadas, sin embargo, Yii lo genera automáticamente. Gracias a la adecuada configuración del archivo que contiene los parámetros de conexión a la base de datos y después del llenado de un par de formularios, podemos obtener automáticamente las clases de cada una de las tablas del MER del nuevo sistema.

Para acceder a la interfaz de generación de modelos es necesario seguir los siguientes pasos:

1. Ir a http://localhost/hcdm/index.php/gii
2. Ingresar la clave de acceso al módulo (Ver Figura 'Habilitación de Gii').
3. Seleccionar la opción 'Model Generator'
4. En el campo 'Table Name', ingresar uno por uno el nombre de cada tabla (opción recomendada, si no se es un desarrollador experto en el uso de Yii) o ingresar el signo de asterisco (*). Este comodín le indica a Gii que debe barrer la base en búsqueda de todas las tablas y generar el Modelo de cada una de ellas. Si se opta por la primera opción, es interesante notar que aunque un segundo campo denominado 'Model Class' es editable, la propia interfaz le da un nombre igual o similar al ingresado en el primer campo de texto. Se recomienda aceptar la

sugerencia de Gii, de esta manera se asegura que todos los modelos sigan un mismo estándar.

5. En el caso puntual de este desarrollo, dejar el check activo del campo 'Build Relations' permitió mantener activas las relaciones entre tablas también a nivel del proyecto PHP. Se recomienda dejar seleccionada esta opción si hay relaciones entre entidades.
6. Dar clic en el botón 'Preview' y observar el resultado.
7. Dar clic en el botón 'Generate' y observar el resultado.
8. Si en el paso 5 se eligió generar un Modelo a la vez, será necesario repetir el proceso hasta completar la totalidad de tablas.

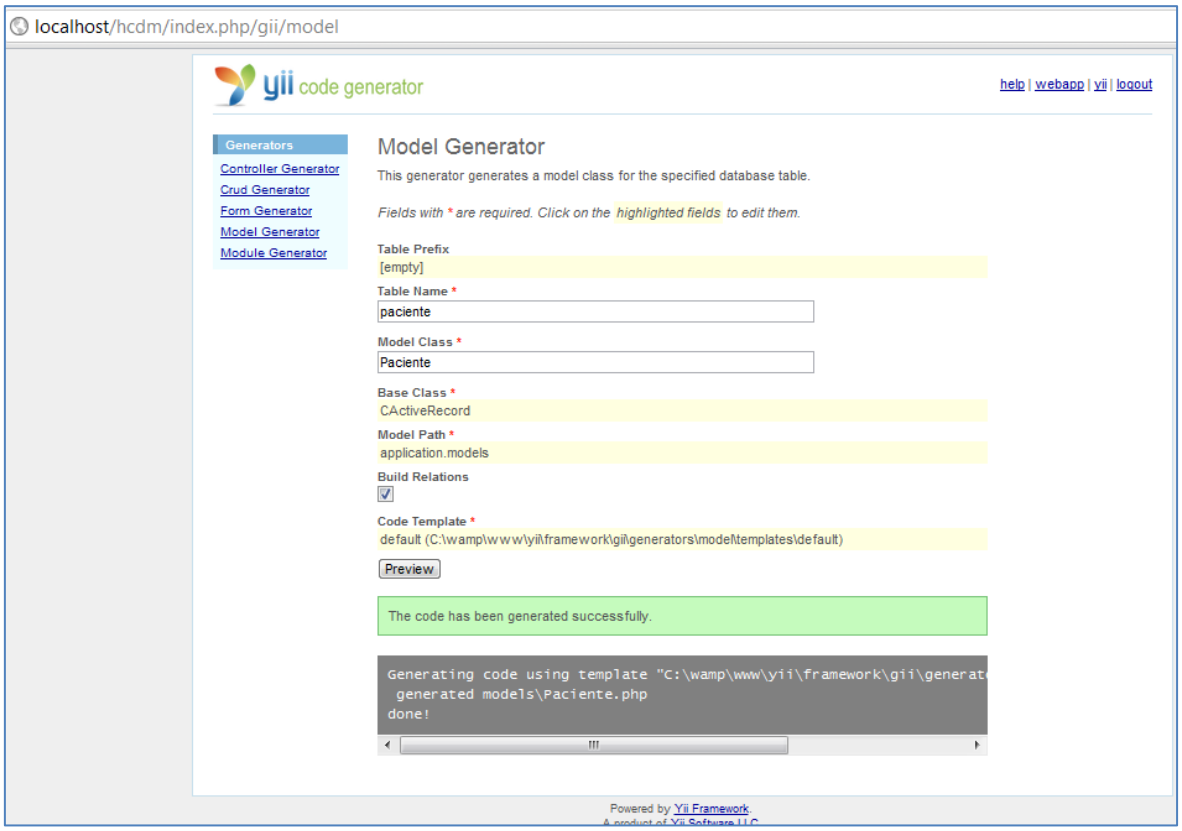


Ilustración 3-8 Generación Automática de un Modelo de MVC en Yii [B]

Se pueden encontrar todos los modelos generados en la siguiente ubicación: /hcdm/protected/models. Cada fichero PHP es en realidad una clase con los campos de la tabla como atributos y, entre otras funciones, las relaciones entre entidades como métodos. Una vez obtenidos todos los Modelos se puede obtener el diagrama de clases de esta capa de MVC, gracias a las bondades ofrecidas por Enterprise Architect.

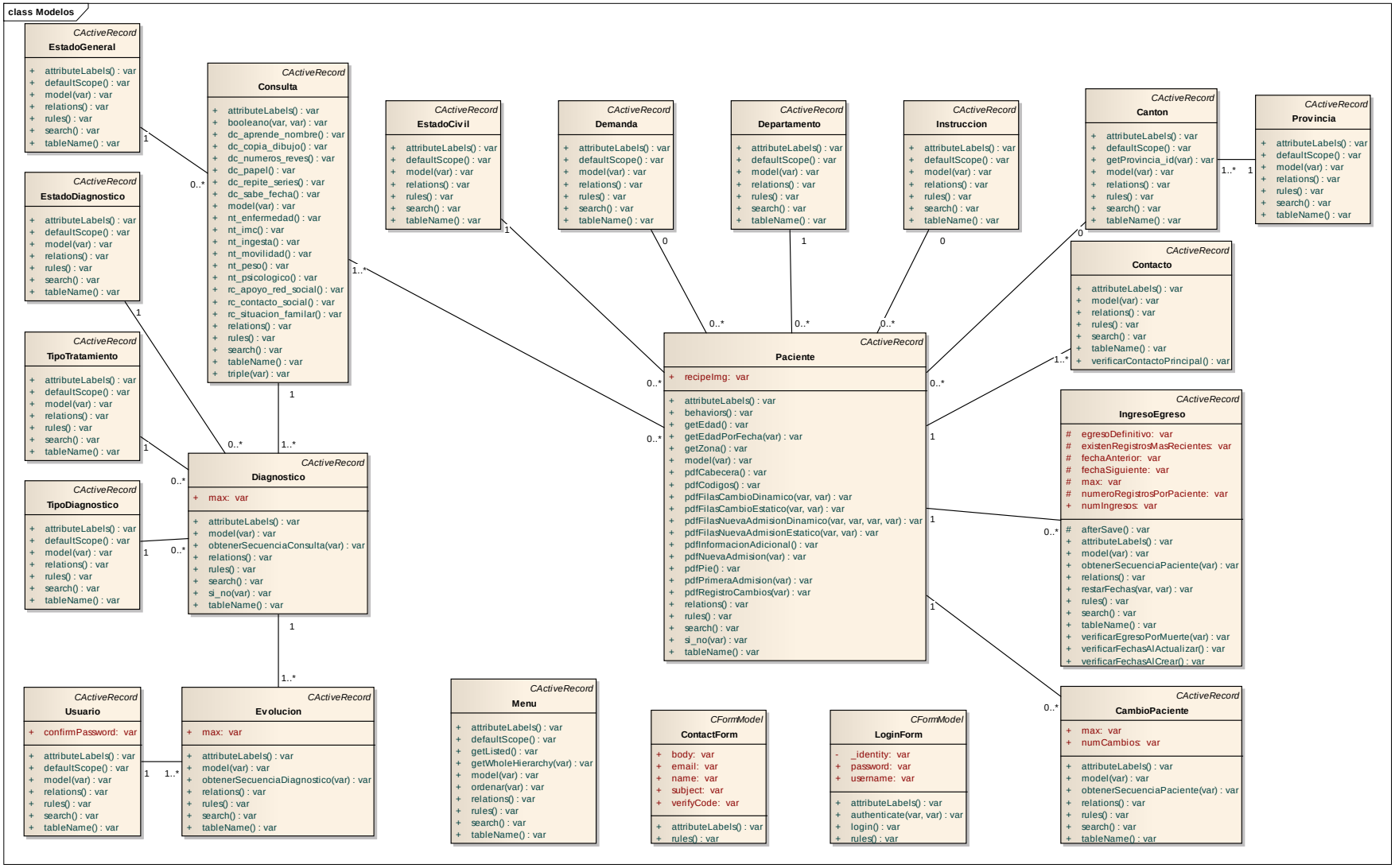


Ilustración 3-9 Diagrama de Clases de Modelos de MVC [B]

El diagrama de clases de los Modelos de MVC presenta muchas similitudes con el Modelo Entidad Relación de la base de datos y claro, siendo generado el modelo de clases a partir de tablas, parecería obvia la profunda relación entre diagramas. Sin embargo, existen diferencias significativas entre ellos:

- El Modelo Entidad Relación incluye tablas del módulo SRBAC que no son consideradas en el diagrama de clases. A pesar de que sí hay programación – y probablemente clases definidas – para tal módulo, la forma en la que ha sido desarrollado está fuera del alcance de este proyecto.
- En el diagrama de clases existen dos entidades adicionales que no aparecen en el Modelo Entidad Relación. Estas clases son 'ContactForm' y 'LoginForm', ambas generadas automáticamente al momento de crear el esqueleto de la aplicación. La usabilidad tanto de la forma que permite ingresar al sistema como de la que permite ponerse en contacto con el administrador, es tan alta que mantener sus respectivas formas dentro del sistema es una necesidad que resulta en ahorro de trabajo.
- El diagrama de clases contiene el listado de atributos y métodos de cada entidad mientras que en el modelo lógico de base de datos, únicamente se muestra el nombre de las tablas y las relaciones entre ellas.

Usualmente sucede al contrario, es decir, a partir de un diagrama de clases se genera el código DDL para crear el MER. No obstante, basándose en un buen modelamiento a nivel de la base de datos, Yii dicta que el camino es de forma inversa y los resultados son excelentes. Haciendo uso de herramientas CASE, se pueden obtener diferentes modelos que más tarde servirán como apoyo fundamental en la fase de mantenimiento del sistema. Vale recalcar que si bien todas las clases son obtenidas a partir del código, las relaciones entre ellas se han creado manualmente.

3.6 Generación de Controladores y Vistas

Al igual que lo expuesto en la sección anterior, la generación de controladores y vistas es realizada por la interfaz Gii.

Los pasos para realizar esta generación, son:

1. Ir a http://localhost/hcdm/index.php/gii
2. Ingresar la clave de acceso al módulo (Ver Figura 'Habilitación de Gii').
3. Seleccionar la opción 'CRUD Generator'. Nótese que existen también las opciones 'Controller Generator' (Controlador) y 'Form Generator' (Vista) sin embargo, estas generan código independiente entre cada capa, mientas que la opción CRUD, genera código que permite la *comunicación* entre todo MVC.
4. En la siguiente pantalla, en el campo 'Model Class' se debe ingresar el nombre de cada clase generada en la sección anterior o, una vez más, se puede hacer uso del signo asterisco (*) para evitar procesar una por una. Lo ingresado es sensible a mayúsculas y minúsculas. El campo 'Controller ID' es llenado automáticamente al ingresar el nombre de la clase. Se pude cambiar tal nombre pero se recomienda no hacerlo.
5. Dar clic en el botón 'Preview'. Antes de generarse el código se despliegan los archivos que se crearán y su respectiva ubicación dentro del proyecto. En condiciones normales se crea 1 archivo que servirá como Controlador y 8 archivos en la capa de Vistas. Con respecto a los ficheros de la capa Vista, se denominan 'vistas parciales' a aquellas cuyo nombre empieza con guión bajo (_), debido a que no son usadas directamente por el usuario sino que son invocadas por otros archivos, los conocidos como 'vistas definitivas' o totales.
6. Dar clic en el botón 'Generate' y observar el resultado
7. Si en el paso 4 se optó por generar un CRUD a la vez, será necesario repetir el proceso hasta completar la totalidad de clases.

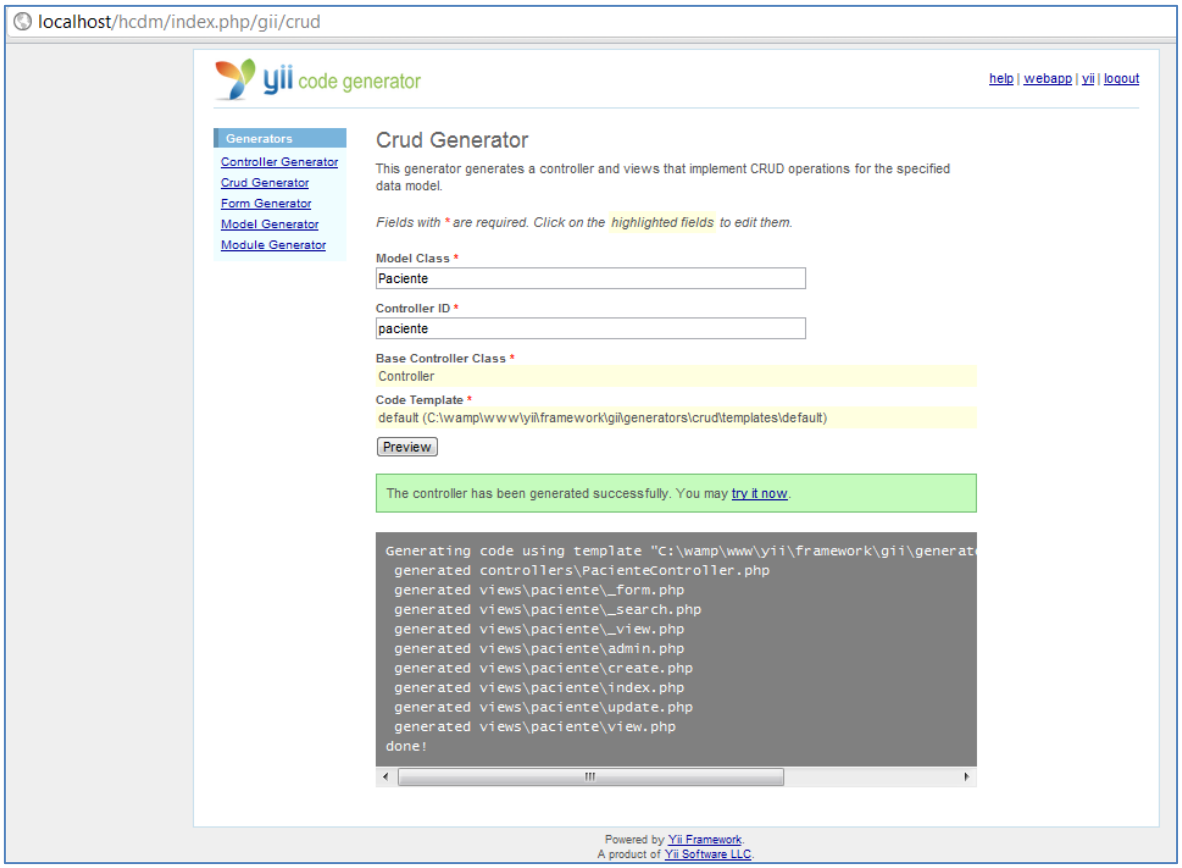


Ilustración 3-10 Generación Automática de CRUDs en Yii [B]

Los controladores generados se encuentran dentro de /hcdm/protected/controllers mientras que las vistas están en /hcdm/protected/views. En el caso de las vistas, por haber más de 1 archivo por cada Modelo y Controlador, la ubicación de los ficheros está un nivel más profundo, es decir, existe un directorio con un nombre similar al del controlador que agrupa los 8 archivos. Ejemplo: /hcdm/protected/views/paciente/.



