



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE INGENIERÍA

DISEÑO Y DESARROLLO DE UN SISTEMA ELECTRÓNICO DE  
REGISTRO PERSONAL DE SALUD

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A:

DIEGO SALAS ANDRADE



DIRECTOR DE TESIS:  
MTRO. ESTEBAN ARRANGÓIZ ARECHAVALA  
2016

CIUDAD UNIVERSITARIA, CDMX

2016



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A mis padres...

# CONTENIDO

<b>I. INTRODUCCIÓN.....</b>	<b>3</b>
I.I. PLANTEAMIENTO DEL PROBLEMA .....	1
I.II. JUSTIFICACIÓN .....	1
I.III. OBJETIVOS.....	3
I.III.I. Objetivos generales .....	3
I.III.II. Objetivos específicos .....	3
I.IV. TRABAJOS RELACIONADOS O ANTECEDENTES .....	4
<b>II. MARCO TEÓRICO.....</b>	<b>6</b>
II.I. LENGUAJES DE PROGRAMACIÓN .....	6
II.I.I. Definición.....	6
II.I.II. Tipos .....	6
II.II. PARADIGMAS DE PROGRAMACIÓN.....	7
II.III. BASES DE DATOS .....	8
II.III.I. Definición.....	8
II.III.II. Ventajas de las bases de datos .....	8
II.III.III. Sistema manejador de bases de datos .....	9
II.III.IV. Lenguajes de las bases de datos.....	9
II.III.V. Modelo de base de datos .....	9
II.IV. SERVIDOR.....	12
II.IV.I. Definición .....	12
II.IV.II. Tipos .....	13
II.IV.III. Arquitectura cliente-servidor .....	13
II.V. INFORMÁTICA BIOMÉDICA .....	13
II.V.I. Definición .....	14
II.V.II. Fundamentos .....	14
II.VI. SISTEMAS DE INFORMACIÓN SANITARIA .....	15
II.VI.I. Definición .....	15
II.VI.II. Tipos .....	15
II.VI.I. Propósito de uso .....	16
II.VII. REGISTRO PERSONAL DE SALUD.....	17
II.VII.I. Definición .....	17
II.VII.II. Tipos .....	17
II.VII.III. Beneficios .....	18

II.VII.IV. Aspectos a tomar en cuenta .....	18
II.VII.V. Estructura y composición .....	19
II.VIII. ASPECTOS ÉTICOS E INFLUENCIA DEL USO DE SIS SOBRE LA RELACIÓN MÉDICO-PACIENTE ...	20
II.IX. MODELOS DE PROCESO .....	21
<b>III. METODOLOGÍA .....</b>	<b>23</b>
III.I. MODELO DE PROCESO UNIFICADO .....	23
III.I.I. Definición .....	23
III.I.II. Beneficios .....	23
III.I.III. Estructura .....	24
III.II. PROGRAMACIÓN EXTREMA .....	26
III.II.I. Definición .....	26
III.II.II. Estructura .....	26
III.II.III. Beneficios .....	29
III.III. ARQUITECTURA MODELO - VISTA - CONTROLADOR .....	29
III.III.I. Definición .....	29
III.III.II. Estructura .....	30
III.III.III. Beneficios .....	30
<b>IV. DESARROLLO .....</b>	<b>32</b>
IV.I. INCEPCIÓN .....	32
IV.I.I. Alcance .....	32
IV.I.II. Requerimientos .....	34
IV.I.III. Plan de fases e iteraciones posteriores .....	38
IV.I.IV. Riesgos potenciales asociados al proyecto .....	38
IV.I.V. Visión general de la arquitectura de software propuesta .....	39
IV.I.VI. Requerimientos de hardware y software .....	40
IV.I.VII. Entregables .....	40
IV.II. ELABORACIÓN .....	41
IV.II.I. Estructura del sistema .....	41
IV.II.II. Resolución de riesgos altos .....	45
IV.II.III. Herramientas utilizadas .....	45
IV.II.IV. Entregables .....	50
IV.III. CONSTRUCCIÓN .....	50
IV.III.I. Configuración .....	51
IV.III.II. Programación .....	53

IV.III.III. Navegación del sistema.....	54
<b>V. RESULTADOS.....</b>	<b>81</b>
<b>VI. CONCLUSIONES.....</b>	<b>84</b>
<b>VII. BIBLIOGRAFÍA .....</b>	<b>87</b>

# **I. INTRODUCCIÓN**

## **I.I. PLANTEAMIENTO DEL PROBLEMA**

La Facultad de Medicina de la Universidad Nacional Autónoma de México imparte la materia de Informática Biomédica con la finalidad de promover el conocimiento y aplicación de la tecnología en el ámbito médico, promoviendo herramientas para el cuidado y manejo de la salud.

De esta manera, el departamento ha identificado (teniendo experiencia de primera mano) que no existe actualmente en México ningún proyecto formal para el manejo profundo de datos de salud personales, sin tener que contar con un expediente clínico provisto por alguna institución; nuestro país presenta un grave problema de manejo y acceso a datos personales [1], [2] que, en cuestiones de salud, surge en parte por la segmentación del sector de asistencia sanitaria, pues cada institución proveedora de estos servicios posee su propio sistema de registros.

También, estos sistemas de registros médicos son en su mayoría físicos y no tienen posibilidad de ser interoperables, lo que puede volver ineficiente la atención médica, provocando una pérdida de tiempo y un sentimiento en general de descontento en la población (tanto para el profesional de la salud como para el paciente).

Debido a la situación de los registros mencionados, se presentan situaciones como: tener que repetir innecesariamente pruebas clínicas debido a la pérdida de información, acceso complicado o restringido a los datos, historiales incompletos, falta de apuntes cotidianos o la necesidad de gastar la mitad del tiempo de una consulta médica en discutir, lo mejor posible y siempre omitiendo detalles, la condición actual del paciente desde la última visita.

Es así que, no sólo no se cuenta en México con un expediente electrónico sanitario que sea accesible a la población en general, sino que la información se encuentra tan dispersa que es difícil tener un seguimiento del cuidado de la salud.

## **I.II. JUSTIFICACIÓN**

El proyecto que se propone pretende suplir el vacío informático existente en el cuidado y manejo de la salud en México, implementando un sistema de Registro Personal de Salud web en la facultad de medicina.

Se optó por el tema porque debido al gran auge de la tecnología y a la creciente necesidad de extender el cuidado más allá del encuentro clínico, va a ser una necesidad social (para cualquier grupo de la población) tener la costumbre de cargar un mayor peso en el cuidado de su salud; sobre todo para personas con enfermedades crónicas, de edad avanzada o para guardianes de alguna otra persona.

Se reconoce que un fundamento informático sólido utilizado como fuente de datos en los cuales se basen políticas racionales de salud, es primordial para mejorar la calidad de los cuidados de salud, reducir costos y asegurar acceso a estos cuidados [3].

Además, se puede sacar provecho de que las facultades de medicina incorporan en los programas de formación de sus estudiantes, aspectos legales y éticos relacionados con el uso de los computadores en la práctica clínica [4], pudiendo proponer que los alumnos revisen un sistemas que revise, aunque de manera general, todos estos aspectos.

Así, un Registro Personal de Salud (RPS) puede extender el cuidado más allá del encuentro clínico, sin embargo, su implementación debe realizarse en coordinación con el médico del paciente, un nivel de funcionalidad de la que muchos sistemas independientes (no asociados a ninguna institución de salud) carecen [5].

Se tiene el objetivo de ayudar a recopilar datos de salud (proporcionados por el mismo usuario o su tutor) y de esta manera tener los datos accesibles y gestionables, promoviendo una mayor inmersión en el cuidado propio de la salud y enriqueciendo a la vez la relación médico-paciente.

De esta manera, la posibilidad de que la Facultad de Medicina de la UNAM cuente con un sistema web cuyos usuarios puedan registrar, modificar y visualizar sus datos de salud más relevantes (hospitalizaciones, vacunaciones, enfermedades, anotaciones cotidianas, etc.) tomando como base la historia clínica académica, permitiría el intercambio de datos registrados por el usuario con el Expediente Clínico Electrónico Universitario.

El proyecto va encaminado a realizarse por fases, teniendo a lo largo de cada fase cierta realimentación de usuarios de prueba (principalmente alumnos y personal académico) con el fin de orientar el sistema a ser lo más sencillo de utilizar. Sin embargo y a pesar de la sencillez, se busca también que el sistema sea lo más completo y robusto posible para posteriormente darle seguimiento y ajustarlo dependiendo de las necesidades cambiantes de la población, buscando llegar a aquellos grupos que se beneficiarían más con su uso.