Ilustración 3-11 Diagrama de Clases de Controladores de MVC [B]

Este diagrama contiene las clases con atributos y métodos que representan a los controladores de MVC. Llama la atención el hecho de que no haya relaciones entre clases; esto se debe a que el propósito de un controlador es gestionar requerimientos desde las Vistas hacia los Modelos y las relaciones se manejan a nivel de estos últimos.

Nótese que no existe un diagrama de clases de Vistas. La explicación para esto es que los archivos que componen las vistas están compuestos, casi en su totalidad, por código HTML, es decir, no son clases y por tanto es imposible obtener un diagrama de ese tipo para representarlos.

En este punto del desarrollo, un usuario final está en la posibilidad de ingresar nuevos registros, modificar o eliminar los existentes, ver el listado completo y realizar búsquedas simples o avanzadas. Todo esto para el caso de la entidad con la que se ha trabajado como ejemplo: Paciente. La siguiente figura muestra la interfaz de una de esas posibles acciones.



Ilustración 3-12 Pantalla de Creación de Pacientes [B]

3.7 Características Adicionales de Yii

Entre el amplio y generoso listado de características de Yii, se incluyen las siguientes:

- Generación automática de breadcrumbs o migas de pan. Gracias a éstas, el usuario puede saber exactamente en qué parte del sistema está ubicado y puede regresar al nivel deseado en cualquier momento mediante un simple clic.
- Menú de 'Opciones Básicas' al lado derecho de la aplicación. Dependiendo de la vista en la que el usuario esté ubicado, se despliegan distintas opciones.

Vista	Opciones Disponibles en Menú 'Opciones Básicas'
Listar todos los registros	Crear y Buscar
Ver un registro	Actualizar y Eliminar
Ingresar	*No presenta opciones*
Actualizar	*No presenta opciones*
Eliminar	*No presenta opciones*

Tabla 3-2 'Opciones Básicas' de cada Vista [B]

Estas características están presentes en el nuevo sistema debido a su gran usabilidad para el usuario final.

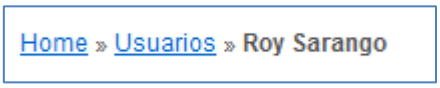


Ilustración 3-13 Bredcrumbs [B]



Ilustración 3-14 'Opciones Básicas' [B]

3.8 Desarrollo del Módulo de Configuración

A partir de esta sección se sobreentiende que para cada tabla de la base de datos – excepto aquellas que pertenecen al módulo SRBAC – se han generado Modelos y CRUDs, lo que incluye Controladores y Vistas.

El menú de opciones de la aplicación está basado en una extensión de Yii llamada SMenu. Esta extensión permite manejar el conjunto de opciones como arreglos que pueden ser obtenidos directamente desde la base de datos, permitiendo establecer una jerarquización mediante una relación recursiva de la tabla que contiene estos ítems.

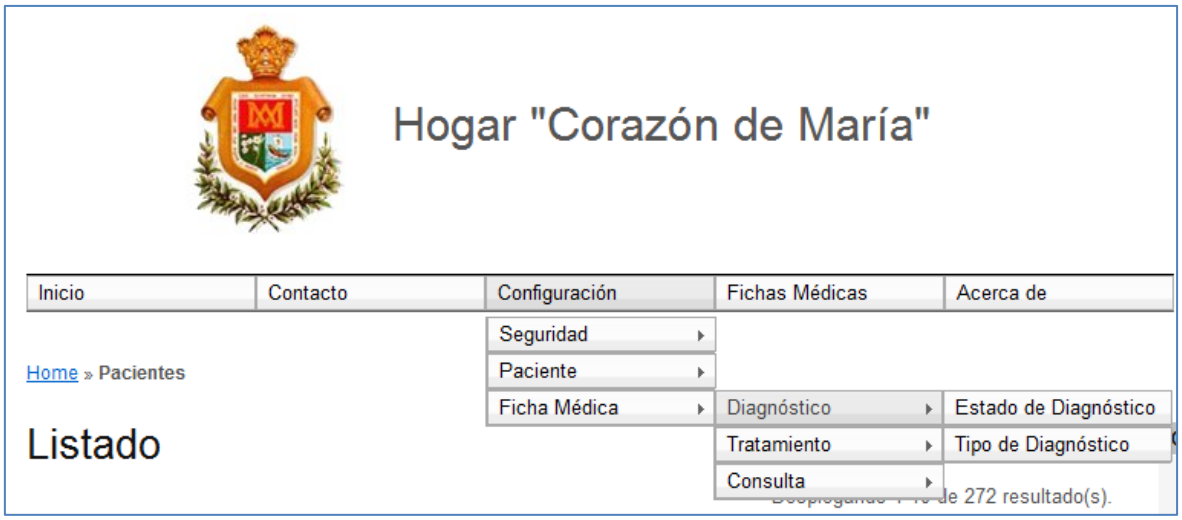


Ilustración 3-15 Despliegue de Opciones de la Barra de Menú [B]

En la figura se aprecia una jerarquización de opciones donde, por ejemplo, en el nivel 0 existen ítems como ‘Inicio’, ‘Contacto’, ‘Configuración’, ‘Fichas Médicas’ y ‘Acerca de’; mientras que en el nivel 1, de padre ‘Configuración’, existen las siguientes opciones: ‘Seguridad’, ‘Paciente’ y ‘Ficha Médica’ y así hasta alcanzar los niveles finales. Cabe indicar que para que el usuario sea direccionado a la opción seleccionada, cada menú de nivel final deberá tener asociado un código que, al ser concatenado con la URL principal, le permite a la aplicación identificar el lugar exacto de destino. Por ejemplo, si el código del ítem ‘paciente’ es ‘/paciente’, la dirección URL que originalmente era http://localhost/hcdm/index.php, se convertirá en http://localhost/hcdm/index.php/paciente.

La clasificación de información en el Módulo de Configuración del Sistema se ha simplificado al máximo. La siguiente lista identada proporciona mayor detalle visual de los niveles de estructuración y de las opciones finales, aquellas a las que el usuario puede acceder mediante un clic:

Configuración

- **Seguridad**

Opciones → Ítems que componen el menú de opciones del sistema.

Usuarios → Usuarios autorizados a ingresar a la aplicación.

Roles y Permisos → Gestión de acceso de usuarios a tareas específicas.

Paciente

Provincia → Registro de provincias del Ecuador. Una provincia puede tener uno o más cantones.

Cantones → Registro de cantones del Ecuador. Un paciente pertenece a un cantón. Debido a que conociendo un cantón, se conoce también su provincia, no existe una relación directa entre paciente y provincia (se usa el cantón como puente).

Estado Civil → Registro de los diferentes estados civiles disponibles y autorizados en el Ecuador.

Instrucción → Registro de los distintos niveles de instrucción educacional.

Departamento → Departamentos físicos del Hogar Corazón de María. De acuerdo a la infraestructura del Hogar, el sexo del paciente y su condición física y mental, cada adulto mayor es asignado al departamento más adecuado.

Demanda → Tipos de demanda por la que el paciente ingresó al Hogar. Se destacan las demandas institucionales, familiares y particulares.

Ficha Médica

Diagnóstico

Estado de Diagnóstico → Listado de estado de diagnósticos

Tipo de Diagnóstico → Tipo médico del diagnóstico dado.

Tratamiento

Tipo de Tratamiento → Tipo médico del tratamiento a seguir.

Consulta

Estado General → Posibles estados generales de un paciente.

Cada texto mostrado en azul corresponde a una opción del Módulo de Configuración. Mientras el texto en negrita es un menú padre cuyo propósito es agrupar opciones a más bajo nivel y no es seleccionable por el usuario, el texto en cursiva representa las opciones finales a las que el usuario sí puede acceder y por tal razón, se incluye una brevísima descripción de cada una de ellas. Todas estas opciones de último nivel han sido generadas

automáticamente por el propio Yii y las modificaciones realizadas a ese código se han limitado casi con exclusividad al ítem 'Opciones' y su adaptación a la extensión SMenu. Sin embargo, como se han mencionado anteriormente, el ítem 'Roles y Permisos' es una invocación completa – lo cual incluye trabajo de configuración – al módulo externo SRBAC. En la siguiente sección se darán más detalles de este sub sistema.

3.8.1 Adopción de SRBAC como Módulo de Administración de Usuarios

La creación de un módulo de administración de usuarios no es una tarea sencilla, en especial si la disgregación de permisos es de alto nivel. Afortunadamente, este requerimiento es tan común e indispensable en casi cualquier aplicación, que terceros se han dado el trabajo de desarrollarlo y han tenido a bien compartirlo gratuitamente para su uso.

El módulo SRBAC – por sus siglas en inglés de Control de Acceso Basado en Roles – al igual que cualquier otro módulo o extensión de Yii, es provisto bajo la misma licencia del framework, esto es, BSD License. Las características principales del módulo son las siguientes:

- Control de acceso de usuarios basado en roles.
- Disgregación de permisos en 2 grupos jerarquizados: Tareas y Operaciones. Un rol puede tener N tareas asignadas y, a su vez, una tarea puede tener asignadas N operaciones. El usuario administrador del módulo está en la capacidad de consultar, crear, modificar y eliminar roles, tareas y operaciones de acuerdo a sus necesidades.
- Creación automática de permisos. Si un controlador de MVC ha sido correctamente desarrollado y la clase que lo compone ha sido debidamente extendida por una clase específica de SRBAC, todos los métodos internos se convertirán en operaciones del módulo de accesos. SRBAC barre todos los controladores del sistema y, basándose en sus funciones, crea tareas y operaciones. Por defecto se realiza la siguiente distribución para cada modelo: las operaciones de *listar* todos los registros y *ver* un registro específico, son encasilladas dentro de la tarea *Lectura* y cualquier otra operación – incluyendo las que se hayan desarrollado manualmente – formarán parte de la tarea *Administración*, usualmente dentro de esta última están

operaciones como *crear*, *insertar* y *eliminar* registros.

- Si por a o b razones un controlador es eliminado, SRBAC permite limpiar los permisos obsoletos que pertenecían a esa clase.
- Existen opciones en un sistema a las que cualquier usuario puede acceder. Para este fin SRBAC administra un 'Listado de Siempre Accesibles'.
- Consulta de permisos otorgados a cada usuario.

La mayor parte del texto mostrado en el módulo ya ha sido traducido al español. Sin embargo, ciertas opciones como el tipo de ítem (rol, tarea u operación) y los permisos generados automáticamente, aún permanecen en su idioma original, el inglés.

La siguiente figura muestra la pantalla de Gestión de Permisos:

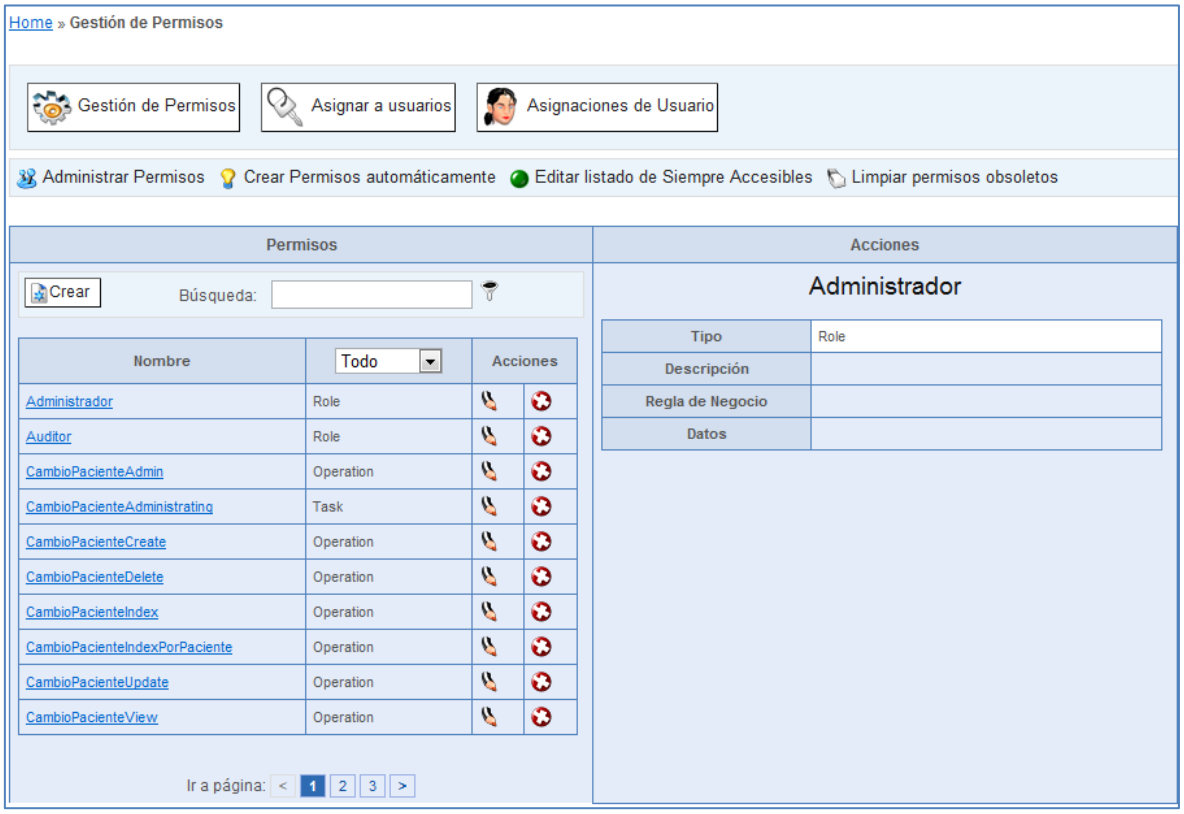


Ilustración 3-16 SRBAC, Gestión de Permisos [B]

Fácilmente se pueden apreciar dos secciones: 'Permisos' y 'Acciones'. Si en la primera sección se selecciona una acción (ver, editar o eliminar) sobre un ítem disponible en el listado, asincrónicamente – usando AJAX – en el lado derecho se muestra información

relativa a la acción requerida. Es así que lo capturado en la figura corresponde específicamente a la acción 'Ver' (después de dar clic sobre el link que compone el nombre del ítem).

La asignación de roles por usuario puede apreciarse gráficamente en la siguiente figura:

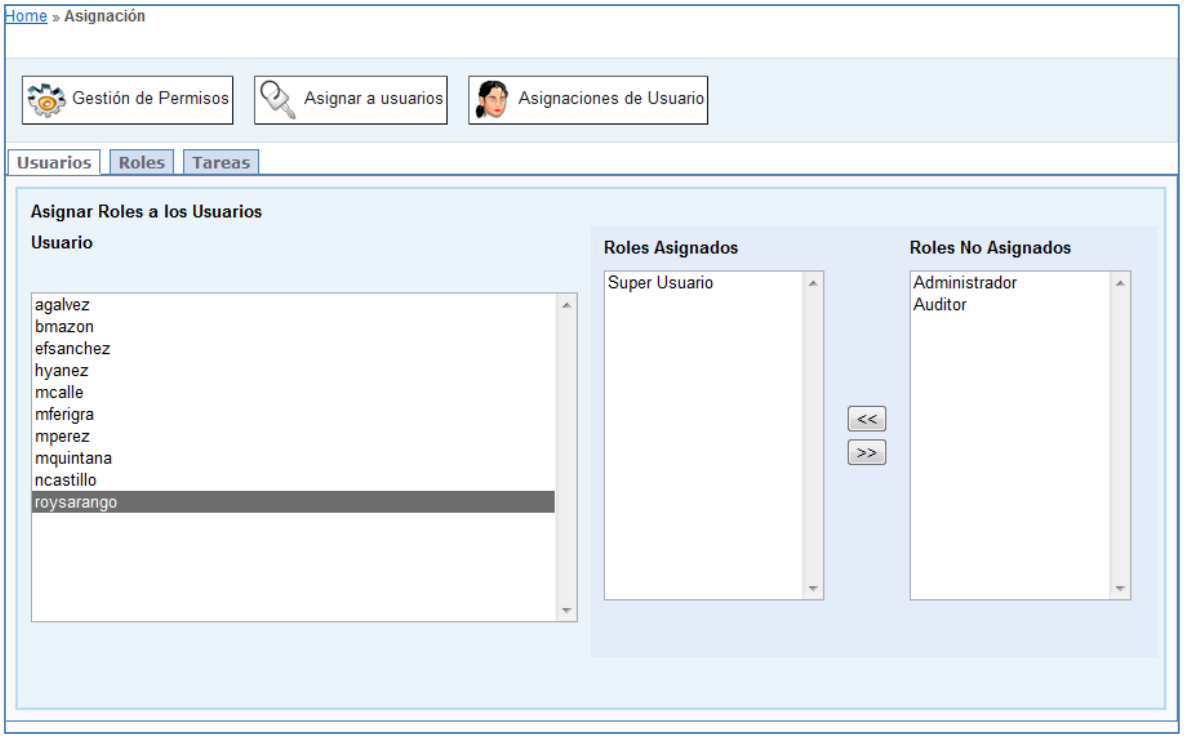


Ilustración 3-17 SRBAC, Usuarios y Roles Asignados vs. Roles Disponibles [B]

Además de la pestaña 'Usuarios', se pueden visualizar dos pestañas más: 'Roles' y 'Tareas'. Así como en lo mostrado en la figura se aprecian los roles que se le pueden asignar a un usuario, las otras pestañas permiten asignar tareas a un rol y operaciones a una tarea, respectivamente.

3.8.1.1 Configuración de SRBAC

El primer paso a realizar es descargar el módulo y ubicar los archivos que lo componen, en el directorio /hcdm/protected/modules.

Además de la inclusión de las tablas 'item', 'asignacion' y 'autorizacion' como objetos de

base de datos, es necesario configurar el módulo para que, entre otras tareas, interactúe con la tabla que contiene los usuarios de la aplicación

```php
<?php
...
'modules' => array(
        ...
        'srbac' => array(
            'userclass' => 'Usuario', //Clase que contiene la entidad de Usuarios.
            'userid' => 'usuario_id', //Clave primaria de la entidad representada por la clase anterior.
            'username' => 'username', //Atributo con el que distingue el nombre del usuario.
            'debug' => false, //true, si se está en modo de debug o false, si está en producción.
            'pageSize' => 10, // Paginación de los ítems mostrados en la pantalla de Administración de Permisos.
            'superUser' => 'Super Usuario', //default: Authorizer.
            'css' => 'srbac.css', //Archivo CSS aplicado al módulo.
            'layout' => 'application.views.layouts.main', //Ubicación del layout principal.
            'notAuthorizedView' => 'srbac.views.authitem.unauthorized', //Ubicación del archivo que muestra un
            //mensaje de acceso no autorizado.
            'alwaysAllowed' => array(
                'SiteLogin', 'SiteLogout', 'SiteIndex', 'SiteAdmin',
                'SiteError', 'SiteContact'), //Arreglo de vistas disponibles para todos los usuarios.
            'userActions' => array('Show', 'View', 'List'), //Acciones disponibles - al generarse permisos
            //automáticos - para tareas de Lectura.
            'listBoxNumberOfLines' => 15, //Número de registros mostrados en los list boxes en la pantalla de
            //asignación.
            'imagesPath' => 'srbac.images', //Ubicación de imágenes e íconos usados en el módulo.
            'imagesPack' => 'noia', //Nombre del paquete de imágenes.
            'iconText' => true, //true, si se desea que los íconos incluyan texto.
            'header' => 'srbac.views.authitem.header', //Ubicación de la cabecera del módulo.
            'footer' => 'srbac.views.authitem.footer', //Ubicación del pie de página del módulo.
            'showHeader' => true, //true, si se desea mostrar la cabecera.
            'showFooter' => true, //true, si se desae mostrar el pie de página.
            'alwaysAllowedPath' => 'srbac.components', // Ubicación del path disponible en todo momento.
        )
    ),
...
?>
```

Ilustración 3-18 Configuración de SRBAC [B]

La figura permite visualizar la configuración completa del módulo en el archivo /hcdm/protected/config/main.php. Además de parámetros y valores, existe una breve descripción del propósito de cada componente del arreglo.

Nativamente los controladores autogenerados por Gii extienden una clase componente del framework conocida como 'CController'. Para habilitar la creación automática de permisos y tareas, es necesario cambiar esa extensión a otra clase propia del módulo. Esta clase se denomina 'SBaseController' la cual, a su vez, extiende a CController para no perder las bondades del core del marco de trabajo. El trabajo de cambio de extensión debe realizare manualmente y de manera particular en los controladores de los que se desea obtener permisos automáticos. Una vez más el archivo de configuración del proyecto debe ser modificado para permitir la extensión a la nueva clase.

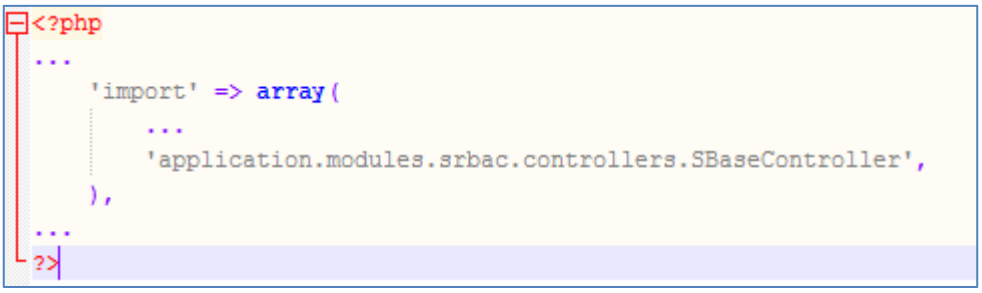


Ilustración 3-19 Importación de la Clase SBaseController [B]

Con la inclusión de este módulo se satisface el requerimiento de control disgregado de permisos basados en el rol o roles asignados a cada usuario.

3.8.2 Casos de Uso

A partir de los requerimientos se han creado diagramas de casos de uso con el fin de delimitar de mejor manera los actores y las funcionalidades que tendrá el sistema.

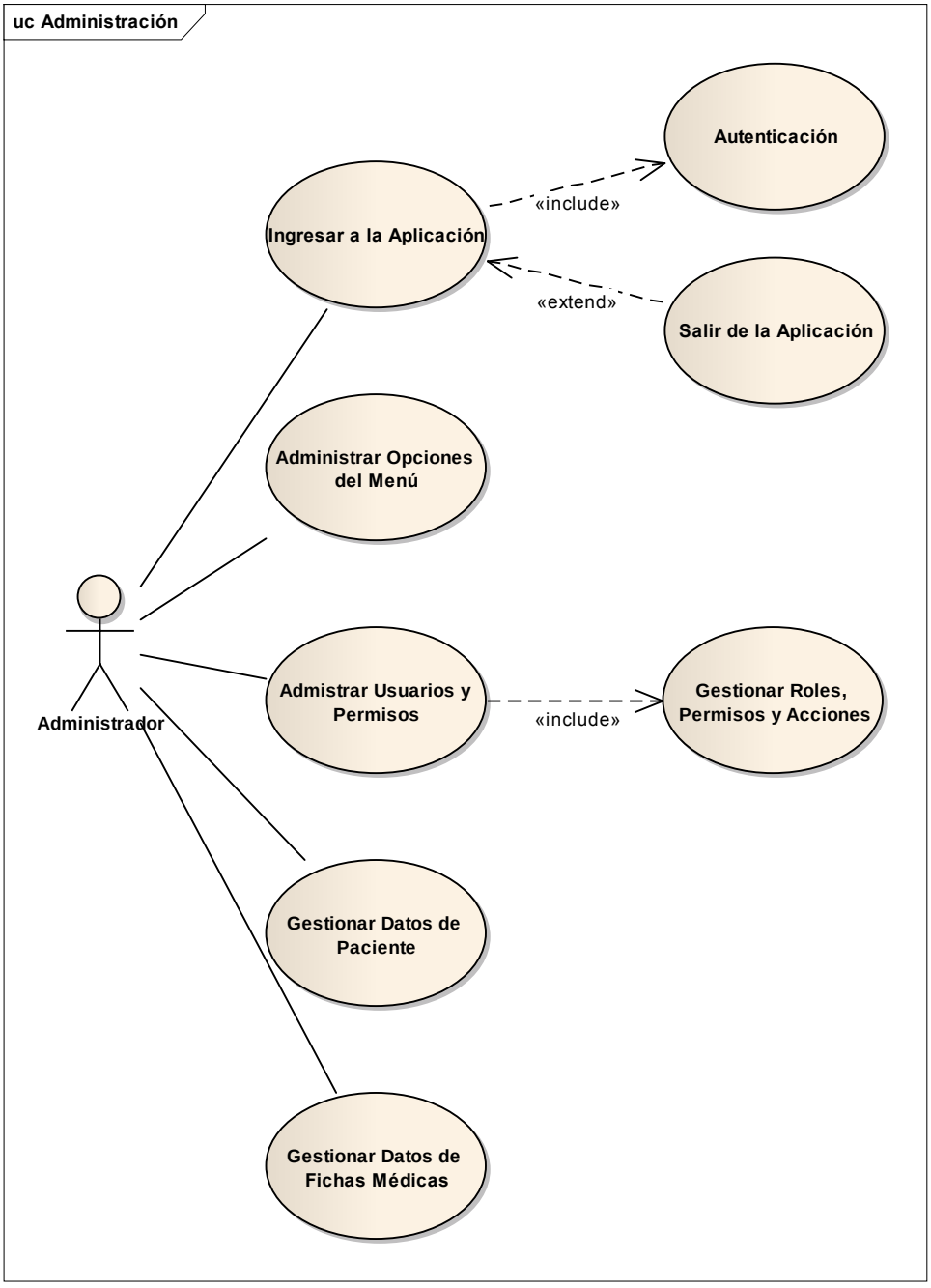


Ilustración 3-20 Casos de Uso, Módulo de Configuración [B]

En el contexto informático y más específicamente en los casos de uso creados para el desarrollo de esta aplicación, los términos ‘Gestionar’ y ‘Administrar’ son sinónimos y ambos incluyen las seis operaciones básicas a realizar sobre uno o varios registros, dependiendo del caso: seleccionar, insertar, actualizar, eliminar, buscar y ordenar.

3.9 Desarrollo del Módulo de Fichas Médicas

El core del sistema yace en el módulo de fichas médicas. Todas las posibles acciones a realizar sobre un Paciente son gestionadas desde esta sección de la aplicación.

Al igual que en el módulo de configuración, la mayor parte del trabajo la ha realizado Yii gracias su generación automática de código. Sin embargo, en lo que respecta a Fichas Médicas, se han ejecutado muchas más personalizaciones.

3.9.1 Distribución de la Información

Una de las diferencias entre el sistema anterior y el migrado es que en este último, la tabla ‘paciente’ contiene con exclusividad datos generales del adulto mayor y de ninguna manera incluye información médica o psicológica. Antes, todos los datos propios del paciente así como los relativos a su historia clínica, pertenecían a la misma tabla de base de datos.

La nueva distribución de la información permite que un mismo paciente pueda tener varias consultas y aunque es verdad que en el modelo de ficha médica que propone el Ministerio de Salud Pública hay datos inmutables, como los relativos a antecedentes personales o familiares, no es menos cierto que hay datos que pueden variar de consulta a consulta, por ejemplo, los estados emocional y psicológico del paciente.

La solución planteada en el sistema fue dividir la tabla antes unificada, en dos: ‘paciente’ y ‘consulta’, con una relación de uno a muchos respectivamente. Un paciente puede o no tener consultas y una consulta necesariamente debe estar asociada a un paciente. Este enfoque permite satisfacer el requerimiento que solicitaba la división entre datos generales e historia clínica.

Las siguientes dos figuras muestran la distribución de información en las tablas:



Registro	3790
Apellido Paterno	AGUIRRE
Apellido Materno	GONZALEZ
Primer Nombre	JOSE
Segundo Nombre	HONORIO
Estado	Inactivo
Cédula	1700144130
Dirección	
Barrio	Cotocollao
Provincia	
Cantón	
Zona	
Teléfono	
Fecha de Nacimiento	1922-06-07
Edad	N/A
Lugar de Nacimiento	
Nacionalidad	
Grupo Cultural	
Género	Masculino
Estado Civil	
Instrucción	
Ultimo Año Aprobado	
Fecha de Ingreso	2001-02-19
Ocupación	
Empresa	
Tipo de Seguro de Salud	
Referido De	
Departamento	Semanería Hombres
Demanda	
Jubilado	
Montepio	
ISSFA	
Discapacidad	No
Enfermedad Crónica	
Bono de Desarrollo Humano	

Ilustración 3-21 Datos Generales del Paciente [B]

La gráfica muestra la información general del adulto mayor y además permite visualizar el menú ‘Otras Opciones’, el cual ofrece un listado de acciones adicionales a las que el usuario podría acceder, todas ellas relacionadas con el paciente en cuestión. El primer elemento del listado se denomina ‘Consultas’ y, haciendo clic sobre ella, se accede al historial de atenciones del adulto mayor. Un usuario, con los permisos necesarios, podría crear, editar o eliminar consultas.