El enfoque de este trabajo es un tanto diferente a otros realizados anteriormente en cuanto a los usuarios de prueba pues se combinan dos mercados potenciales para el sistema; lo anterior debido a que su primer implantación será de origen académico (en la Facultad de Medicina) y principalmente teniendo como usuarios a alumnos. Por esta razón, se puede tener por una parte la experiencia de usuarios generales (pacientes) y por el otro lado la experiencia de usuarios orientados al cuidado de la salud (médicos).

Finalmente, el proyecto está vinculado con un cambio en la relación médico-paciente, empoderando al usuario con sus datos de salud y promoviendo un mayor acercamiento al autoconocimiento e interés del cuidado personal. Una cuestión muy importante considerando que actualmente los servicios de salud en México enfrentan todavía una serie de desafíos importantes, pero teniendo la ventaja, a diferencia otros sistemas que existen, de ser respaldado por una institución tan reconocida como la UNAM.

### **I.III. OBJETIVOS**

#### **I.III.I. Objetivos generales**

Diseñar y desarrollar un sistema electrónico de registro y gestión de información de salud personal para la Facultad de Medicina, disponible en línea y de libre acceso, para unificar y simplificar la visualización así como el manejo de datos de salud registrados por el usuario.

Proveer conexión con el Expediente Clínico Electrónico Universitario (ECEU) para compartir información en ambos sentidos, a fin de reforzar y facilitar la relación médico-paciente.

#### **I.III.II. Objetivos específicos**

El sistema permitirá a los usuarios:

- Registrar información sobre su salud como:
  - Alergias y reacciones adversas a medicamentos
  - Enfermedades crónicas y agudas
  - Hospitalizaciones
  - Historia familiar
  - Reportes de escaneos y pruebas de laboratorio
  - Cirugías y otros procedimientos
  - Antecedentes
  - Mediciones de salud y observaciones de la vida diaria
  - Enfermedades con enfoque de género
- Gestionar:
  - Control de medicamentos

- Prescripciones de medicamentos

En esta tesis se pretende también demostrar que se facilita la implementación de un RPS si se cuenta con el apoyo de profesionales en el cuidado de la salud, en este caso, principalmente profesores y, debido a los anteriores, estudiantes de medicina.

#### **I.IV. TRABAJOS RELACIONADOS O ANTECEDENTES**

Son pocos los trabajos documentados a nivel internacional que toman un enfoque práctico en cuanto al Registro Personal de Salud pues en su mayoría, simplemente mencionan este tipo de registro de una manera puramente descriptiva o informativa. Y aún de los trabajos prácticos mencionados, no en todos se trata de una implementación electrónica sino todavía como un registro físico que cada persona mantiene y sin un formato estandarizado. En otros casos, se trata de sistemas locales (limitados a pocas clínicas) y cuyos trabajos escritos se enfocan más en el análisis de percepción de uso, no en la estructura y desarrollo como tal.

Algunas de las muestras más relevantes de sistemas implementados son: HealthVault (Microsoft), Myhealth manager (KaiserPermanente) y GoogleHealth (Google) como ejemplo de un sistema exitoso, uno medianamente exitoso y uno que fracasó, respectivamente.

En cuanto a los trabajos que se encontraron de sistemas funcionales e implementados, se toma por ejemplo, el sistema de Sudáfrica [6], el cual es la prueba de éxito de estos sistemas, sobre todo en un área de necesidades sanitarias como viene a ser la creciente población de México.

Aparte de lo anterior, se han encontrado varios formatos para la implementación de registros personales físicos, sin documentación alguna sobre el uso que ha tenido o la inmersión que ha tenido en el sistema de salud. Estos trabajos han permitido entender la base de un RPS al contrastar los elementos que había en común, de tal manera que se consideraron como los fundamentales de un sistema de registro personal.

Se mencionan a continuación algunos puntos importantes que se concluyeron después de revisar trabajos relacionados y que es conveniente mencionar para tomar en cuenta en este proyecto.

- Una proporción sustancial de pacientes de cuidados primarios utilizará un Registro Personal de Salud en línea que pueda interactuar con el expediente clínico electrónico [5].
- Entre pacientes, un factor clave influenciando el uso del Registro Personal de Salud fue tener una condición comórbida [6].

- Datos que sólo pudieron ser generados por el paciente fueron muy valiosos puesto que permitieron que tanto el paciente como el equipo de salud se prepararan con anticipación para la visita. Además, se proveían datos más completos y acertados, y se enfocaba la discusión en la planificación del cuidado y opciones de tratamiento del paciente [7].
- El contenido, la apariencia, el rendimiento del sitio, la funcionalidad y navegación, son todos aspectos que impactan en el uso y en la activación del paciente [7].
- Los RPS necesitan estar bien alineados con las estrategias existentes de cuidado de la salud, con el fin de coordinarse con otros programas organizacionales en la atención de aquellos individuos en mayor riesgo [7].
- Estudios de Registros Personales de Salud de los Estados Unidos han sido en gran parte descriptivos, investigando la adopción y uso, políticas y gobernanza, y como un medio para mejorar la centralización en los pacientes [8].
- De manera importante, los obstáculos para desarrollar un RPS exitoso no son solamente de naturaleza técnica, sino también tienen dimensiones organizacionales, culturales y legales. Investigadores han usado el término “socio-técnicos” para describir estas complejas cuestiones abordando la naturaleza interconectada de la sociedad y la tecnología del cuidado de la salud [8].
- Análisis demuestran que cada actor participando en esta implementación de RPS (tanto a nivel individual como organizacional) tiene diferentes criterios para definir el éxito de la implementación [8].
- Estudios anteriores sobre el proceso de implementar largos sistemas técnicos en parámetros del cuidado de la salud, han demostrado que la interacción de factores sociales y técnicos da forma de manera crucial a la implementación [8].
- Poco se sabe acerca de si los consumidores del cuidado de salud están interesados en adoptar un RPS [9].
- Registros Personales de Salud conectados a proveedores de Registros de Salud Electrónicos (EHR) son un poderoso vehículo para apoyar la interacción paciente-médico, las necesidades de información del paciente, y el compromiso en el proceso de cuidado [7].

Todas estas anotaciones proponen un panorama de desconocimiento en esta área de la medicina, y una incipiente solución por el lado de la tecnología, siendo que en muchos casos, los sistemas implementados se pusieron a disposición únicamente durante un corto periodo de tiempo y con fines de llevar a cabo un estudio.

## II. MARCO TEÓRICO

### II.I. LENGUAJES DE PROGRAMACIÓN

#### II.I.I. Definición

Cada computadora, de acuerdo a su diseño, puede entender solamente un cierto conjunto de instrucciones elementales denominado **lenguaje máquina o código máquina** [10]. Este lenguaje, es directamente interpretable por el hardware y le indica a la computadora lo que tiene que hacer, no obstante, al utilizar el alfabeto binario (1's y 0's), el código tiende a ser poco comprensible.

Es así que, para facilitar la labor al programador, surgen los lenguajes de programación cuya sintaxis fue progresando para hacerse, si bien más compleja, mucho más comprensible y por lo tanto más fácil de manejar. Entonces, podemos entender un lenguaje de programación como un "*lenguaje artificial que se utiliza para expresar programas de ordenador*" [10] y que puede ser de dos tipos, de acuerdo al siguiente cuadro (**Tabla 2.1**).

**Tabla 2.1**

*Comparación entre lenguajes de programación de alto y bajo nivel*

Lenguajes de programación Comparación entre nivel alto y bajo	NIVEL	
	Alto	Bajo
Dependiente de la arquitectura	No	Si
Rapidez de procesamiento	Baja	Alta
Legibilidad	Alta	Baja

Nota: Según algunos autores, también existen otros lenguajes denominados de nivel medio, los cuales no se incluyen en el cuadro comparativo.

#### II.I.II. Tipos

Dependiendo de la forma en la que son procesados, hay dos tipos de lenguajes:

**Lenguajes interpretados:** cuando un programa es escrito en un lenguaje interpretado, la computadora toma el código fuente y posteriormente lo va traduciendo y ejecutando instrucción por instrucción [10].

**Lenguajes compilados:** cuando un programa es escrito en un lenguaje compilado, la computadora toma el código fuente y produce un programa equivalente en código máquina o byte code llamado programa objeto o código objeto [10] que será posteriormente ejecutado.