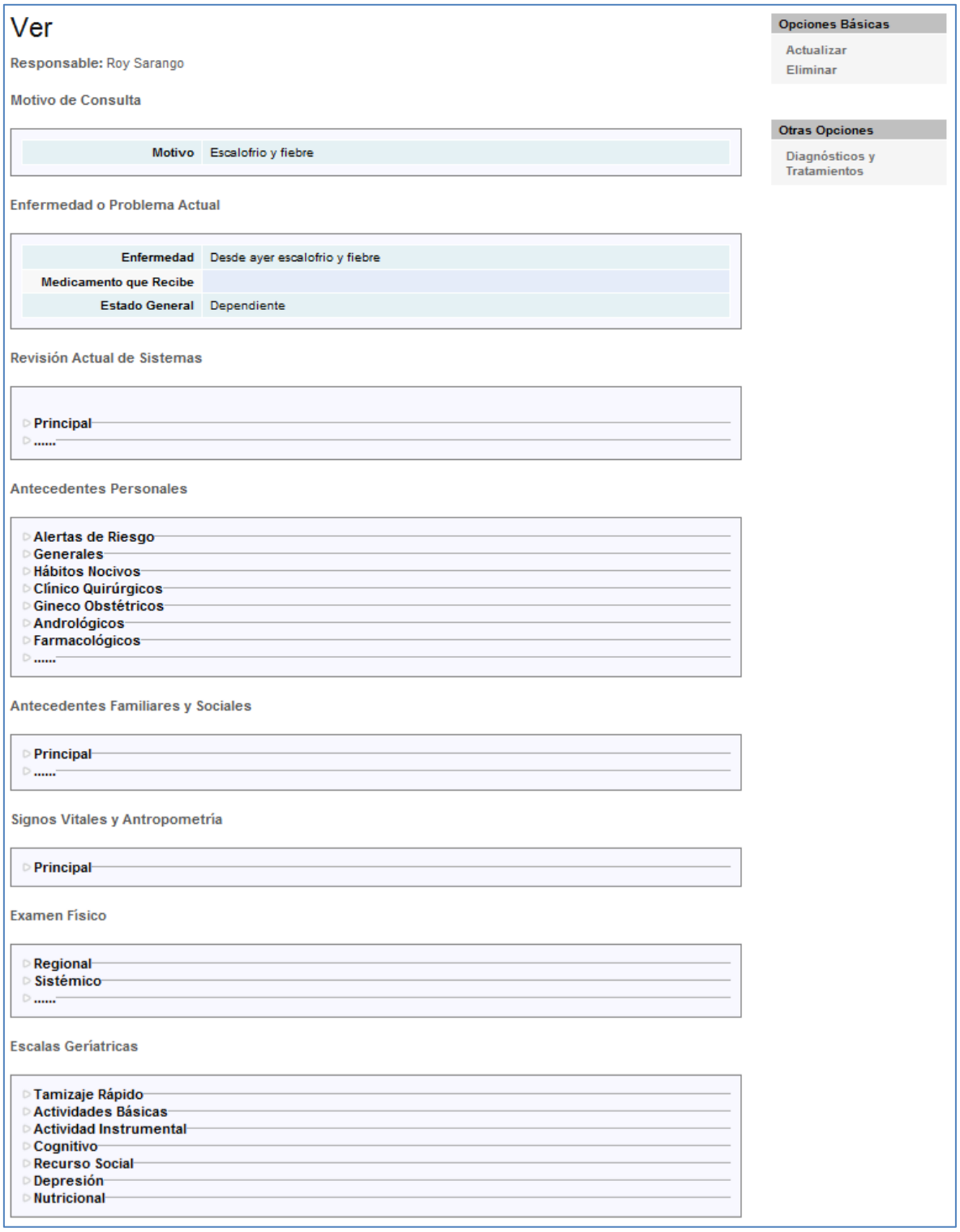


Ilustración 3-22 Consulta [B]

Como se puede apreciar, la gran mayoría de datos relativos a la consulta, no aparecen directamente en la figura. Esto se debe a la enorme cantidad de ítems que forman parte de la ficha médica. A partir de la tercera sección – Revisión Actual de Sistemas – las sub

secciones internas están colapsadas.

Tal como sucede con la relación entre las tablas 'paciente' – 'consulta', existen relaciones con la misma cardinalidad entre las tablas 'consulta' – 'diagnostico' (que incluye el tratamiento al cual se somete el paciente) y 'diagnostico' – 'evolucion'. Basándose en los datos de todas estas tablas se obtiene el reporte HCU-form.057/2010; otro de los requerimientos específicos que impulsaron el proceso de migración. La generación de reportes en formato PDF fue conseguida con la ayuda de la extensión TCPDF [15].

Se han añadido un par de ayudas visuales con el fin de mejorar la experiencia del usuario en la utilización del sistema:

1. Todos los campos de fecha pueden ser llenados gracias a un pop up que se encarga de mostrar un calendario en línea. Esta funcionalidad fue conseguida por la utilización de la extensión SCalendar [16].
2. Por cada paciente se puede incluir una foto personal. Independientemente del tamaño de la foto, el usuario está en la posibilidad de hacer una carga del archivo binario que contiene la imagen y el propio sistema se encargará de modificar su tamaño para que concuerde con el espacio asignado para su despliegue. Una vez más se echó mano de una extensión de Yii para logar este objetivo. El nombre de la extensión es Image [17].

3.9.2 Contactos

Un paciente específico puede tener varios contactos con los cuales el Hogar pueda comunicarse.

Esta característica sí estaba presente en el sistema anterior y por tanto no satisface ninguno de los nuevos requerimientos. Sin embargo, entre las dos aplicaciones hay una diferencia significativa referente a la administración de contactos del adulto mayor. El reporte HCU.form.001/2010 requiere el llenado del campo 'En caso necesario llamar a' y, para obtener este dato único, muchos sistemas seleccionan el primer registro de contacto del paciente en cuestión. No obstante, esta forma de selección no es la más acertada. Nadie asegura que aquel primer contacto ingresado sea el principal o el que se deba tener en

cuenta en caso de emergencia. Por esta razón el sistema migrado incluye una distinción entre tipos de contactos de tal manera que por cada paciente puede existir un único contacto principal y n contactos secundarios. Es este usuario principal el que se toma en cuenta al momento de obtener el reporte HCU.form.001/2010.



Ilustración 3-23 Contacto de Paciente [B]

La figura muestra los datos que tiene cada contacto de un paciente particular. Destaca el campo ‘Tipo’, que indica si esta persona es la más próxima al adulto mayor.

3.9.3 Egresos y Reingresos

Esporádicamente un anciano es retirado del hogar por familiares o amigos y reubicado en casas particulares o en algún otro centro gerontológico. Cuando esto sucede, el sistema permite gestionar tal egreso de manera que el estado del paciente cambie de activo (presente en el hogar) a inactivo.

Más de una vez ha sucedido que un paciente que egresó vuelva a ingresar. Esto se conoce como ‘Reingreso’ y como es natural, al registrar el reingreso del adulto mayor, su estado cambia de inactivo a activo.

Ambos registros – egreso y reingreso – son almacenados en la misma tabla de base de datos: ‘ingresos_egresos’. Para distinguir una acción de la otra, se maneja un ‘tipo’ con un valor específico asignado para cada caso.

La aplicación permite registrar un número indefinido de ingresos y reingresos. Además,

basándose en el estado del paciente, reconoce automáticamente cuál es el nuevo tipo. Sin embargo, hay una restricción: Al registrar un egreso se debe obligatoriamente indicar el motivo del mismo y si tal motivo fue por muerte del paciente – algo no poco común en centros de atención al adulto mayor – el sistema tendrá en cuenta este particular y no permitirá futuros reingresos. También se incluyen controles para que no se permita traslape de fechas en el orden cronológico de egresos y reingresos.

El sistema no permite eliminar estos registros debido a su afectación directa al estado del paciente y, aunque se permite la modificación, también se controla que el tipo – egreso o reingreso – no sea modificado.



Ilustración 3-24 Reingreso de Paciente [B]

3.9.4 Historial de Cambios

Por requerimiento del Ministerio de Salud Pública, cada cambio realizado sobre los datos generales de un paciente debe registrarse en el sistema. Este historial de modificaciones es requerido en el reporte HCU.form.001/2010. Cualquier modificación sobre uno o varios campos, realizada antes de 'guardar' los cambios, se almacena como un registro nuevo en la tabla 'cambio_paciente'. Cabe recalcar que esta tabla es una copia casi exacta de la tabla 'paciente'. Únicamente se aumentan un par de atributos: el primero de tipo entero auto incrementable, servirá para guardar la clave primaria de la nueva tabla, mientras que el segundo de tipo fecha, almacenará el dato de cuándo se realizó el cambio. La relación entre estas dos entidades – paciente y cambio_paciente – es de uno a muchos, respectivamente.

Los valores existentes antes de aplicar el cambio, son los que se guardan en la nueva tabla, mientras que la entidad principal – la de paciente – almacena los nuevos.

Debido a que los registros de cambios se almacenan automáticamente, el sistema no

permite eliminar o modificar esta información.



Ilustración 3-25 Registro de Modificación de Paciente [B]

Como corresponde, no todos los campos se llenan, sino únicamente aquellos que fueron modificados en la tabla principal.

Datos de las secciones ‘Contactos’, ‘Egresos y Reingresos’ e ‘Historial de Cambios’ más los datos generales del paciente, son usados para generar el reporte HCU.form.001/2010. Con esto se satisface otro de los requerimientos específicos del nuevo sistema.

3.9.5 Personalización de Búsqueda de Pacientes

Es el propio generador de código de Yii el encargado de proporcionar la funcionalidad de búsquedas básica y avanzada de paciente. Sin embargo, para adaptarse a las necesidades específicas del hogar, ha sido necesario realizar ciertas modificaciones.

El primer cambio tiene que ver con la modificación del tipo de elemento gráfico que presenta la información. Por defecto todos los datos son mostrados como campo de texto, no obstante, para buscar pacientes cuyos valores pertenecen a un listado ya definido, es necesario cambiar el campo de texto y convertirlo en una lista desplegable. Esto requiere de cambios en la vista de MVC y en la forma como se obtienen los valores. Estos últimos pueden estar quemados cuando se limitan a opciones Si/No, género Masculino/Femenino, etcétera; o pueden ser recuperados de una tabla relacionada.

La segunda modificación es la inclusión de un botón para exportar los datos a Excel. Si bien los filtros funcionan de manera estupenda en el navegador, muchas veces es necesario obtener esos datos en un archivo editable y en un formato popular.

Mientras que la búsqueda básica permite filtrar la información por los datos más relevantes del paciente, la búsqueda avanzada realiza la misma función pero además le ofrece al usuario la posibilidad de filtrar los resultados encontrados, usando cualquier campo que componga la entidad.

Las características de búsqueda básica y búsqueda avanzada son mérito exclusivo de Yii y su generador automático de código. Esto significa que cualquier tabla de base de datos puede beneficiarse de tales funcionalidades y, de hecho, así lo hace todo el módulo de configuración.

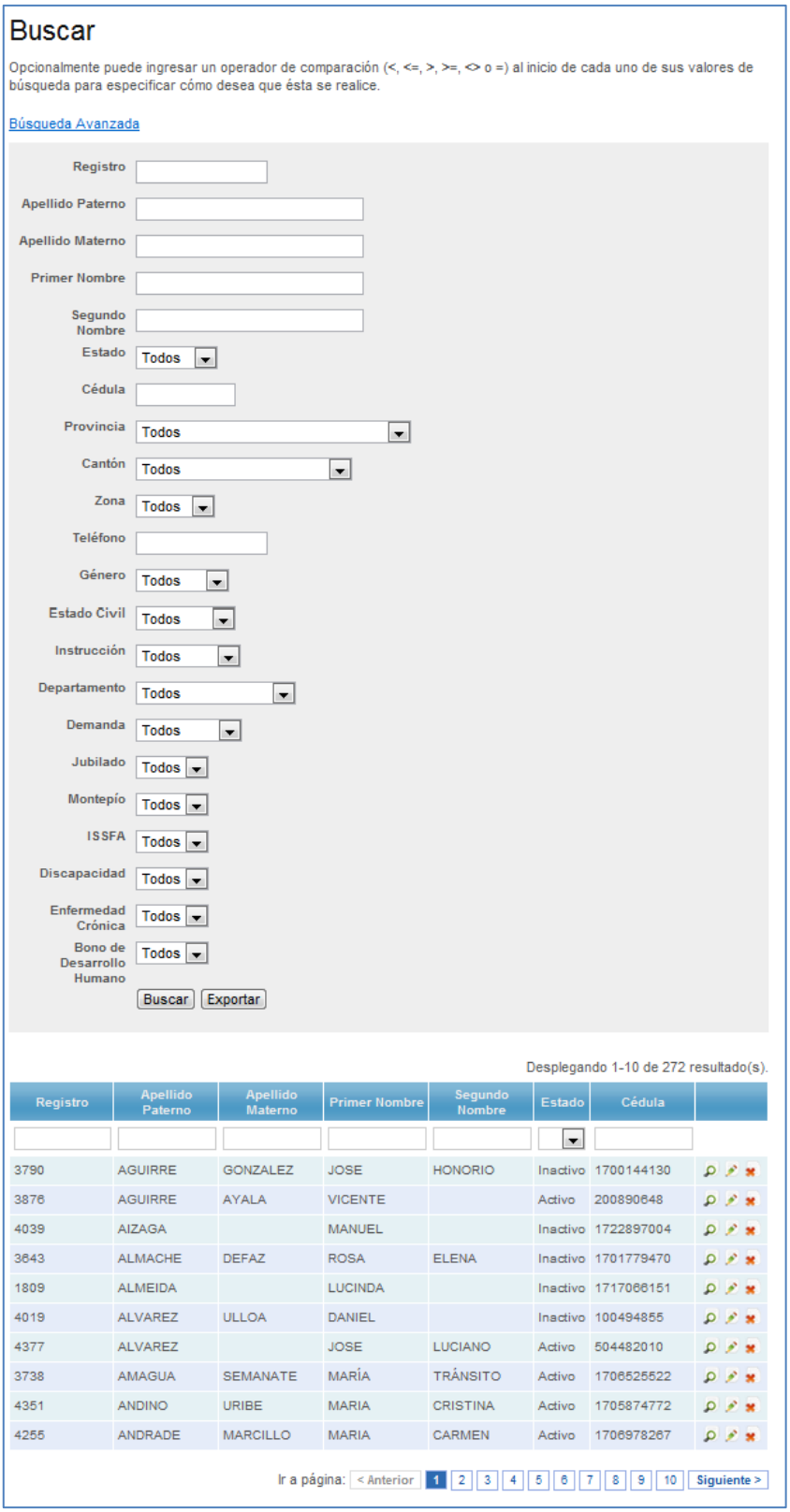


Ilustración 3-26 Búsqueda de Pacientes [B]

En la figura, la sección con fondo gris muestra la búsqueda avanzada mientras que la fila

con fondo blanco justo sobre la sección de fondo celeste, muestra la búsqueda básica. Nótese que en la búsqueda avanzada se encuentra el botón 'Exportar', mencionado anteriormente. Cómo es lógico, los resultados encontrados son presentados con paginación y cada uno de ellos puede ser visualizado, editado o eliminado.

3.9.6 Casos de Uso

Tal como el apartado 'Casos de Uso' de la sección referente al módulo de configuración, el siguiente diagrama muestra cuales son los casos de uso y las relaciones que existen entre estos y los usuarios finales.

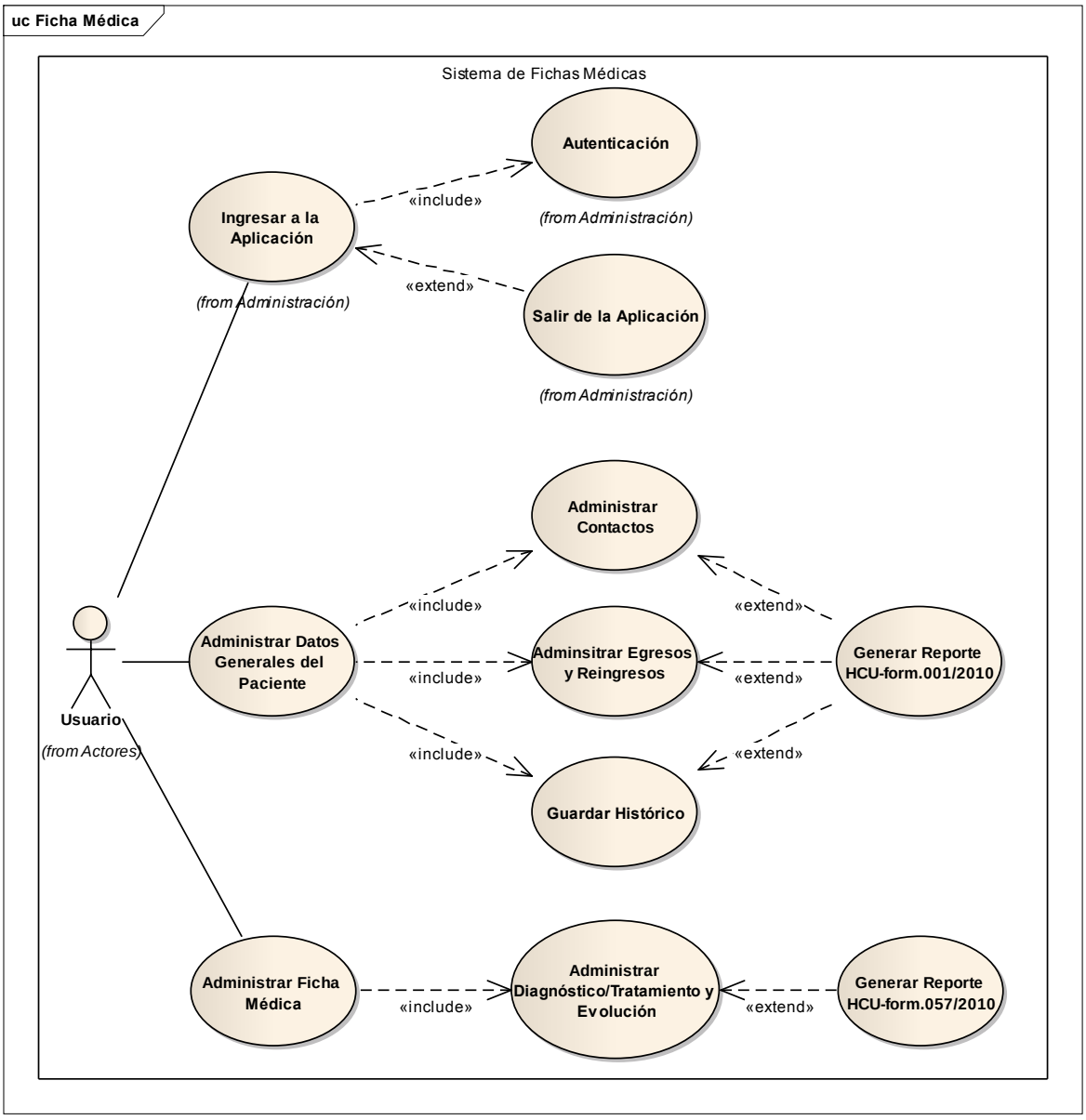


Ilustración 3-27 Casos de Uso, Módulo de Fichas Médicas [B]

En la figura se puede notar que existe un único actor denominado ‘Usuario’. A este actor se le concedió el permiso de realizar todas las acciones posibles. De esta manera se evitó hacer distinción entre diferentes roles con el propósito de que sea el Administrador del Sistema quien otorgue o quite permisos dependiendo del perfil de la persona que use la aplicación. Como se mencionó con anterioridad, el módulo SRBAC se encarga de la configuración y administración de tales tareas.

4 LIBERACIÓN Y CAPACITACIÓN

Concluida la etapa de desarrollo y pruebas, la siguiente y última fase es la de liberación o lanzamiento de la aplicación. Hay varios puntos a ser considerados antes de pasar un sistema del ambiente de desarrollo al de producción. Las siguientes secciones hacen hincapié en cada uno de esos aspectos.

4.1 Carga Inicial de Información

El sistema anterior no contaba con toda la información de los adultos mayores residentes del Hogar y los pocos datos existentes no estaban actualizados. Para evitar ingresar datos manualmente o desactualizados, se echó mano del listado que está bajo mantenimiento constante por uno de los departamentos del centro. Este archivo, en formato XLS, ha permitido realizar la importación de datos a la fecha.

El primer paso para la carga inicial de la información fue modificar el archivo para que los valores que en el sistema son obtenidos de una lista desplegable que referencia a otra tabla, contengan los códigos de sus registros correspondientes en lugar del texto que entiende el usuario a simple vista. Tomando como ejemplo el caso más simple, consideramos que el género de la persona puede ser masculino o femenino y así lo entiende el usuario pero a nivel de base de datos, esos valores se guardan como 1 o 2, respectivamente. Este tipo de cambios fueron necesarios para conseguir una importación exitosa.

Si bien es cierto que cambiar manualmente tales correspondencias es un trabajo repetitivo y agotador, también es verdad que se ahorra tiempo al no ingresar todos los campos sino modificar tan solo unos cuantos para su posterior importación.

Una vez terminada la modificación del archivo, fue necesario cambiar su tipo, de XLS (propio de Microsoft Excel en versiones 2003 o anteriores) a CSV (Valores separados por comas). Este paso es fácilmente conseguido con la ayuda del propio Excel y su característica básica de ‘Guardar Como’.

Para realizar la importación como tal, se utilizó una sentencia nativa del motor de base de

datos de MySql. Esta sentencia se llama 'LOAD DATA INFILE' [18] y permite realizar la importación desde la misma línea de comandos. Después de ingresar exitosamente a la base de datos con los permisos requeridos, se ejecutó la siguiente sentencia a nivel de consola:

```
LOAD DATA INFILE 'pacientes_hogar_corazon_de_maria.csv' INTO TABLE
hcdm.paciente
```

La importación total fue de 265 registros de pacientes, entre activos e inactivos.

4.2 Contratación de Hosting y Dominio

Entre los requerimientos iniciales de la migración constaba el cargar el sistema en Internet. Gracias a que el desarrollo se lo hizo 100% sobre plataformas, lenguajes de programación y base de datos gratuitos, la contratación del hosting no exigió el pago de licencias adicionales. Además, se ha aprovechado la seguridad y robustez de Linux como servidor de aplicación y de base de datos.

El caso del dominio es diferente en varios aspectos. Primeramente, un dominio no tiene nada que ver con el rendimiento de una aplicación y tampoco su costo está relacionado con las tecnologías y licencias que se usen. Además, la disponibilidad de un domino específico depende de si alguien más ya esté usando el nombre y el sufijo deseados. Un dominio entero es asignado a una única persona, entidad o empresa. Por ejemplo, sabemos que el dominio google.com es propiedad del *monstruo* informático Google y que nadie más puede hacer uso de él – probablemente tampoco de las extensiones asignadas a este motor de búsqueda en cada país.

El hosting y el dominio pueden ser alquilados de manera independiente. Es decir, se puede alquilar el hosting a una empresa y comprarle el dominio a otra distinta. Todo depende de las necesidades del contratante.

En cuanto al tema económico se puede decir que el precio por la obtención de estos dos servicios depende de un gran número factores – tiempo de alquiler o compra, software a ser utilizado, licencias, número de cuentas de correo electrónico provistas, capacidad de transferencia mensual, capacidad de almacenamiento, número de bases de datos, etcétera – sin embargo, existen proveedores reconocidos que ofrecen servicios adicionales y que garantizan mayor disponibilidad de acceso a la información. Una vez más, la decisión final queda a discreción del usuario.

En el caso particular del sistema migrado es necesario indicar que el Hogar Corazón de María se ha visto beneficiado de un programa impulsado por el Ministerio de Inclusión Económica y Social, el cual busca colaborar con los centros gerontológicos del país en temas informáticos. Entre otras ayudas ofrecidas, este ministerio – por medio de una tercera empresa – se ha encargado de proveerle al hogar, ambos servicios: Hosting y Domino. El hosting donde residirá la aplicación migrada se denomina tuWeb.us y el link referencial es el siguiente: http://www.tuweb.us. Con esta misma empresa y como parte de un paquete profesional se compró el siguiente dominio: http://www.hogarcorazondemaria.com.

Dando clic en el último link, cualquier usuario registrado y con los permisos debidos, puede hacer uso de la aplicación desde cualquier parte del mundo.

4.3 Liberación al Ambiente de Producción

La contratación del hosting y dominio es solo el primer paso en el proceso de preparación del ambiente de producción. Existe la necesidad de realizar configuraciones adicionales para asegurar el correcto funcionamiento de la aplicación y optimizar su performance.

Desarrolladores experimentados suelen hacer uso del modo *debug* para detectar y corregir errores de programación. Este modo, también conocido como modo de depuración, implica un mayor consumo de recursos del sistema y aunque es altamente aconsejable el tenerlo activo en un ambiente de desarrollo, su uso en ambiente de producción podría ralentizar el comportamiento de la aplicación y, como consecuencia, tener usuarios insatisfechos. La activación/desactivación del modo debug suele ser un proceso engorroso. Ventajosamente,

en el caso de aplicaciones creadas con Yii, no lo es. El único cambio que se requiere para desactivarlo es eliminar las siguientes líneas de código en el archivo /hcdm/index.php:

```
defined('YII_DEBUG') or define('YII_DEBUG',true);
defined('YII_TRACE_LEVEL') or define('YII_TRACE_LEVEL',3);
```

Con el modo de depuración inactivo, el siguiente paso fue migrar la base de datos del sistema. Gracias a PhpMyAdmin y su interfaz Web – todo provisto por el panel de control de tuWeb.us – la creación de una nueva base se limita a unos pocos clics en wizards altamente legibles.

Para asegurar que el usuario administrador de la base tenga permisos totales sobre ésta, se realizaron los siguientes pasos:

1. En PhpMyAdmin, clic en la pestaña ‘Privilegios’.
2. Selección de la opción ‘Crear un nuevo Usuario’.
3. Llenado de los datos de acuerdo a las necesidades. El campo ‘Host’ fue llenado con la palabra ‘localhost’, lo que indica que la base de datos reside en el mismo servidor desde donde se accederá al nuevo esquema. En esa misma pantalla, en la sección ‘Base de Datos para Usuarios’, seleccionar la opción ‘Crear base de datos con el mismo nombre y conceder todos los privilegios’. Esta opción asegura que se cree una base de datos con el mismo nombre del usuario a añadir y que éste tenga accesos totales a dicha base.
4. Clic en el botón ‘Crear Usuario’.

Una vez creada la base y el usuario en ambiente de producción, el paso siguiente inmediato es la importación de la información. Antes, como es lógico, fue necesario exportar los datos desde la base de desarrollo. Para realizar la exportación se realizó lo siguiente:

1. En PhpMyAdmin de desarrollo, verificación de que la base seleccionada es la que contiene las tablas del sistema migrado.
2. Clic en la pestaña ‘Exportar’, dejar las opciones por defecto.
3. Clic en el botón ‘Ok’.

La importación, al igual que las demás tareas mencionadas, fue conseguida mediante el uso de PhpMyAdmin. Los pasos realizados fueron los siguientes:

1. Clic en la pestaña 'Importar'.
2. Clic en el botón 'browse' del primer campo de texto editable. Selección del archivo SQL de exportación de la base del sistema. Dejar todas las opciones por defecto.
3. Clic en el botón 'Ok'.

Con estos pasos se culmina la importación de la base de datos en el nuevo ambiente.

El siguiente paso es importar la aplicación como tal y para conseguirlo se usó una cuenta FTP provista por tuWeb.us. Usando el protocolo indicado se copiaron – desde el ambiente de producción hacia el de desarrollo – todos los archivos que componen el sistema así como el marco de trabajo Yii.

Con esto se pone punto final al proceso de migración de la aplicación.

Sin embargo, una acción adicional fue la creación de cuentas de correo electrónico para cada usuario del sistema. Aprovechando el dominio adquirido y los 2Gb de almacenamiento ofrecidos por cuenta, se crearon correos electrónicas que siguen el siguiente formato:
Primera letra del primer nombre + Apellido Paterno + "@hogarcorazondemaria.com".
Ejemplo: rsarango@hogarcorazondemaria.com

Para acceder a su correo, el usuario puede hacer uso del cliente Web mail habilitado en la dirección: http://webmail.hogarcorazondemaria.com.

4.4 Capacitación

Al seguir la metodología de desarrollo Extreme Programming, los usuarios estuvieron en contacto permanente con cada nueva característica programada. Se ejecutaron pruebas y, cuando existían errores, se los corrigieron sobre la marcha.

Esta forma de trabajo permite que los usuarios finales estén familiarizados plenamente con la nueva aplicación. No obstante, no todo el personal que hará uso del sistema formó parte del equipo de desarrollo. Por tal motivo y a manera de reforzar el conocimiento que ya adquirieron los usuarios que estuvieron presentes en las pruebas, se realizaron varias sesiones de capacitación, tal como lo detalla la siguiente tabla:

Fecha	Asistentes	Tema
24-Sep	Ma. Fernanda Ferigra (Acción Social) Ma. Teresa Calle (Acción Social) Benita Mazón (Administración)	Capacitación General del Sistema.
01-Oct	Nelsi Castillo (Fisioterapia) Arnaldo Gálvez (Medicina) Ma. Angélica Pérez (Psicología) Ma. Elisa Quintana (Psicología) Hernán Yánez (Fisioterapia)	Capacitación General del Sistema.
08-Oct	Ma. Fernanda Ferigra (Acción Social) Ma. Teresa Calle (Acción Social) Benita Mazón (Administración)	Capacitación exhaustiva del módulo de Administración del Sistema.

Tabla 4-1 Sesiones de Capacitación [B]

5 CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- El sistema migrado tiene un mayor rendimiento que el anterior. Esto se debe principalmente a que el servidor donde reside la aplicación cuenta con las características adecuadas para tal fin.
- La mayor ventaja de tener un sistema en línea es el hecho de que los usuarios puedan ingresar y hacer uso de la aplicación desde cualquier computador con acceso a internet. Esto es especialmente cierto si se toma en cuenta que buena parte de los trabajadores del Hogar están físicamente presentes tan sólo los fines de semana.
- La disponibilidad del nuevo sistema es, según el proveedor de Hosting, del 99.7%. Esto equivale a 26.3 horas al año en el peor de los caos; poco más de un día. Considerando que el sistema anterior residía en una máquina de escritorio susceptible a encendido y apagado de usuarios o a efectos de corte del suministro eléctrico, las prestaciones de disponibilidad de la aplicación actual son ampliamente superiores.
- Los reportes solicitados por los Ministerios de Salud Pública y de Inclusión Económica y Social son obtenidos fácilmente desde el sistema. La generación automática de tales formularios evita que los departamentos de Medicina y Acción Social tengan que llenar los reportes de manera manual.
- Con un promedio de 250 residentes activos, no resulta fácil poder recordar los datos de cada adulto mayor. El uso del sistema permite optimizar el tiempo que invierten los usuarios en la búsqueda de pacientes y consultas.
- Las seguridades del sistema, a nivel de roles, permisos y operaciones, son altamente fiables. El uso de un módulo externo ampliamente difundido y probado, permite asegurar la confidencialidad de la información.
- Yii, El framework usado para la generación automática de Modelos, Controladores y Vistas, permitió optimizar el tiempo de desarrollo. Únicamente se programó lo realmente particular para el Hogar, mientras que el marco de trabajo se encargó de las tareas comunes y repetitivas.