## II.II. PARADIGMAS DE PROGRAMACIÓN

Se puede definir un paradigma de programación como una *“colección de patrones conceptuales que moldean la forma de razonar sobre un problema, de formular soluciones y de estructurar programas”* [10], es decir, el modelo o enfoque a seguir para conceptualizar un problema, abstraerlo en tareas y resolverlo mediante un programa.

Entre los paradigmas más utilizados actualmente [10] se encuentran:

- Programación estructurada o imperativa
- Programación funcional
- Programación lógica
- Programación orientada a objetos

### **Programación orientada a objetos**

El concepto de la programación orientada a objetos (POO) es abstraer la realidad en plantillas denominadas clases, cada una representando un concepto con características y funciones genéricas y a partir de las cuales se crean entes u objetos con características (atributos) y funciones (métodos) particulares. En sí, este tipo de programación considera cada programa como una colección de objetos que interactúan entre sí y que están continuamente comunicándose por medio de mensajes.

### **Ventajas de la POO**

La POO proporciona las siguientes ventajas sobre otros paradigmas de programación, de acuerdo a [11] algunas son:

**Protección de la información:** el encapsulamiento de los datos, junto con la definición de niveles de protección, proporcionan un alto nivel de seguridad de la información al restringir el acceso solamente a los medios a los que se les dé permiso de manera explícita.

**Trabajo colaborativo:** cada módulo puede desarrollarse mediante el uso de clases diferentes que serán posteriormente enlazadas, gracias a lo cual el trabajo puede segmentarse y ser repartido entre varios colaboradores sin que esto implique hacer la programación más compleja.

**Aumento de claridad:** los atributos (datos que componen a los objetos) y métodos (procedimientos que manipulan a los objetos) están agrupados en clases de acuerdo a las estructuras de información que el programa necesita, siendo estas clases definidas de la manera más explícita e informativa posible.

**Facilidad de extensibilidad:** la ampliación de un programa es más simple ya que, para futuros cambios, no es necesario en principio modificar todos los archivos del sistema sino solamente las clases que se desee actualizar o, en caso de ser necesario, agregar clases nuevas.

**Facilidad de modificaciones y correcciones:** para actualizar algún método o atributo y verlo reflejado en todo el programa, basta con modificar la clase a la que éste pertenece y todas las instancias que existan contarán con los cambios realizados.

**Obtener componentes reutilizables:** al utilizar abstracciones para formar las plantillas, un programa puede reusar clases definidas anteriormente para otros programas, pudiendo incluso adaptar estas estructuras de datos a las nuevas necesidades. Esto implica que, en el mejor de los casos, *“el desarrollo de un programa puede llegar a ser una simple combinación de objetos ya definidos y donde éstos están relacionados de una manera particular”* [11].

## II.III. BASES DE DATOS

### II.III.I. Definición

Una base de datos es, de manera muy general, “un único repositorio que contiene varios elementos de datos individuales” los cuales “no existen en aislamiento en la base de datos” sino que “están relacionados unos con otros” [12], en otras palabras, una base de datos contiene un conjunto de elementos de datos que están relacionados.

Sin embargo, para que este conjunto de datos sea considerado como una base de datos, debe ser almacenado sin redundancias innecesarias en una estructura informática a la cual debe poderse acceder por distintos usuarios y aplicaciones, y de manera simultánea. Además, es importante mencionar que estos datos deben estar estructurados y conservados de forma independiente respecto a las aplicaciones que los utilizan [13].

### II.III.II. Ventajas de las bases de datos

Las bases de datos presentan ciertas ventajas con respecto a otro tipo de formas de almacenamiento de datos, de acuerdo a [14] véanse algunas:

- **Independencia de datos respecto a programas y procesos:** de esta manera, el código de las aplicaciones no cambia aunque se modifiquen los datos.
- **Menor redundancia:** teniendo un buen diseño de la base de datos, se tiene la menor repetición de datos posible.
- **Integridad de los datos:** los datos que son introducidos en una base de datos se mantienen precisos, válidos y con coherencia debido a que, generalmente, los repositorios

están sujetos a restricciones y cuentan con verificaciones de integridad. De esta manera es difícil que se pierdan datos dentro de la base o que exista alguna incoherencia.

- **Mayor seguridad en los datos:** al limitar el acceso a ciertos usuarios, se evita que cualquier persona pueda usar o modificar la información almacenada.
- **Datos más documentados:** puesto que existen metadatos que describen las características del contenido de la base de datos.
- **Acceso a los datos más eficiente:** como los datos tienen cierta organización, se tiene un rendimiento eficiente.
- **Menor espacio de almacenamiento:** al estar los datos estructurados, se tiene un mejor uso de espacio en memoria.

### **II.III.III. Sistema manejador de bases de datos**

Se denomina como sistema gestor de bases de datos (SGBD) o sistema manejador de bases de datos, a un *“conjunto de programas para acceder [a una] colección de datos interrelacionados denominada base de datos”* [15]. Su principal objetivo es servir de interfaz para almacenar y recuperar la información de una base de datos de manera práctica y eficiente.

Para manejar correctamente los datos de la base, los SGBD deben definir las estructuras para almacenar la información así como los mecanismos para manipularla. Además, se debe proveer fiabilidad de la información almacenada, sin importar que se caiga el sistema o que haya intentos no autorizados de acceso. También, si los datos van a ser compartidos entre diversos usuarios, el sistema debe evitar anomalías en los resultados [15].

### **II.III.IV. Lenguajes de las bases de datos**

Para poder modificar la base de datos, el SGBD proporcionan dos tipos de lenguajes; un lenguaje de definición de datos para especificar el esquema (estructura) de la base de datos y un lenguaje de manipulación de datos para expresar las consultas (peticiones) a la base de datos y las modificaciones [15].

### **II.III.V. Modelo de base de datos**

El modelo de una base de datos hace referencia la estructura lógica del contenedor de datos de la base (donde se almacenan los datos), es decir, la manera como son representados elementos de la realidad en una estructura de datos. También describe los métodos usados para almacenar, organizar, manipular y recuperar los datos.

En este sentido, un modelo de base de datos no es algo concreto, sino un más bien una abstracción para crear un sistema de base de datos que se ajuste a cierto problema lo mejor posible y de esa manera tener una aplicación eficiente.

Algunos modelos con frecuencia utilizados en las bases de datos son:

- Modelo jerárquico
- Modelo de red
- Modelo relacional
- Modelo orientado a objetos
- Modelo documental

De los cuales, el más utilizado actualmente es el modelo relacional y el cual se explicará a continuación.

### **Bases de datos relacionales**

Debido a ciertas deficiencias como la poca flexibilidad para los cambios y un acceso relativamente lento a la información, propias de los modelos jerárquicos y en red, surgen con gran aceptación las bases de datos relacionales [16]. Entre las principales razones de éxito de este tipo de bases están: la creación más sencilla de sistemas gestores de bases de datos que soporten el modelo relacional y los sólidos fundamentos teóricos sobre los que se basa.

En las bases de datos que siguen el modelo relacional, los datos se representan como un conjunto de tablas interconectadas o relacionadas entre sí (de ahí el nombre “relacional”), en las cuales cada registro de la base contiene información concerniente solamente a un tema.

Además, cada tabla está conformada por filas, también llamadas tuplas o registros, organizadas verticalmente y por columnas organizadas horizontalmente, formando así una estructura rectangular.

De esta manera, bajo cada columna o campo, se enlistan todos los datos ingresados de cierta propiedad (atributo) de la tabla, mientras que cada fila representa todas las entradas realizadas para cierto registro. Se puede decir que la suma de todas las entradas de los campos forma un registro o fila, debido a lo cual generalmente las tablas se leen horizontalmente.

En la **Tabla 2.2**, se muestra un esquema que representa de manera general una tabla bajo el modelo relacional.



**Tabla 2.2***Esquema general de una tabla relacional*

	NOMBRE DE LA ENTIDAD					
	columna 1	columna 2	columna 3	columna 4	...	columna n
	Atributo 1	Atributo 2	Atributo 3	Atributo 4	...	Atributo n
fila 1	valor 1, 1	valor 1, 2	valor 1, 3	valor 1, 4	...	valor 1, n
fila 2	valor 2, 1	valor 2, 2	valor 2, 3	valor 2, 4	...	valor 2, n
fila 3	valor 3, 1	valor 3, 2	valor 3, 3	valor 3, 4	...	valor 3, n
fila 4	valor 4, 1	valor 4, 2	valor 4, 3	valor 4, 4	...	valor 4, n
...	...	...	..	...	...	...
fila n	valor n, 1	valor n, 2	valor n, 3	valor n, 4	...	valor n, n

Nota: "n" representa el número de la última fila o columna existente.