5.2 Recomendaciones

- Existen referencias calculadas empíricamente en cuanto al número de pacientes que ingresan y egresan del Hogar en un periodo de tiempo dado. Con los datos ya guardados en la base de datos, esta estadística puede dejar de ser un estimado para convertirse en un dato 100% fidedigno. Se pueden crear vistas a nivel de base de datos para obtener esta información.
- El sistema almacena detalles específicos relacionados con el origen de cada adulto mayor. Esta información podría permitir obtener reportes detallados acerca del número de ancianos residentes del Hogar por sector del país, así como el porcentaje que este indicador representa dentro de la institución.
- Si bien el objetivo principal de crear el sistema fue mantener una historia clínica de cada paciente del Hogar, el sistema puede satisfacer nuevas necesidades y requerimientos, por ejemplo:
    - Crear un módulo que permita gestionar cabalmente los medicamentos que reciben los ancianos.
    - Incluir datos psicológicos y fisioterapéuticos. El sistema podría ser extendido para que los datos relevantes de estas áreas puedan ser parte de nuevas clases.
    - Añadir un módulo que satisfaga las necesidades exclusivas del departamento de Acción Social. Esto podría ayudar enormemente a los encargados de brindar una mejor ayuda al adulto mayor.
    - Guardar las acciones más relevantes como pistas de auditoría. Tal funcionalidad podría ser parte de un nuevo módulo de la aplicación.
    - Desarrollar un módulo financiero que permita controlar los ingresos y egresos de la institución.
- Este mismo sistema podría ser usado – con las personalizaciones adecuadas – por otros Centros Gerontológicos.
- La contratación del servicio de hosting y compra del dominio tiene vigencia de un año a partir del 13 de Julio del 2011. Se debería realizar la respectiva renovación antes de su caducidad.
- Se recomienda que el mantenimiento del sistema quede a cargo de estudiantes de Ingeniería en Sistemas de niveles superiores; preferiblemente con conocimientos de

PHP y MySql.

- Las circunstancias actuales del Hogar han cambiado. La directora del Centro Gerontológico y principal impulsadora del uso del sistema, fue asignada a otro país. Por este motivo se recomienda al departamento de Acción Social de la Pontificia Universidad Católica del Ecuador, dar seguimiento a las actividades y relaciones con esta institución e impulsar el uso del nuevo sistema.
- A pesar de que la confidencialidad de la información es alta por la distribución de roles, permisos y operaciones, el sistema podría tener vulnerabilidades de seguridad en términos de SQL Injection. Se recomienda encriptar la clave con MD5 hash o similares.

BIBLIOGRAFÍA Y REFERENCIAS

Documentos Impresos

[A] Formulario HCU-form.057/2010
2010
Ministerio de Salud Pública del Ecuador

[B] MIGRACIÓN DEL SISTEMA DE FICHAS MÉDICAS DEL HOGAR CORAZÓN DE MARÍA DE JBOSS Y JAVA SOBRE WINDOWS A APACHE Y PHP SOBRE LINUX.
Pontificia Universidad Católica del Ecuador
2011
Roy Sarango

[C] SISTEMA DE ADMINISTRACIÓN DE FICHAS MÉDICAS – MANUAL DE USUARIO
Pontificia Universidad Católica del Ecuador
2010
Santiago Borja

[D] POO Y MVC EN PHP
2011
Eugenia Bahit

Páginas Web

[1] Ministerio de Inclusión Económica y Social
http://www.mies.gob.ec/index.php?option=com_content&view=article&id=65:-permiso-de-funcionamiento-para-centros-gerontologicos&catid=13&Itemid=54
31 de Agosto de 2011

[2] Hogar Corazón de María, Ecuador
http://hogarcorazondemaria.blogspot.com/2011/03/residencias-y-casas-de-la-congregacion.html
1 de Septiembre de 2011

[3] Sistema Estatal (Mexicano) para el Desarrollo Integral de la Familia – Dirección de Atención a Población Vulnerable
http://www.dif.gob.mx/adultosmayores/media/1.pdf
1 de Septiembre de 2011

[4] Wikipedia (2011). Menarquía
http://es.wikipedia.org/wiki/Menarquia
1 de Septiembre de 2011

[5] Edusex (2011). Menarquía y Ciclo Menstrual
http://www.edusex.cl/capitulos/ciclo-menarquia.php
1 de Septiembre de 2011

[6] Wikipedia (2011). Dismenorrea
http://es.wikipedia.org/wiki/Dismenorrea
1 de Septiembre de 2011

[7] Centros para el Control y la Prevención de Enfermedades (2011). Antecedentes Familiares de Enfermedad
http://www.cdc.gov/genomics/spanish/famhistory/
3 de Septiembre de 2011

[8] Adictos al Trabajo (2011). PHP Vs. Java
http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=PHPVsJava
4 de Septiembre de 2011

[9] Wikipedia (2011). Normalización de Base de Datos
http://es.wikipedia.org/wiki/Normalizaci%C3%B3n_de_bases_de_datos
5 de Septiembre de 2011

[10] Yii Framework (2011). SRBAC
http://www.yiiframework.com/extension/srbac
5 de Septiembre de 2011

[11] Wikipedia. Modelo Vista Controlador
http://es.wikipedia.org/wiki/Modelo_Vista_Controlador
7 de Septiembre de 2011

[12] Yii Framework. Características
http://www.yiiframework.com/features
7 de Septiembre de 2011

[13] Apache Friends. XAMPP
http://www.apachefriends.org/en/xampp.html
7 de Septiembre de 2011

[14] Yii Framework. Creating Your First Yii Application
http://www.yiiframework.com/doc/guide/1.1/en/quickstart.first-app
7 de Septiembre de 2011

[15] Yii Framework. TCPDF
http://www.yiiframework.com/extension/tcpdf/
9 de Septiembre de 2011

[16] Yii Framework. Calendar
http://www.yiiframework.com/extension/calendar/
9 de Septiembre de 2011

[17] Yii Framework. Image
http://www.yiiframework.com/extension/image/
9 de Septiembre de 2011

[18] MySql. LOAD DATA INFILE
http://dev.mysql.com/doc/refman/5.0/en/load-data.html
11 de Septiembre de 2011

GLOSARIO

A

- **Apache**: es un servidor Web de código abierto y multiplataforma que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Es propiedad de Apache Software Foundation.

C

- **CMS**: Content Management System, es un sistema que como su nombre lo indica, permite administrar contenidos en un medio digital. Entre las tareas de administración destacan las siguientes: crear, clasificar y publicar cualquier tipo de información en una página Web**.**

- **CRUD**: Create Read Update Delete, es el acrónimo de las 4 operaciones básicas aplicadas sobre registros de base de datos: Crear, Obtener, Actualizar y Borrar.

D

- **DDL**: Data Definition Language, es un lenguaje basado en sentencias SQL que permite definir las estructuras que almacenarán la información.

- **DML**: Data Manipulation Language, es un leguaje basado en sentencias SQL que permite seleccionar, insertar, modificar o eliminar información de una base de datos.

- **Dump**: es un tipo de archivo que contiene sentencias DDL y DML en los cuales se basa la estructura y la información de una base de datos.

F

- **FK**: Foreign Key, es una restricción a nivel de tablas de base de datos que indica

que una clave primaria de un tabla ajena viene a ser heredada por la tabla en cuestión. De esta manera se puede establecer una relación entre ambas tablas; misma que usualmente tiene cardinalidad de uno a muchos. En este tipo de restricciones se basa la integridad referencial de base de datos relacionales.

- **Fedora**: es una distribución de Linux basada en la popular Red Hat pero sin soporte comercial. Maneja el mismo tipo de paquetes RPM y su interfaz amigable con el usuario la ha convertido en una de las mejores opciones entre las diferentes distros disponibles en el mundo Linux.

I

- **InnoDB**: es una tecnología de almacenamiento de datos en el motor MySql. Aunque el rendimiento suele ser menor que el de MyISAM, sus prestaciones relacionales son mayores, pues permite mantener integridad referencial entre tablas y soporta el concepto de atomicidad.

- **Intranet**: es una red de computadores privados que utiliza tecnología de Internet para compartir diferentes recursos entre dispositivos.

- **IP**: Internet Protocol, es el número que identifica a cada dispositivo dentro de una red de computadores.

L

- **Localhost**: es un nombre reservado de dispositivos informáticos que pueden pertenecer a una red. Tal reservación refiere al dispositivo en sí.

- **Linux**: es un sistema operativo tipo Unix. Por ser de código abierto, es ampliamente usado para evitar el pago de licencias. Además, sus características de robustez, agilidad, seguridad y rendimiento, hacen de él una dura competencia para Sistemas Operativos comerciales.

M

- **MER**: Modelo Entidad Relación, es una herramienta para el modelado de datos de un sistema de información. Estos modelos expresa entidades relevantes para un sistema de información así como sus interrelaciones y propiedades.

- **MVC**: Modelo Vista Controlador, es un patrón de la arquitectura de software que separa los datos de la aplicación, la interfaz de usuario, y la lógica del negocio en tres componentes distintos: Modelo, representación específica de la información con la cual opera el sistema. Vista, forma de presentación de la información. Controlador, manejador de eventos de usuario.

- **MyISAM**: es una tecnología de almacenamiento de datos en el motor MySql. Su principal característica es la gran velocidad que obtiene en las consultas ya que no tiene que hacer comprobaciones de integridad referencial, ni bloquear tablas para hacer lectura de ellas.

- **MySql**: es un sistema de gestión de base de datos, propiedad de Oracle Corporation. Entre sus características se destacan su modo de trabajo multiusuario y multiplataforma. Además, es de código abierto.

- **MySql Administrator**: Es un software provisto por la Corporación Oracle y que permite realizar múltiples actividades de administración de base de datos, desde simples sentencias en lenguaje SQL hasta gestión de permisos y respaldos de esquemas. Como su nombre lo indica, tales tareas son aplicables sobre el motor MySql.

- **Mysqld**: es el archivo de configuración del sistema gestor de base de datos MySql.

O

- **Oracle**: es una de las empresas más grandes de la industria del software.

Usualmente, 'Oracle' también se refiere al principal producto comercial de esta empresa, su sistema gestor de base de datos.

P

- **PHP**: Hypertext Pre-processor, es un lenguaje de programación interpretado diseñado para la creación de páginas Web dinámicas. Su sintaxis es similar a la de otros lenguajes ampliamente conocidos como C++ o Java. PHP es de uso libre.

- **PhpMyAdmin**: es una herramienta escrita en PHP cuya intención es permitirle a los usuarios de la base de datos, contar con una interfaz gráfica en la Web que colabore en diversas tareas relativas a la administración del motor MySql.

- **PhpUnit**: es un framework de PHP que permite crear y ejecutar juegos de tests unitarios de manera sencilla. Entre otras acciones, PhpUnit analiza los resultados de tales pruebas y guarda los resultados en una base de datos para posteriormente ser consultados de una manera legible para los técnicos programadores.

- **PK**: Primary Key, es una restricción a nivel de tablas de base de datos que indica que esa columna o conjunto de columnas identifican de manera única a la tabla y que el valor o valores que la componen no pueden ser nulos.

- **Perl**: Practical Extracting and Reporting Language, es un lenguaje de programación muy utilizado para construir aplicaciones CGI para el Web. Perl es de uso libre y en la actualidad también es multiplataforma aunque originalmente fue concebido para Unix.

R

- **Root**: Root, en el ámbito informático, suele referirse al súper usuario del software en cuestión. Es ampliamente conocido en sistemas operativas tipo Unix y en la base de datos MySql.

S

- **SQL**: Structure Query Language, es un lenguaje diseñado para administrar cualquier tipo de información referente a una base de datos relacional.

- **SRBAC**: Role Base Access Control, es un módulo que añade una interfaz gráfica al sub sistema de autorización y autenticación del framework Yii. Se basa en la clase CDbAuthManager, propia del marco de trabajo, por lo que requiere de ciertas tablas de base de datos para funcionar correctamente.

U

- **UK**: Unique Key, es una restricción a nivel de tablas de base de datos que sirve para definir un índice único sobre una columna o combinación de columnas. Sin embargo, se permiten valores nulos en uno o varios componentes de la restricción.

- **UNIX**: es un sistema operativo comercial, multiusuario y multiplataforma. Sus características hacen de ésta, una de las plataformas más seguras y eficientes del mercado.

X

- **XAMPP**: es una distribución de Apache que además de contener el popular Web server, también incluye la base de datos MySql y los lenguajes de programación PHP y Perl. El empaquetado de todas estas aplicaciones y su sencilla instalación, han popularizado fuertemente a esta línea de ensamblaje de ambientes de desarrollo.