Así, si pensamos que hay una tabla llamada persona, cada columna almacena una propiedad de una persona como el nombre, apellidos, edad, etc., y cada fila representa un registro en el que todos los campos se relacionan para formar la información de una persona. Esto puede verse ejemplificado en la siguiente **Tabla 2.3**.

**Tabla 2.3***Ejemplo de tabla relacional*

	PERSONA					
	campo 1	campo 2	campo 3	campo 4	campo 5	campo 6
	Nombre	Apellido P.	Apellido M.	Edad	Sexo	Con hijos
registro 1	Esteban	Sánchez	Poncio	25	M	SI
registro 2	Dalia	Pérez	Aguilar	46	F	SI
registro 3	Juan	Romero	Cruz	63	M	NO
registro 4	Rosario	Farah	Santana	37	F	SI
registro 5	Yahir	Godínez	Zaldivar	24	M	NO
registro 6	Marlette	Guzmán	Sánchez	17	F	NO

En el cual se podría leer el registro #3 como una persona llamada "Juan Romero Cruz", de 63 años de edad, hombre y sin hijos.

El manejo de estas tablas puede realizarse mediante operaciones clásicas de la teoría de conjuntos como la unión, intersección, diferencia y producto cartesiano, o también mediante operaciones específicas del modelo relacional como la selección, proyección, reunión y división [13].

### **Ventajas de las bases de datos relacionales**

Las bases de datos relacionales proveen un número de ventajas sobre previos modelos, según [17] estas son:

**Integridad multinivel incorporada:** la integridad de los datos es realizada por el SGBD de manera interna y a través de varios escalones. Se mencionan a continuación los niveles de integridad.

1. A nivel de campo: para asegurarse de la exactitud de los datos, el usuario define las especificaciones de la estructura del campo, buscando que se cumplan las características de un “campo ideal”;
2. a nivel de tabla: para asegurarse de la integridad de los registros en general, asegurándose que no hay registros duplicados, no hay llaves primarias nulas, los valores de las llaves primarias son únicos y que las llaves primarias identifican exclusivamente a un registro en las tablas;
3. a nivel relacional: para asegurar que la relación entre tablas es válida, el usuario define llaves primarias y foráneas para hacer la relación (uno-a-uno o uno-a-muchos) así como reglas de supresión de los datos, entre otras medidas;
4. y a nivel de negocios: para asegurarse que los datos son precisos en términos del negocio como tal y dependiente de la manera en que la organización que implementa la base de datos utiliza su información.

**Independencia lógica y física de las aplicaciones con respecto a la base de datos:** ni los cambios lógicos (por parte del usuario) ni físicos (por parte del proveedor del software) que se hagan al diseño de la base, afectarán de manera desfavorable a las aplicaciones integradas.

**Consistencia y precisión de los datos:** con un correcto diseño de la base, los datos son consistentes y precisos debido a los varios niveles de integridad que se pueden imponer dentro de la base de datos y al procesamiento de transacciones, mediante el cual se tiene a la base en un estado consistente en cualquier momento, y en el cual debe permanecer inclusive en caso de haber un fallo en el sistema.

**Fácil recuperación de datos:** los datos pueden ser recuperados, ya sea de una tabla particular o de cualquier número de tablas relacionadas dentro de la base, y de la manera más adecuada gracias al lenguaje de consultas y al SGBD quien optimiza el proceso de búsqueda.

## **II.IV. SERVIDOR**

### **II.IV.I. Definición**

La creciente necesidad de almacenar y centralizar grandes volúmenes de datos, fue la razón de que se crearan los servidores [18], ya que éstos se definen como máquinas especializadas en brindar servicios a otras máquinas llamadas clientes, por lo general se trata de computadoras.

En este sentido, un servidor puede verse de dos maneras; como software: las aplicaciones necesarias para atender las peticiones de otra máquina y devolver la respuesta; o como hardware: los circuitos físicos diseñados para hospedar a las aplicaciones mencionadas anteriormente.

En conjunto, ambas partes proporcionan las herramientas necesarias para manejar la gran demanda que pueda haber dentro de la red; tanto el hardware como el software, están diseñados para ofrecer rendimiento, seguridad, redundancia, disponibilidad, además para poder ser escalables y utilizados por varios usuarios a la vez.

#### **II.IV.II. Tipos**

De acuerdo al tipo de funcionalidad que se provea, hay muchos tipos de servidores, entre los más usados se encuentran:

- Servidor web
- Servidor de correo
- Servidor de bases de datos
- Servidor proxy
- Servidor de respaldo
- Servidor de seguridad

#### **II.IV.III. Arquitectura cliente-servidor**

Como se mencionó anteriormente, un servidor está diseñado para servir una función específica para otras computadoras denominadas clientes. En aplicaciones web, el servidor y el navegador colaboran en lo que se denomina un sistema cliente-servidor en el cual, el navegador web actúa como el cliente, el cual obtiene información de la red por medio del servidor web.

Siguiendo el esquema, es importante notar que el navegador web es un programa que corre en el sistema de cómputo del cliente, y que el servidor es un programa que corre en el sistema de cómputo del servidor. Ambos dispositivos deben tener acceso al servicio de Internet [19].

### **II.V. INFORMÁTICA BIOMÉDICA**

En la actualidad es una necesidad manejar grandes cantidades de información para realizar acciones en el cuidado de la salud, ya sea que se hable de un profesional del cuidado de la salud o del paciente en sí. Debido a este hecho surge la Informática Biomédica, con la finalidad de mejorar situaciones como: tomar decisiones sobre procedimientos, conocer la situación de una institución de salud, manejar datos epidemiológicos, etc [20].

De esta manera, es importante enfatizar que la Informática Biomédica es la disciplina bajo la cual se encuadran aplicaciones como la realizada para el proyecto; debido a que este campo de estudio

hace uso del manejo de datos por parte de la informática, y se combina con el cuidado de la salud, por parte de la medicina.

### **II.V.I. Definición**

La Asociación Médica de Informática de EEUU (AMIA) define a la Informática Biomédica (IB) como el “campo interdisciplinario que estudia y persigue el uso efectivo de datos biomédicos, información, y conocimiento para investigación científica, resolución de problemas y toma de decisiones, motivado por esfuerzos para mejorar la salud humana”.

Es así que las necesidades del cuidado de la salud hacen que quienes se encuentran comprometidos con estas responsabilidades deban conocer, como mínimo, las nociones de lo que es un sistema, y tener una idea básica de lo que la informática puede hacer en estos ámbitos [20].

Sin embargo, la informática biomédica se sustenta no tanto en el conocimiento de la ingeniería de la computación como tal (en cómo funcionan las herramientas informáticas), sino más bien se enfoca en el valor de la información y en la capacidad para utilizarla [20]. Esta disciplina simplemente participa en la adquisición, almacenamiento y utilización de los datos, como en la toma de decisiones bajo un razonamiento probabilístico.

### **II.V.II. Fundamentos**

De acuerdo a [20], la Informática Biomédica se basa en cuatro pilares que buscan el desarrollo de un nuevo paradigma para el manejo de la información, en lo relativo al campo de la salud. Estos pilares son:

- Producir estructuras para representar datos y conocimiento.
- Desarrollar métodos para una correcta y ordenada adquisición y representación de los datos.
- Manejar el cambio entre los procesos y las personas involucradas para optimizar el uso de la información.
- Integrar la información de diferentes fuentes.

La finalidad de la consecución de estos datos será el tener una adecuada retroalimentación, que permita tomar decisiones oportunas y previsiones en futuras acciones que tiendan a brindar una solución ante una situación determinada.

Es así que la informática en salud tiene como meta la expansión y comunicación de la información, pero ésta es sólo una parte y no el equivalente del complejo proceso de la generación de conocimiento. La información por sí misma no asegura decisiones y no es el fin último de esta herramienta; sino su puesta en marcha para la consecución de una meta más grande, como es la visión y la misión institucional [20].

Finalmente, cabe mencionar que dentro de la informática médica se encuentran los sistemas de información sanitaria, a los cuales se les dedicará el capítulo siguiente debido a su importancia.

## **II.VI. SISTEMAS DE INFORMACIÓN SANITARIA**

### **II.VI.I. Definición**

Dentro de las Tecnologías de Información para la Salud se encuentran los Sistemas de Información Sanitaria (SIS), herramientas que, como cualquier otro sistema de información, permiten el manejo, organización y administración de datos, en este caso datos relacionados a la salud.