ANEXOS

ANEXO 1: Nueva Ficha Médica solicitada por el MIES..... 106
ANEXO 2: HCU-form.001/2010 121
ANEXO 3: HCU-form.057/2010 123
ANEXO 4: Escalas Geriátricas..... 126

ANEXO 1: Nueva Ficha Médica solicitada por el MIES

Nº	Sección	Sub Sección (Si Aplica)	Campo	Tipo / Opciones
1	Motivo de Consulta		Motivo	Texto
2	Enfermedad o Problema Actual		Enfermedad	Texto
			Medicamento que Recibe	Texto
			Estado General	Lista: • Dependiente • Frágil • Independiente
3	Revisión Actual de Sistemas		Visión	Check box simple
			Audición	Check box simple
			Olfato y Gusto	Check box simple
			Respiratorio	Check box simple
			Digestivo	Check box simple
			Cardiovascular	Check box simple
			Genitourinario	Check box simple
			Musculo esquelético	Check box simple
			Endócrino	Check box simple
			Hemo linfático	Check box simple
			Observaciones	Texto

4	Antecedentes Personales	Alertas de Riesgo	Caída	Check box simple
			Dismovilidad	Check box simple
			Pérdida de Peso	Check box simple
			Astenia	Check box simple
			Desorientación	Check box simple
			Alteración del Comportamiento	Check box simple
		Generales	Inmunización	Check box simple
			Higiene General	Check box simple
			Higiene Oral	Check box simple
			Ejercicio	Check box simple
			Alimentación	Check box simple
			Actividad Recreacional	Check box simple
			Cont. Salud	Check box simple
			Alergias	Check box simple
			Otros	Check box simple
		Hábitos Nocivos	Tabaquismo	Check box simple
			Alcoholismo	Check box simple
			Adicciones	Check box simple
			Otro Hábito	Check box simple
		Clínico	Dermatológicos	Check box simple

		Quirúrgicos	Visuales	Check box simple
			Otorrino	Check box simple
			Estomatológicos	Check box simple
			Endócrinos	Check box simple
			Cardiovasculares	Check box simple
			Respiratorios	Check box simple
			Digestivos	Check box simple
			Neurológicos	Check box simple
			Urológicos	Check box simple
			Hemo linfáticos	Check box simple
			Infecciosos	Check box simple
			Oncológicos	Check box simple
			Musculo esqueléticos	Check box simple
			Psiquiátricos	Check box simple
		Gineco Obstétricos	Edad de Menopausia	Texto
			Edad Última Mamografía	Texto
			Edad Última Citología	Texto
			Número de Embarazos	Texto
			Número de Partos Naturales	Texto
			Número de Cesáreas	Texto

			Terapia Hormonal	Check box simple
		Andrológicos	Edad Última Prueba APE	Texto
			Terapia Hormonal	Check box simple
		Farmacológicos	Aines	Check box simple
			Analgésicos	Check box simple
			Antidiabéticos	Check box simple
			Antihipertensivos	Check box simple
			Anticoagulantes	Check box simple
			Psicofármacos	Check box simple
			Antibióticos	Check box simple
			Otros	Check box simple
			Número de Prescriptores	Texto
		Notas	Observaciones	Texto
5	Antecedentes Familiares y Sociales		Cardiopatías	
			Diabetes	Check box simple
			Hipertensión	Check box simple
			Neoplasia	Check box simple
			Alzheimer	Check box simple
			Parkinson	Check box simple
			Tuberculosis	Check box simple

			Violencia Intrafamiliar	Check box simple
			Síndrome del Cuidador	Check box simple
			Observaciones	Texto
6	Signos Vitales y Antropometría		Presión Arterial Acostado	Texto
			Presión Arterial Sentado	Texto
			Temperatura º C	Texto
			Pulso / minuto	Texto
			Frecuencia Respiratoria / minuto	Texto
			Peso / Kilogramos	Texto
			Talla / Centímetros	Texto
			IMC	Texto
			Perímetro Cintura	Texto
			Perímetro Cadera	Texto
			Perímetro Pantorrilla	Texto
7	Examen Físico	Regional	Piel	Check box simple
			Cabeza	Check box simple
			Ojos	Check box simple
			Oídos	Check box simple
			Boca	Check box simple
			Nariz	Check box simple

			Cuello	Check box simple
			Axila Mama	Check box simple
			Tórax	Check box simple
			Abdomen	Check box simple
			Columna	Check box simple
			Periné	Check box simple
			Miembros Superiores	Check box simple
			Miembros Inferiores	Check box simple
		Sistémico	Sentidos	Check box simple
			Respiratorio	Check box simple
			Cardiovascular	Check box simple
			Digestivo	Check box simple
			Genitourinario	Check box simple
			Musculo esquelético	Check box simple
			Endócrino	Check box simple
			Hemo linfático	Check box simple
			Neurológico	Check box simple
		Notas	Observaciones	Texto
8	Escalas Geriátricas	Tamizaje Rápido	Dificultad Visual	Check box simple

			Dificultad Auditiva	Check box simple
			“Levántate y Anda” mayor a 15 segundos	Check box simple
			Pérdida Involuntaria de Orina	Check box simple
			Pérdida de Memoria Reciente	Check box simple
			Pierde peso más de 4.5 Kg en 6 meses	Check box simple
			Se siente triste	Check box simple
			Puede Bañarse Solo	Check box simple
			Sale de Compras Solo	Check box simple
		Actividades Básicas	Se Baña Solo	Radio Button: • Independiente • Dependiente Parcial • Dependiente Total
			Se Viste / Desviste Solo	Radio Button • Independiente • Dependiente Parcial • Dependiente Total
			Cuida su Apariencia Personal	Radio Button: • Independiente • Dependiente Parcial • Dependiente Total

			Utiliza el Inodoro	Radio Button: • Independiente • Dependiente Parcial • Dependiente Total
			Controla Esfínteres	Radio Button: • Independiente • Dependiente Parcial • Dependiente Total
			Se Traslada, Acuesta, Levanta	Radio Button: • Independiente • Dependiente Parcial • Dependiente Total
			Camina	Radio Button: • Independiente • Dependiente Parcial • Dependiente Total
			Se Alimenta	Radio Button: • Independiente • Dependiente Parcial • Dependiente Total

		Actividad Instrumental	Usa el Teléfono	Radio Button: • Independiente • Dependiente Parcial • Dependiente Total
			Usa Medios de Transporte	Radio Button: • Independiente • Dependiente Parcial • Dependiente Total
			Va de Compras	Radio Button: • Independiente • Dependiente Parcial • Dependiente Total
			Prepara la Comida	Radio Button: • Independiente • Dependiente Parcial • Dependiente Total
			Controla su Medicina	Radio Button: • Independiente • Dependiente Parcial • Dependiente Total

			Maneja Dinero	Radio Button: • Independiente • Dependiente Parcial • Dependiente Total
		Cognitivo	Sabe La Fecha	Radio Button • Año, Mes, Día, Semana • Año, Mes, Día • Año, Mes • Año • No puede
			Aprende el Nombre de 3 Objetos	Radio Button: • 3 Objetos • 2 Objetos • 1 Objeto • No puede
			Repite Números al Revés	Radio Button: • 5 Números • 4 Números • 3 Números • 2 Números

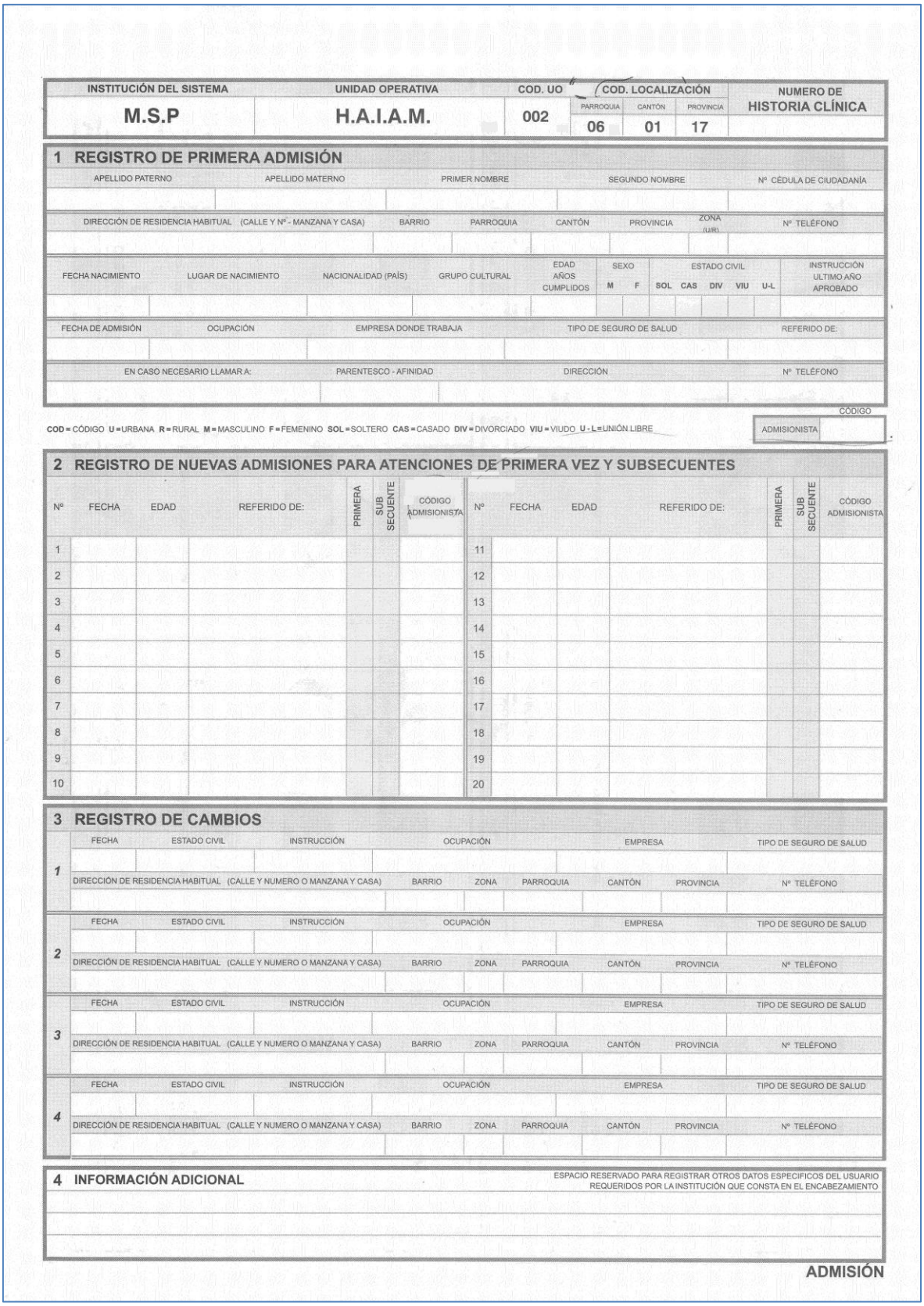
				• 1 Número • No puede
			Toma, Dobla y Coloca Papel	Radio Button: • Toma, Dobla y Coloca • Toma y Dobla • Toma • No puede
			Repite Serie de 3 Palabras	Radio Button: • 3 Palabras • 2 Palabras • 1 Palabra • No puede
			Copia Dibujo de 2 Círculos Cruzados	• Si Puede • No Puede
		Recurso Social	Situación Familiar: Vive Con	Radio Button: • Familia o Pareja, Sin Conflictos • Pareja de Edad Similar • Familia o Pareja, Con Conflictos • Solo, Familia no Cubre Necesidades • Solo, Desatendido, Sin Familia

			Relaciones y Contactos Sociales	Radio Button: • Con Todos, Sale • Con Familia y Amigos, Sale • Con Familia, Sale • Con Familia o Visitas, No Sale • Con Nadie, No Sale
			Apoyo de la Red Social	Radio Button: • No Necesita • De Familia o Vecinos • Formal Suficiente • Formal Insuficiente • Informal • No Recibe y Necesita
		Depresión	Está Satisfecho con su Vida	Check box simple
			Ha Dejado de Hacer Actividades de Interés	Check box simple
			Siente su Vida Vacía	Check box simple
			Se Aburre con Frecuencia	Check box simple
			Está de Buen Ánimo la Mayor Parte del Tiempo	Check box simple
			Está Preocupado Porque Algo Malo va a Sucederle	Check box simple
			Se Siente Feliz la Mayor Parte del Tiempo	Check box simple

			Se Siente a Menudo Desamparado	Check box simple
			Prefiere Estar en Casa a Salir a Actividades Nuevas	Check box simple
			Tiene más Problemas de Memoria que los Demás	Check box simple
			Cree que es Maravilloso Estar Vivo	Check box simple
			Se Siente Inútil	Check box simple
			Se Siente Lleno de Energía	Check box simple
			Se Siente sin Esperanza Ante la Situación Actual	Check box simple
			Siente que la Mayoría de la Gente está Mejor que Él/Ella	Check box simple
		Nutricional	Diminución de Ingesta en Último Trimestre	Radio Button: • Severa • Moderada • Ausente
			Pérdida de Peso en Ultimo Trimestre	Radio Button: • Pierde > 3Kg o Pesa < 45,5 Kg • No Sabe • De 1 a 3Kg • No Pierde

			Índice de Masa Corporal	Radio Button: • < 19 • 19 a 20 • 21 a 22 • 23 o Más
			Inmovilidad	Radio Button: • Cama o Silla • Se Levanta, No sale • Va afuera
			Problema Neuropsicológico (Demencia o Depresión)	Check box simple
			Enfermedad Aguda en Ultimo Trimestre	Radio Button: • Severa • Moderada • Ausente

ANEXO 2: HCU-form.001/2010



INSTITUCIÓN DEL SISTEMA	UNIDAD OPERATIVA	COD. UO	COD. LOCALIZACIÓN			NUMERO DE HISTORIA CLÍNICA
			PARROQUIA	CANTÓN	PROVINCIA	
M.S.P	H.A.I.A.M.	002	06	01	17	

1 REGISTRO DE PRIMERA ADMISIÓN

APELLIDO PATERNO	APELLIDO MATERNO	PRIMER NOMBRE	SEGUNDO NOMBRE	Nº CÉDULA DE CIUDADANÍA

DIRECCIÓN DE RESIDENCIA HABITUAL (CALLE Y Nº - MANZANA Y CASA)	BARRIO	PARROQUIA	CANTÓN	PROVINCIA	ZONA (U/R)	Nº TELÉFONO

FECHA NACIMIENTO	LUGAR DE NACIMIENTO	NACIONALIDAD (PAÍS)	GRUPO CULTURAL	EDAD AÑOS CUMPLIDOS	SEXO		ESTADO CIVIL					INSTRUCCIÓN ULTIMO AÑO APROBADO
					M	F	SOL	CAS	DIV	VIU	U-L	

FECHA DE ADMISIÓN	OCUPACIÓN	EMPRESA DONDE TRABAJA	TIPO DE SEGURO DE SALUD	REFERIDO DE:

EN CASO NECESARIO LLAMAR A:	PARENTESCO - AFINIDAD	DIRECCIÓN	Nº TELÉFONO

COD = CÓDIGO U = URBANA R = RURAL M = MASCULINO F = FEMENINO SOL = SOLTERO CAS = CASADO DIV = DIVORCIADO VIU = VIUDO U-L = UNIÓN LIBRE

	CÓDIGO
ADMISIONISTA	

2 REGISTRO DE NUEVAS ADMISIONES PARA ATENCIONES DE PRIMERA VEZ Y SUBSECUENTES

Nº	FECHA	EDAD	REFERIDO DE:	PRIMERA	SUB SECUENTE	CÓDIGO ADMISIONISTA	Nº	FECHA	EDAD	REFERIDO DE:	PRIMERA	SUB SECUENTE	CÓDIGO ADMISIONISTA
1							11						
2							12						
3							13						
4							14						
5							15						
6							16						
7							17						
8							18						
9							19						
10							20						

3 REGISTRO DE CAMBIOS

	FECHA	ESTADO CIVIL	INSTRUCCIÓN	OCUPACIÓN	EMPRESA	TIPO DE SEGURO DE SALUD
1						
	DIRECCIÓN DE RESIDENCIA HABITUAL (CALLE Y NUMERO O MANZANA Y CASA)	BARRIO	ZONA	PARROQUIA	CANTÓN	PROVINCIA	Nº TELÉFONO
2						
	DIRECCIÓN DE RESIDENCIA HABITUAL (CALLE Y NUMERO O MANZANA Y CASA)	BARRIO	ZONA	PARROQUIA	CANTÓN	PROVINCIA	Nº TELÉFONO
3						
	DIRECCIÓN DE RESIDENCIA HABITUAL (CALLE Y NUMERO O MANZANA Y CASA)	BARRIO	ZONA	PARROQUIA	CANTÓN	PROVINCIA	Nº TELÉFONO
4						
	DIRECCIÓN DE RESIDENCIA HABITUAL (CALLE Y NUMERO O MANZANA Y CASA)	BARRIO	ZONA	PARROQUIA	CANTÓN	PROVINCIA	Nº TELÉFONO

4 INFORMACIÓN ADICIONAL

ESPACIO RESERVADO PARA REGISTRAR OTROS DATOS ESPECIFICOS DEL USUARIO REQUERIDOS POR LA INSTITUCIÓN QUE CONSTA EN EL ENCABEZAMIENTO

ADMISIÓN

ANEXO 3: HCU-form.057/2010

ESTABLECIMIENTO	APELLIDOS Y NOMBRES	EDAD	VIVE CON	OCUPACION ANTERIOR / OCUPACION ACTUAL	N° HISTORIA CLÍNICA
H.A.I.A.M.					