Los sistemas mencionados tienen el fin de facilitar el manejo de la situación sanitaria, ya que diariamente se genera una gran cantidad de datos provenientes de muy diversas fuentes, los cuales deben ser seleccionados y manejados de forma ágil y segura [20], para obtener valiosa información de la salud.

En este sentido, según [21], hay dos características principales que debe tener cualquiera de estos sistemas:

- a) convertir los datos en información inteligible, entendiendo que “información = datos x procesamiento” [20] y,
- b) posibilitar la utilización de esa información de manera práctica, como la generación de estadísticas o enfoque en ciertas áreas del cuidado de la salud.

### **II.VI.II. Tipos**

De acuerdo a su manejo, el Manual del expediente clínico electrónico de la Secretaría de Salud [22] agrupa los Sistemas de Información Sanitaria como sigue:

#### **Registro Médico Electrónico (en inglés: Electronic Medical Record):**

Registro clínico sobre un paciente internado o ambulatorio, creado y gestionado independientemente por cada institución médica con sus profesionales de la salud autorizados, con el propósito de mantener los registros de esa organización y fungiendo en ocasiones como reemplazo de la historia clínica física, para dar diagnóstico y tratamiento. También se conoce como Registro Clínico Electrónico.

#### **Registro Electrónico de Salud (en inglés: Electronic Health Record):**

Registro clínico, de carácter legal, que concuerda con estándares de interoperabilidad reconocidos nacionalmente [22], sobre un paciente internado o ambulatorio, manejado generalmente por un cuerpo gubernamental por profesionales de la salud autorizados a través de varios proveedores de

servicios de salud, con la finalidad de mantener los registros de esa organización y, generalmente, es más amplio que el EMR ya que puede incluir otros aspectos además del físico.

**Registro Personal de Salud (en inglés: Personal Health Record):**

Registro de salud sin carácter legal de un individuo, iniciado y mantenido por un individuo (generalmente el paciente), pero con múltiples fuentes de información, para propósitos de cuidado continuo.

**Sistema de Información Hospitalaria (Hospital Information System):**

Sistema integral de información diseñado para administrar los aspectos financieros, clínicos y operativos de una organización de salud, manejado por. Puede incluir o estar conectado con un Expediente Clínico Electrónico.

Cabe mencionar un par de cosas: la primera es que a la fecha, todavía no están universalmente definidas las categorías en que se dividen los SIS, ya que pueden cambiar dependiendo del autor, llegando incluso a mencionar el EMR y el EHR como uno sólo. El segundo punto es que realmente los principales tipos de sistemas son los primeros tres, pues están más enfocados a la parte clínica, a diferencia del último, el cual es predominantemente de administración.

Finalmente, tanto el EMR como el EHR, ya han sido implementados mundialmente en varias instituciones de salud como sistemas electrónicos, no así el Registro Personal de Salud, el cual, debido a la falta de cultura del manejo personal de la salud, no ha sido implementada en tantos lugares, y en aquellos casos, generalmente sigue siendo un registro físico.

**II.VI.I. Propósito de uso**

Los Sistemas de Información Sanitaria se desarrollan como herramientas educativas para la intervención y práctica, principalmente médica, basada en evidencia, mejorando la calidad de los datos y las intervenciones clínicas en la atención primaria. *“La meta es una condición previa y el sistema de información es el instrumento de medición, no el fin”* [20].

Siendo así, hay que tener una clara idea de los datos que quieren obtenerse, para tenerlos en cuenta al definir como ejes principales las necesidades de las personas, como un grupo, y un enfoque biomédico, necesario para la selección de las variables epidemiológicas. Uniendo ambos ejes se puede planificar de manera correcta actividades por profesionales de la salud entorno a la información obtenida y, proveer un acceso dinámico a los datos [20].

## **II.VII. REGISTRO PERSONAL DE SALUD**

### **II.VII.I. Definición**

El Registro Personal de Salud (RPS, aunque más conocido por sus siglas en inglés PHR) es un tipo de Sistema de Información Sanitaria que tiene como principal función recopilar información de salud proveniente de diversas fuentes y sobre una persona en particular. Generalmente este registro es creado, compartido y gestionado por la misma persona [23] y, al no ser estrictamente un documento legal, puede o no cumplir con estándares de interoperabilidad nacionales.

### **II.VII.II. Tipos**

Se pueden mencionar dos tipos de RPS en función de su relación con alguna institución de salud, sin embargo, y al igual que sucede con los Sistemas de Información en Salud, las características bajo las cuales debe identificarse cada sistema varían en ciertos aspectos dependiendo del autor. En este caso tomaremos, por su especificidad, la clasificación de [8] que agrupa los RPS en tres clases:

#### **RPS ligados o conectados a una institución de salud**

Sistemas que se agrupan bajo esta categoría tienen relación con alguna organización del cuidado de la salud, generalmente a través de un EHR o sistema similar. Sólo se permite a los pacientes ver su información de salud, almacenada en el registro de su proveedor de cuidados médicos, pero los pacientes no tienen el poder de cambiar el contenido o agregar nueva información.

#### **RPS desligados, autónomos o independientes**

En este tipo de sistema, referido en inglés como “standalone”, el paciente o tutor del paciente son las únicas personas que registran los datos, aunque la fuente es diversa e incluso puede provenir de algún experto en el cuidado de la salud. Como no se utiliza el sistema de algún proveedor, los usuarios utilizan software comercialmente disponible o aplicaciones web, y ellos mismos quienes deciden quién puede ver la información recopilada.

#### **RPS interoperables**

Se menciona en la literatura, una especie de amalgama entre estos dos extremos: los denominados sistemas “interoperables”, que incorporan características de los sistemas ligados a una institución así como de los sistemas autónomos. Los sistemas interoperables combinan la habilidad de los registros con ataduras de conectarse directamente al EMR de un hospital, con la interfaz de usuario de un sistema autónomo.

Al haber una relación de intercambio de información entre ambos sistemas, puede que ciertas partes del registro o todo, pueda ser visualizado por la institución de salud, sin embargo, también permite

al paciente crear, almacenar y editar su información de salud así como también compartirla con otros proveedores. Se suele agrupar este tipo de registros dentro de los RPS conectados a una institución de salud.

Independientemente del tipo de Registro Personal de Salud que se use, todos impulsan a los pacientes a tomar un rol más activo en el cuidado de su salud al darles más responsabilidad de mantener un estilo de vida saludable y del manejo de enfermedades crónicas.

### **II.VII.III. Beneficios**

Szolovits, Doyle, Long, & Kohane visualizan un Sistema de Información Sanitaria como un registro personal, como un “Ángel guardián” que recoge información médica y la unifica, pero además, la revisa e interpreta para finalmente explicarle al sujeto hechos y planes médicamente relevantes. Con estas acciones, se puede obtener una mejor calidad en la toma de decisiones médicas y minimizar los errores médicos [6], [24].

Debido a que ninguna práctica médica puede compartir registros médicos entre ellas [6], los RPS proveen a los consumidores del cuidado de la salud de una manera completamente diferente, centrada en el paciente, de manejar registros de salud e información médica [9], también representan una manera efectiva en costos de mejorar la calidad del cuidado de pacientes.

Finalmente, hay que resaltar el papel que puede jugar la tecnología en el manejo de este tipo de registros al automatizar los sistemas y permitir su uso en línea. Debido a esto, un RPS bien diseñado es mucho más fácil de entender y de usar que un registro físico, provee acceso a información de asistencia médica en cualquier parte del mundo, quizá lo más importante de todo, facilita la comunicación continua entre pacientes y médicos [6].

### **II.VII.IV. Aspectos a tomar en cuenta**

Los beneficios potenciales de los RPS son numerosos, sin embargo, tienen una complicación debido a que requieren de un compromiso a término largo en cuanto a registros y manejo de información por los consumidores, quienes buscan que los beneficios eventualmente sobrepasen los esfuerzos de la configuración inicial y los costos de aprendizaje [9].

Aunado a esto, la privacidad y seguridad son cuestiones primarias [9] que se tienen que atender y tomar en cuenta desde el momento de su diseño, ya que información tan delicada como la manejada por estos sistemas puede utilizarse para perjudicar al dueño del registro. La privacidad en los registros de salud es crítica y fallar en manejar este asunto puede afectar el éxito de un sistema RPS, mecanismos como la separación de la información médica de la información personal y un formulario de acceso son empleados para asegurar la privacidad de los pacientes [6].