1 MOTIVO DE CONSULTA — INFORMANTE: USUARIO ☐ CUIDADOR ☐

2 ENFERMEDAD O PROBLEMA ACTUAL — CRONOLOGÍA, LOCALIZACIÓN, CARACTERÍSTICAS, INTENSIDAD, CAUSA APARENTE, FACTORES QUE AGRAVAN O MEJORAN, SÍNTOMAS ASOCIADOS, EVOLUCIÓN, RESULTADOS DE EXÁMENES ANTERIORES, CONDICIÓN ACTUAL, ATIPIAS

MEDICAMENTOS QUE RECIBE			
ESTADO GENERAL	INDEPENDIENTE ○	FRAGIL ○	DEPENDIENTE ☐

3 REVISIÓN ACTUAL DE SISTEMAS — CIRCULO= CON PATOLOGIA, DESCRIBIR ANOTANDO EL NUMERO; CUADRADO= SIN PATOLOGIA, NO DESCRIBIR

1. VISION ○☐	2. AUDICION ○☐	3. OLFATO Y GUSTO ○☐	4. RESPIRATORIO ○☐	5. CARDIO VASCULAR ○☐	6. DIGESTIVO ○☐
8. GENITO - URINARIO ○☐	9. MUSCULO ESQUELETICO ○☐	10. ENDOCRINO ○☐	11. HEMO LINFATICO ○☐	12. NERVIOSO ○☐	

4 ANTECEDENTES PERSONALES — CUADRADO= SIN PROBLEMAS, NO DESCRIBIR; CIRCULO= CON PROBLEMAS, DESCRIBIR ABAJO, REGISTRANDO EL NÚMERO

ALERTAS DE RIESGO	1. CAIDA ○	2. DISMOVILIDAD ○	3. PERDIDA DE PESO ○	4. ASTENIA ○	5. DESORIENTACION ○	6. ALTERACION DEL COMPORTAMIENTO ○
GENERALES	1. INMUNIZACIONES ○☐	2. HIGIENE GENERAL ○☐	3. HIGIENE ORAL ○☐	4. EJERCICIO ○☐	5. ALIMENTACION ○☐	6. ACTIVIDAD RECREATIVA ○☐
7. CONTROLES DE SALUD ○☐	8. ALERGIAS ○☐	9. OTROS ○☐				
HABITOS NOCIVOS	1. TABAQUISMO ○	2. ALCOHOLISMO ○	3. ADICCIONES ○	4. OTRO HABITO ○		
CLINICO QUIRURGICOS	1. DERMATO LOGICOS ○	2. VISUALES ○	3. OTORRINO ○	4. ESTOMATO LOGICOS ○	5. ENDOCRINOS ○	6. CARDIO VASCULARES ○
7. RESPIRATORIOS ○	8. DIGESTIVOS ○	9. NEUROLÓGICOS ○	10. UROLOGICOS ○	11. HEMO LINFATICOS ○	12. INFECCIOSOS ○	13. ONCOLOGICOS ○
14. MUSCULO ESQUELETICOS ○	15. PSIQUIATRICOS ○					

GINECO OBSTETRICOS	1. EDAD DE MENOPAUSIA		2. EDAD DE ULTIMA MAMOGRAFIA		3. EDAD DE ULTIMA CITOLOGIA		4. EMBARAZOS		5. PARTOS		6. CESÁREAS		7. TERAPIA HORMONAL	
ANDROLOGICOS	1. EDAD ULTIMA PRUEBA APE		2. TERAPIA HORMONAL										
FARMACO LOGICOS	1. AINES ○		2. ANALGESICOS ○		3. ANTI DIABETICOS ○		4. ANTI HIPERTENSIVOS ○		5. ANTI COAGULANTES ○		6. PSICO FARMACOS ○		
7. ANTIBIOTICOS ○	8. OTROS ○☐		9. NUMERO DE PRESCRIPTORES										

5 ANTECEDENTES FAMILIARES Y SOCIALES — CIRCULO: CON PATOLOGIA, DESCRIBIR ANOTANDO EL NUMERO; CUADRADO= SIN PATOLOGIA, NO DESCRIBIR

1. CARDIO PATÍAS ○☐	2. DIABETES ○☐	3. HIPER TENSION ○☐	4. NEOPLASIA ○☐	5. ALZHEIMER ○☐	6.PARKINSON ○☐	7. TUBERCULOSIS ○☐
8. VIOLENCIA INTRAFAMILIAR ○☐	10. SINDROME DEL CUIDADOR ○☐					

ATENCION AL ADULTO MAYOR (1)



6 SIGNOS VITALES, ANTROPOMETRIA Y TAMIZAJE

P.ARTERIAL ACOSTADO	P. ARTERIAL SENTADO	TEMPERATURA °C	PULSO / min	FRECUENCIA RESPIR. / min	PESO / Kg	TALLA / cm	IMC	PERIMETRO CINTURA	PERIMETRO CADERA	PERIMETRO PANTORRILLA	RESPONSABLE (SIGLAS)

TAMIZAJE RAPIDO	1. DIFICULTAD VISUAL ○	2. DIFICULTAD AUDITIVA ○	3. "LEVANTATE Y ANDA" MAYOR A 15s ○	4. PERDIDA INVOLUNT. DE ORINA ○	5. PERDIDA DE MEMORIA RECIENTE ○	6. PIERDE PESO MAS DE 4,5 KG EN 6 M. ○
7. SE SIENTE TRISTE O DEPRIMIDA ○	8. PUEDE BAÑARSE SOLO ○	9. SALE DE COMPRAS SOLO ○	10. VIVE SOLO ○			RESPONSABLE (SIGLAS)

7 EXAMEN FÍSICO

CIRCULO= CON PATOLOGIA, DESCRIBIR CON EL NUMERO
CUADRADO= SIN PATOLOGIA, NO DESCRIBIR

			REGIONAL (1 - 14)		SISTEMICO (1 - 9)		
1. PIEL ○ □	5. BOCA ○ □	9. TÓRAX ○ □	12. PERINÉ ○ □	1. ÓRG. DE LOS SENTIDOS ○ □	4. DIGESTIVO ○ □	7. ENDOCRINO ○ □
2. CABEZA ○ □	6. NARIZ ○ □	10. ABDOMEN ○ □	13. M. SUPERIORES ○ □	2. RESPIRATORIO ○ □	5. GENITO URINARIO ○ □	8. HEMO LINFATICO ○ □
3. OJOS ○ □	7. CUELLO ○ □	11. COLUMNA ○ □	14. M. INFERIORES ○ □	3. CARDIO VASCULAR ○ □	6. MÚSCULO ESQUELETICO ○ □	9. NEUROLOGICO ○ □
4. OÍDOS ○ □	8. AXILA - MAMA ○ □					

8 DIAGNOSTICOS

		CIE	PRESUNTIVO / DEFINITIVO	CLINICO, SINDRÓMICO, PSICOLOGICO, FUNCIONAL, NUTRICIONAL
1				
2				
3				
4				
5				

SINDROMES GERIATRICOS

FRAGILIDAD	○	DISMOVILIDAD	○
DEPRESION	○	CAIDA	○
DELIRIO	○	MALNUTRICION	○
ULCERAS POR PRESION	○	DEMENCIA	○
INCONTINENCIA	○	IATROGENIA	○

9 PRUEBAS DIAGNOSTICAS

REGISTRAR LOS EXAMENES DE LABORATORIO Y ESPECIALES SOLICITADOS

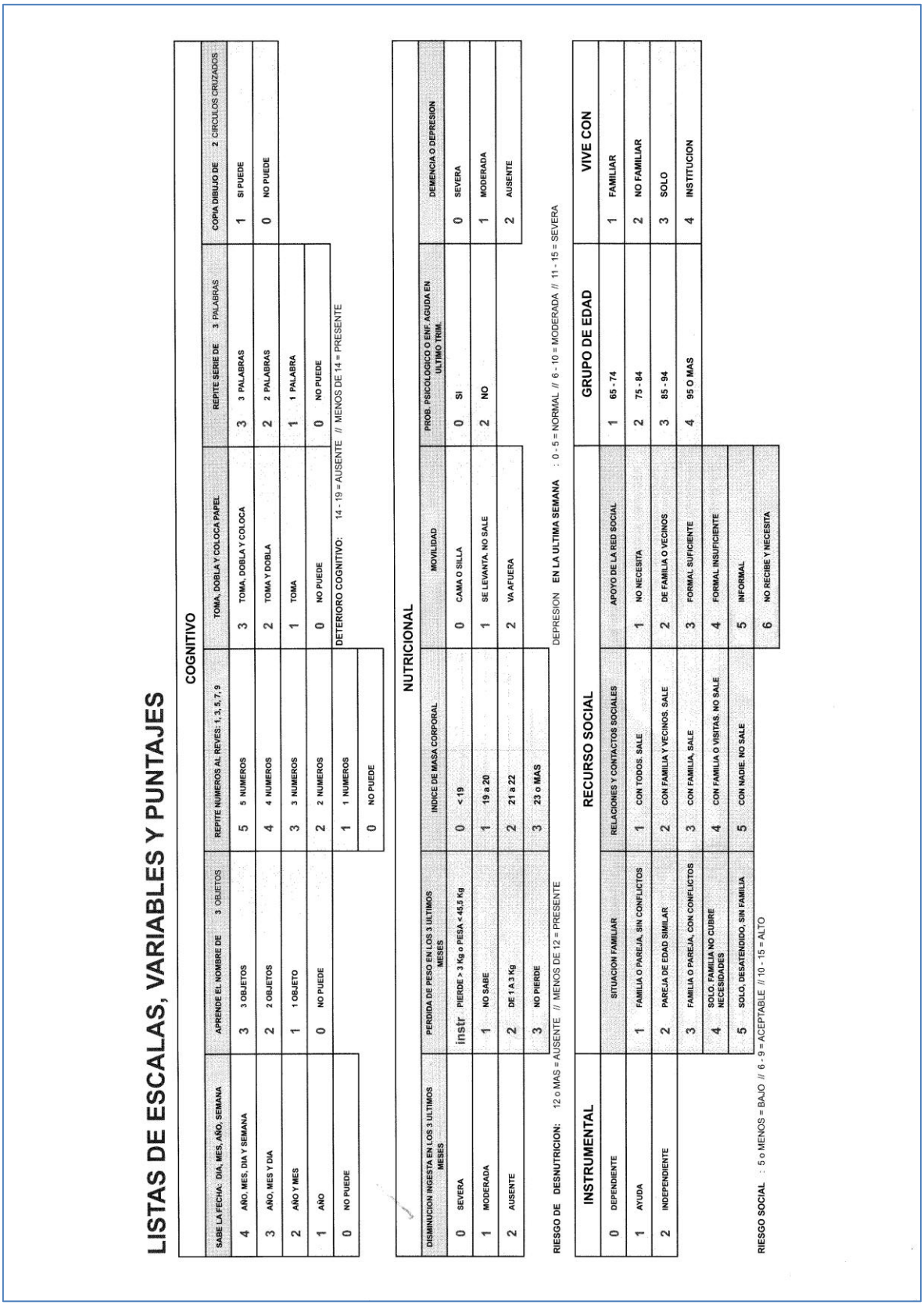
10 TRATAMIENTO

1. FUNCIONAL, 2. NUTRICIONAL, 3. PSICOLOGICO, 4. SOCIAL, 5. EDUCATIVO, 6. FARMACOLOGICO

PROFESIONAL	FIRMA	CODIGO	FECHA PROXIMA CITA	FECHA Y HORA DE ATENCION	HOJA

ATENCION GERIATRICA (2)

ANEXO 4: Escalas Geriátricas



LISTAS DE ESCALAS, VARIABLES Y PUNTAJES

COGNITIVO

SABE LA FECHA: DIA, MES, AÑO, SEMANA	APRENDE EL NOMBRE DE 3 OBJETOS	REPITE NUMEROS AL REVES: 1, 3, 5, 7, 9	TOMA, DOBLA Y COLOCA PAPEL	REPITE SERIE DE 3 PALABRAS	COPIA DIBUJO DE 2 CIRCULOS CRUZADOS
4 AÑO, MES, DIA Y SEMANA	3 3 OBJETOS	5 5 NUMEROS	3 TOMA, DOBLA Y COLOCA	3 3 PALABRAS	1 SI PUEDE
3 AÑO, MES Y DIA	2 2 OBJETOS	4 4 NUMEROS	2 TOMA Y DOBLA	2 2 PALABRAS	0 NO PUEDE
2 AÑO Y MES	1 1 OBJETO	3 3 NUMEROS	1 TOMA	1 1 PALABRA	
1 AÑO	0 NO PUEDE	2 2 NUMEROS	0 NO PUEDE	0 NO PUEDE	
0 NO PUEDE		1 1 NUMEROS	DETERIORO COGNITIVO: 14 - 19 = AUSENTE // MENOS DE 14 = PRESENTE		
		0 NO PUEDE			

NUTRICIONAL

DISMINUCION INGESTA EN LOS 3 ULTIMOS MESES	PERDIDA DE PESO EN LOS 3 ULTIMOS MESES	INDICE DE MASA CORPORAL	MOVILIDAD	PROB. PSICOLOGICO O ENF. AGUDA EN ULTIMO TRIM.	DEMENCIA O DEPRESION
0 SEVERA	instr PIERDE > 3 Kg o PESA < 45,5 Kg	0 < 19	0 CAMA O SILLA	0 SI	0 SEVERA
1 MODERADA	1 NO SABE	1 19 a 20	1 SE LEVANTA, NO SALE	2 NO	1 MODERADA
2 AUSENTE	2 DE 1 A 3 Kg	2 21 a 22	2 VA AFUERA		2 AUSENTE
	3 NO PIERDE	3 23 o MAS			

RIESGO DE DESNUTRICION: 12 o MAS = AUSENTE // MENOS DE 12 = PRESENTE DEPRESION EN LA ULTIMA SEMANA : 0 - 5 = NORMAL // 6 - 10 = MODERADA // 11 - 15 = SEVERA

INSTRUMENTAL

0	DEPENDIENTE
1	AYUDA
2	INDEPENDIENTE

RECURSO SOCIAL

SITUACION FAMILIAR	RELACIONES Y CONTACTOS SOCIALES	APOYO DE LA RED SOCIAL
1 FAMILIA O PAREJA, SIN CONFLICTOS	1 CON TODOS, SALE	1 NO NECESITA
2 PAREJA DE EDAD SIMILAR	2 CON FAMILIA Y VECINOS, SALE	2 DE FAMILIA O VECINOS
3 FAMILIA O PAREJA, CON CONFLICTOS	3 CON FAMILIA, SALE	3 FORMAL SUFICIENTE
4 SOLO, FAMILIA NO CUBRE NECESIDADES	4 CON FAMILIA O VISITAS, NO SALE	4 FORMAL INSUFICIENTE
5 SOLO, DESATENDIDO, SIN FAMILIA	5 CON NADIE, NO SALE	5 INFORMAL
		6 NO RECIBE Y NECESITA

RIESGO SOCIAL : 5 o MENOS = BAJO // 6 - 9 = ACEPTABLE // 10 - 15 = ALTO

GRUPO DE EDAD

1	65 - 74
2	75 - 84
3	85 - 94
4	95 O MAS

VIVE CON

1	FAMILIAR
2	NO FAMILIAR
3	SOLO
4	INSTITUCION