En un principio, los registros personales de salud fueron meramente un suplente de la vida médica en casa [5], siendo la función más sencilla de un RPS el guardar datos similares, a menudo ingresados por el paciente (primer nivel de atención). Sin embargo, el paciente moderno tiene más posibilidades de involucrarse en su salud que simplemente guardar y mantener registros, ya que con estas herramientas, se puede potenciar las decisiones médicas propias de los pacientes [23].

#### **II.VII.V. Estructura y composición**

Kharrazi, Chisholm, VanNasdale, & Thompson [6], mencionan que los elementos de información esenciales para un RPS completo son:

**Alergias:** La falta de información puede afectar a los pacientes ya que los doctores pueden prescribirles medicamentos a los cuales el paciente es alérgico. Por ejemplo, si un paciente desarrolla una reacción alérgica a un medicamento prescrito, de no contar en futuras atenciones con tal información, otro doctor podría causar más daño al paciente recetándole la misma droga.

**Inmunizaciones:** La vacunación reiterativa y redundante, generalmente a causa de no llevar un registro, vuelve a las vacunas inefectivas [6], haciendo esta herramienta inútil y costosa tanto para la medicina preventiva como para la medicina reactiva.

**Cirugías:** Por ejemplo, se pueden evitar diagnósticos quirúrgicos a pacientes a quienes ya se les hayan realizado estos procedimientos y que no se cuente con registro de ello lo cual, a pesar de los tiempos actuales y sobre todo en zonas provinciales o rurales, puede llegar a suceder. De esta manera, si a un paciente se le extrajo el apéndice inflamado y posteriormente presenta síntomas similares, no se debe considerar apendicitis como posible diagnóstico.

**Condiciones crónicas:** En pacientes con enfermedades crónicas, debido a la larga duración que representan sus atenciones, es necesario recopilar información con el fin de asegurar la continuidad del cuidado y monitorear cualquier cambio ya sea favorable o desfavorable que se pueda producir en el individuo.

**Medicaciones:** Con el fin de dar mejor diagnóstico y seguimiento, los doctores necesitan saber qué tipo de medicación el paciente ha estado tomando, evitando contraindicaciones, reacciones o cualquier otro problema que pueda surgir de la interacción entre fármacos.

**Historia familiar:** Los antecedentes familiares pueden dar indicios al médico sobre cuáles enfermedades, sobre todo de carácter genético o hereditario, puede llegar a padecer el paciente, así como ayudar al doctor en diagnosticar una dolencia.

**Información de imagen médica:** El registro de resultados de estudios realizados al paciente, como radiografías o termografías, pueden evitarle al usuario el tener que repetir las pruebas, lo que implicaría un gasto y una molestia innecesarios. Inclusive, el mismo departamento que realice las pruebas podría enviar al doctor las imágenes médicas, en formato electrónico, quien posteriormente podría compartirlas al paciente de una manera más cómoda.

Además, Iakovidis sugiere que el valor del registro médico de los pacientes recae en algunas de propiedades [23]:

**Accesibilidad y disponibilidad:** El registro del paciente debe ser accesible y estar disponible de manera continua.

**Usabilidad y flexibilidad:** El sistema debe poder soportar múltiples visualizaciones de usuarios a la vez, así como las interacciones que estos hagan con su registro, permitiendo así, tanto la salida como la entrada de datos de manera simultánea.

**Alcance integral (Integración):** Se debe lograr una gran cobertura de las necesidades o requerimientos de los usuarios mediante el uso en conjunto de varias TIC disponibles, como otros sistemas clínicos o administrativos, de manera que todo funcione como un solo sistema para centralizar la información.

**Rendimiento:** El sistema debe ser capaz de procesar de manera adecuada y rápida las tareas que los usuarios realizan, es decir, una pronta y eficaz respuesta a la interacción con el usuario.

**Confidencialidad y auditabilidad:** Se debe determinar que la aplicación del sistema está conforme a los estándares y que los datos contenidos están conforme a lo esperado, y utilizados solamente por las entidades autorizadas para su uso.

**Fiabilidad:** Asegura la integridad de los datos y la permanencia de la información original en un formato establecido y por un tiempo dado.

## **II.VIII. ASPECTOS ÉTICOS E INFLUENCIA DEL USO DE SIS SOBRE LA RELACIÓN MÉDICO-PACIENTE**

Emanuel y Emanuel [9] identifican dos tipos principales de relación entre el paciente y el médico, de acuerdo al grado de implicación de cada una de las partes en el manejo de la salud del paciente.

Se menciona por un lado una relación de tipo *informativa* o *autónoma*, en la cual el paciente tiene completo control sobre las acciones a tomar con el fin de procurarse salud y el médico solamente provee experiencia técnica de acuerdo a lo planeado por el paciente.

En este caso, el paciente se considera un “usuario”, un cliente más que está comprando el servicio de la medicina, así que él toma las decisiones, lo cual puede ser contraproducente para el mismo paciente, porque en casos de sufrimiento el paciente es vulnerable a perder su autonomía [4].

Por el otro lado, se menciona la relación de tipo *paternalista*, en la cual el médico toma una posición dominante y pasa a desempeñar un papel más como de guardián que como guía, puesto que es quien le dice al paciente o su tutor el tratamiento a tomar, a pesar de que la aprobación al final venga del mismo paciente.

Sin embargo, los autores también proponen que la relación ideal entre el paciente y el médico, es un punto intermedio entre ambos extremos; llaman a este tipo de correspondencia como modelo *deliberativo* o *responsable*, en el cual médicos y pacientes colaboran para establecer medidas y procedimientos apropiados. Es el más adecuado, puesto que ambas partes asumen la responsabilidad y poder de decisión, conservando tanto el médico como el paciente su autonomía y libertad de acción [4].

## II.IX. MODELOS DE PROCESO

Brevemente, y antes de pasar a la metodología utilizada, se expondrán las categorías de Modelos de Proceso para desarrollo de software existentes actualmente, ya que estos han evolucionado junto con los avances en el poder de procesamiento en computadoras [25].

En general, podemos estructurar una metodología como un **proceso**, que está formado por **fases**, cada una con varias **actividades**, y cada actividad es un conjunto de **tareas**. De esta manera, todos los Modelos de Proceso tendrán fases; sin embargo hay importantes diferencias entre modelos en términos de qué actividades constituyen sus fases y la secuencia en que se presentan.

Una clasificación de los modelos de proceso, de acuerdo a [25], y en función de las fases es la siguiente:

- **Modelos de proceso lineales:** Sus fases se suceden secuencialmente, una tras otra, sin poder saltar o realizar otra fase hasta haber liberado el entregable del periodo actual. Tampoco es posible regresar a una fase anterior una vez que se terminada. Este tipo de modelos es ideal para proyectos que con requerimientos muy definidos y detallados, a los cuales no se les hará modificaciones mayores.

- **Modelos de proceso iterativos:** El producto se construye mediante la repetición de fases en ciclos, siendo cada iteración un refinamiento del producto [25]. De esta manera y, a diferencia del tipo de modelo anterior, el término de todas las fases no necesariamente implica la terminación del proyecto.
- **Modelos de proceso paralelos:** actividades que ocurren concurrentemente, usan un estilo similar de iteración que los iterativos – el producto se construye en iteraciones – sin embargo, los procesos paralelos permiten más concurrencia en actividades y tareas [25].

Este enfoque no implica que un modelo sea mejor que otro, pues es tentador adoptar las últimas innovaciones en los modelos de proceso para el desarrollo de software, sin embargo, vale la pena ganar una perspectiva de las relativas fortalezas y debilidades de cada modelo. Así, un proyecto puede encontrar que algunos modelos son más complejos o exhaustivos de lo que se necesita [25].

### **III. METODOLOGÍA**

Para el desarrollo del sistema, se utilizaron tres metodologías complementarias: para planeación y administración de tiempos, el **Modelo de Proceso Unificado**; para el desarrollo de código la **Programación Extrema** y finalmente; para la estructuración del proyecto, la **Arquitectura MVC** (Modelo, vista y controlador). A continuación, se profundiza en cada modelo.

#### **III.I. MODELO DE PROCESO UNIFICADO**

##### **III.I.I. Definición**

El modelo de proceso unificado es una metodología de desarrollo de software que puede agruparse dentro de los denominados modelos iterativos, ya que su estructura básica tiene fases secuenciales dentro de un ciclo repetitivo. Es así que, dentro de la mayoría de las fases de este modelo, el trabajo ocurre en pequeñas iteraciones hasta que las fases estén completas, lo que usualmente sucede cuando un hito o milestone (un punto específico e identificable en el proyecto) es alcanzado [25].

La estructura general del proceso unificado es iterativa, sin embargo, el modelo permite que tareas de una fase ocurran también en otra. Así, una tarea de requerimientos o actividad pueden suceder en más una sola fase específica; puede suceder a lo largo de las demás fases. Esto último le otorga a la metodología el carácter de proceso paralelo, además de iterativo.

En este sentido, actividades como requerimientos, diseño e implementación pueden suceder al mismo tiempo, y todas ellas conducen al producto a una fase “de entrega”, en la cual se libera un producto. Sin embargo, una vez que el producto se ha liberado, el proceso puede comenzar otro ciclo para refinar y mejorar el producto. Las iteraciones también pasan entre las fases.

##### **III.I.II. Beneficios**

El Modelo de Proceso Unificado, al igual que las demás metodologías, se basa en dividir el trabajo lo mejor posible, de manera que se tenga una adecuada planeación dependiendo de los tiempos en los cuales se divida el proyecto; esto es, asignar cada parte de desarrollo a un determinado lapso, en el cual se debe terminar el trabajo para marcarlo como terminado. Todo lo anterior con el fin de tener una claridad en cuanto al trabajo que se va a realizar.

Sin embargo, y debido a ser una metodología ágil, el proceso unificado es muy adecuado para grandes proyectos en los que las iteraciones permiten al producto crecer naturalmente más allá de la concepción original del producto, ya que permite a los desarrolladores el mantenerse enfocados

en los requisitos principales del proyecto y en hacer correcciones lo más pronto posible para cumplir con lo que el cliente pidió.

### **III.I.III. Estructura**

Para profundizar en la estructura de esta metodología, hay que mencionar que el Modelo de Proceso Unificado tiene 4 fases, a través de las cuales pasa el trabajo, cada fase con distintos objetivos o miras, las cuales se describirán a continuación.

#### **Fase de inyección**

Esta primera fase está destinada a ser de corta duración; el tiempo suficiente para establecer un caso de negocios fuerte para continuar con las siguientes fases y hacer el producto, para lo cual generalmente se delimitan las principales interacciones del usuario con el producto.

También durante esta etapa son definidos los alcances y riesgos potenciales del proyecto, por ejemplo las secciones que los usuarios podrían registrar, o qué pasaría si los usuarios no quisieran usar el sistema, respectivamente.

Esta etapa es la única de las fases del modelo de proceso unificado que no sucede en iteraciones ya que únicamente se da al inicio del proyecto, de manera que si la fase de inyección tarda demasiado, puede ser que se esté desperdiciado tiempo analizando los requerimientos.

El cumplimiento de la fase de inyección está marcado por un ciclo de vida de hitos y el producto de completar este periodo del proyecto es una descripción razonable de la viabilidad del producto y la disposición para moverse a la siguiente fase del proceso.

#### **Fase de elaboración**

El proceso unificado se enfoca en la importancia de refinar la arquitectura del producto - un conjunto de diseños sobre los cuales el producto de software se construye – a lo largo del desarrollo del proyecto [25]. Así, la meta de la fase de elaboración es crear modelos de diseño y prototipos del producto, así como hacer frente a los riesgos.

Además de definir la arquitectura del sistema, los desarrolladores refinan los requerimientos concebidos anteriormente en la fase de inyección. También desarrollan requerimientos clave y documentación de la arquitectura, como los diagramas de casos de uso (una técnica, consistente en un conjunto de interacciones secuenciales posibles entre usuarios y sistemas para identificar, clarificar, y organizar funcionalidades de un producto), y diagramas de alto nivel [25]. Esta fase da el fundamento a seguir durante el desarrollo.

La construcción de un prototipo generalmente pasa a través de varias iteraciones, antes de que los requerimientos y modelos de la arquitectura se puedan considerar lo suficientemente completos como para continuar.

Al final de la fase de elaboración, los desarrolladores entregan un plan para el desarrollo durante la siguiente fase. Este plan básicamente se construye sobre lo que fue desarrollado durante la fase de inyección e integra todo lo aprendido durante la fase de elaboración de modo que la construcción ocurra efectivamente.

### **Fase de construcción**

Similar a la fase precedente, la fase de construcción tiene iteraciones y se enfoca en construir sobre el trabajo obtenido hasta ese momento. Aquí es donde el producto del software comienza a tomar forma.

Como el proceso unificado es un proceso paralelo [25], cuando la fase de construcción comienza, el trabajo de la fase de elaboración todavía continúa, la única diferencia es que el énfasis en el trabajo cambia al producto de la fase actual.

Las actividades de programación y de pruebas pueden haber sido importantes en la fase de elaboración, pero se vuelven más importantes en la fase de construcción. Similarmente, determinar los riesgos es importante en la fase de inyección, pero es menos importante en la fase de construcción.

En esta fase, los casos de uso completos son desarrollados para conducir el desarrollo del producto. Estos casos de uso son más robustos que los iniciados en la fase de inyección, y son a menudo ideas más específicas sobre cómo el producto debe soportar las tareas del usuario final [25].

El producto se construye de manera iterativa a lo largo de la fase de construcción hasta que está listo para ser entregado. En ese momento, el equipo de desarrollo se inicia la transición del producto al cliente y/o usuarios finales [25].

### **Fase de transición**

Durante esta fase el equipo de desarrollo recibe realimentación de los usuarios, de manera que se revela qué tan acorde está el diseño del producto respecto a las necesidades del usuario [25]. Al recolectar la realimentación de los usuarios, el equipo de desarrollo puede hacer mejoras al producto – por ejemplo, arreglar bugs y siguientes entregas.

Al término de la fase de transición, es posible repetir el ciclo a través de las fases del proceso unificado otra vez, por ejemplo, puede haber una necesidad de crear entregas posteriores del producto o incorporar la realimentación como medio para influenciar los planes para desarrollo posterior [25].

Además, cabe mencionar que las fases están formadas por tareas. Las tareas son la unidad más básica de trabajo que serán administradas, involucran pequeñas piezas de trabajo como escribir código para una característica, diseñar una característica, escribir documentación, instalar una librería, etc. es donde el verdadero trabajo se hace pues son pequeñas unidades de trabajo manejables para el proyecto [25].

Finalmente, para el desarrollo se combinó la metodología elegida (modelo paralelo unificado) con algunas prácticas de programación extrema debido a la interoperabilidad que ambas metodologías permiten y con el fin de mejorar la calidad y capacidad de respuesta a los requisitos cambiantes del cliente.

## **III.II. PROGRAMACIÓN EXTREMA**

### **III.II.I. Definición**

La Programación Extrema (XP, del inglés eXtreme Programming) es una metodología de desarrollo de software ligera, diseñada para ser implementada en equipos de pequeño a mediano tamaño, y para proyectos con requerimientos vagos o rápidamente cambiantes [26]. Además, el cliente debe estar disponible presencialmente con el equipo de desarrollo, de esta manera “sólo se hace lo que se necesite hacer para crear valor para el usuario” [26].

### **III.II.II. Estructura**

Esencialmente, esta metodología se basa en seguir ciertas prácticas y valores para el correcto desarrollo de sistemas. Ya que las prácticas de XP trabajan en conjunto, es recomendable la aplicación de todas para obtener resultados óptimos, sin embargo no es necesario. Así, las prácticas básicas implementadas fueron las siguientes, tomadas del libro Extreme Programming Explained: Embrace Change [26].

#### **Práctica 1: El juego de la planeación**

Tanto el equipo de trabajo como el cliente participan en la planeación del producto; al inicio del proyecto con una sesión larga, y a partir de ahí, en cada iteración mediante sesiones más cortas.

Así fue realizado, y no fue necesaria otra reunión tan extensa sino hasta el final del proyecto, cuando se constató que los objetivos iniciales habían sido cumplidos. Se gestionó la planeación quedando cada revisión entre periodos de alrededor de una semana, incluso a veces menos tiempo, buscando



que se diera prioridad a las características necesarias para el funcionamiento del sistema y ajustando el plan de trabajo de acuerdo a la realimentación recibida.

### **Práctica 2: Pequeñas entregas**

Se procura mantener las iteraciones se sucedan en intervalos cortos con el fin de obtener la mayor realimentación posible. Es por esto que se debe dar prioridad a las características pedidas por el usuario.

Se mantuvieron iteraciones semestrales, con el fin de relegar las fases menos pesadas como son las pruebas de usuarios a las últimas semanas de cada ciclo escolar y que coincidieran con las semanas de mayor trabajo, manteniendo así un equilibrio entre las clases y el proyecto.

### **Práctica 3 Metáfora del sistema:**

Explicar la concepción de un sistema es más fácil si se utiliza una metáfora sobre lo que se plantea, en vez de entrar de lleno con detalles tediosos, sobre todo si se expone a alguna persona que no se desarrolle en el ámbito informático o con conocimiento técnico.

Para conceptualizar el proyecto, se decidió hacer una analogía con la versión física del sistema, es decir, mostrar el proyecto como una libreta en la cual se apuntan datos de salud, y que es escrita por cada persona.

### **Práctica 4: Diseño simple**

Debido a la naturaleza de los proyectos que se suelen manejar con esta metodología – proyectos con requerimientos cambiantes o ambiguos – es conveniente dejar el diseño lo más sencillo posible, pero agradable para el usuario, para no gastar demasiado tiempo en diseñar detalles específicos de la interfaz que posteriormente puedan desaparecer o cambiar.

Al ser un proyecto de tesis, y que por lo tanto es posible que en ocasiones posteriores se actualice el sistema, se pensó en una estructura muy básica para el diseño la cual, además de permitir el avance del proyecto más rápidamente, también evitará al usuario una interfaz complicada que pueda ahuyentarlo.

### **Práctica 5: Pruebas continuas**

Las pruebas deben ejecutarse en intervalos cortos de tiempo, principalmente se realizan dos tipos:

- pruebas unitarias: recaen sobre la programación por lo que se enfocan en tareas muy específicas y son realizadas por el equipo de desarrollo, con el fin de verificar que la funcionalidad a bajo nivel sea correcta y;

- pruebas de aceptación, llevadas a cabo por usuarios o el cliente para verificar que características del sistema funcionen de manera aceptable, por lo que son revisiones algo generales.

Teniendo lo anterior en cuenta, se realizaron tests unitarios en el servidor de pruebas para comprobar el funcionamiento interno del sistema, así como también se utilizó un grupo formado tanto por académicos como por estudiantes, para correr pruebas de aceptación y de usabilidad en el servidor de producción.

#### **Práctica 6: Reestructuración del código fuente (refactorización)**

La meta es mejorar el diseño en pasos pequeños y conforme se vaya necesitando; permitiendo que nuevos requerimientos se agreguen de manera paulatina, y por lo tanto, más fácilmente.

Fue necesario reestructurar el código en múltiples ocasiones, siendo el cambio más importante, el hecho de dejar un sólo archivo como cargador y manejador de todos los demás, no cambiando la funcionalidad del sistema en sí, sino simplemente mejorando el diseño para hacer que el sistema sea más fácil de actualizar e incorporar nuevas secciones.

#### **Práctica 10: Ritmo sostenido (40 horas de trabajo por semana)**

El desarrollo es realizado por los programadores, es así que esta práctica pretende establecer un ritmo de trabajo amigable para el equipo, respetando así tanto su ciclo de trabajo el personal.

Esta práctica fue no sólo recomendable, sino necesaria de implementar. Lo anterior debido a que se tuvo que equilibrar la carga de trabajo con la carga académica, ya que cuando se comenzó la tesis aún se llevaban materias de la carrera. Esta doble carga de trabajo implicaba no poder atender al 100% el proyecto. Fue entonces necesario planear los tiempos de programación, de manera que no interfiriera la tesis con la carrera o que se perdiera el hilo del proyecto.

#### **Práctica 11: Cliente presencial (on-site customer)**

Los requisitos son cambiantes, y por ende, es grande la cantidad de realimentación necesaria para mantener en foco sólo las características de más valor para el cliente. Por ello, es necesario que este último esté situado cerca del equipo de desarrollo o que se tenga fácil acceso presencial a lo largo de todo el proyecto, para que pueda resolver y clarificar dudas sobre el trabajo.

En este caso se contó con la disposición presencial el cliente, siendo el departamento de Informática Biomédica y conformado tanto de ingenieros como de médicos para guiar a lo largo del desarrollo.

Como detalle adicional, los estudiantes de medicina, proporcionaban continua realimentación sobre el proyecto, mismos que se encontraban también en el lugar de trabajo.

### **Práctica 12: Estándares de codificación**

Cuando se trabaja en un equipo de desarrollo, es de suma utilidad tener presente estándares para el software (estilo, formato, uso, etc.) a seguir, y a los cuales, todos los programadores se comprometen cumplir.

En este sentido, si bien no hubo necesidad de contar con un estándar para que no hubiera confusión en el equipo de desarrollo, sí que se usaron varias convenciones en la programación individual; tales convenciones como el uso de la arquitectura MVC hacen que el código sea más fácil de administrar y de leer.

Algunas prácticas como la 7 (Programación en pares), 8 (Propiedad colectiva del código) y 9 (Integración continua), no fueron implementadas debido a que se refieren a equipos de trabajo, de manera que no se aplican para el caso actual al tratarse de desarrollo de manera individual.

### **III.II.III. Beneficios**

Debido a que en XP el diseño del proyecto emerge de manera dinámica, en respuesta a los cambiantes requerimientos y a la refactorización [25], se produce software funcional diligentemente. Sin embargo, esto también representa una desventaja ya que puede implicar la repetición de trabajo al no poner tanto esfuerzo en planificar la arquitectura como en las metodologías lineales.

Cabe mencionar que, al haber combinado el modelo de Proceso Unificado con las prácticas ágiles de la Programación Extrema, se genera un tipo de modelo denominado Modelo de Proceso Unificado Ágil (Agile Unified Process), Agile UP o AUP, sobre el cual no se hará mucho más detalle puesto que ya se han explicado las dos partes que lo componen. Solamente se hará hincapié en que esta metodología enfatiza la simplicidad, empoderamiento de la gente y personalización de la metodología para ajustarse a las necesidades de cada proyecto.

## **III.III. ARQUITECTURA MODELO - VISTA - CONTROLADOR**

### **III.III.I. Definición**

El patrón de arquitectura MVC (Modelo-Vista-Controlador) es un patrón de diseño de software construido sobre la interconexión de tres principales tipos de componentes denominados modelos, vistas y controladores. Se basa en la separación del código en capas diferentes, de acuerdo a las actividades que realice cada tarea; por un lado se tienen los datos de la aplicación y por otro lado se tienen las interfaces del usuario.

### III.III.II. Estructura

De acuerdo a [27], los elementos de la arquitectura se pueden entender de la siguiente manera:

**Modelo:** son objetos que representan datos o alguna actividad de la aplicación. En esta parte de la arquitectura se concentra toda la lógica de negocios, entendiéndose como lógica de negocios cualquier cosa específica sobre cómo la aplicación guarda datos, o usa servicios de terceros, para llenar sus requerimientos de negocios.

**Vista:** permiten visualizar el estado actual del modelo. Estos elementos mantienen todos los elementos de la interfaz de usuario de la aplicación; cualquier cosa que el usuario vea o con la que interactúe será mantenida en una vista. Algunas veces lo que el usuario ve en realidad es la combinación de diferentes vistas en la misma petición.

**Controlador:** permiten cambiar el estado del modelo. Es el componente que conecta los modelos con las vistas, asilando así la lógica de negocios de un modelo de los elementos de interfaz de una vista y permitiendo manejar cómo reacciona la aplicación a la interacción con el usuario.

Los controladores son el primer punto de entrada en este trío de componentes, porque la petición es primero pasada a un controlador, quien después instancia los modelos y vistas requeridos para completar la petición de la aplicación.

### III.III.III. Beneficios

La arquitectura MVC tiene varias ventajas clave, entre las que se resaltan las siguientes [28]:

#### **Diseño de código limpio y claro**

En el desarrollo de pequeñas aplicaciones, es posible evitar sin mucho trabajo el denominado código espagueti o el síndrome de la “gran pelota de barro” (incluso utilizando programación estructurada). Sin embargo, cuando esas pequeñas aplicaciones crecen sin un marco de trabajo, se empieza a tener problemas debido a la gran complejidad el sistema puede llegar tener, y por consiguiente, la complejidad de manejo y actualización.

El acercamiento de la metodología MVC promueve el buen diseño de las aplicaciones desde el comienzo, al apegarse a un diseño limpio y claro mediante el framework, segmentando los objetos en capas, se asegura que cada objeto individual sea mucho más simple y comprensible (para el desarrollador actual y los subsecuentes) cómo interactúan los componentes. Así, un diseño claro representa código fácil de entender y actualizar.

### **Encapsulación de datos y funciones**

Este modelo promueve la encapsulación del modelo de datos dentro de clases bien definidas, manteniéndolo separado de la interfaz de usuario. Esto permite independencia entre, los modelos y lógica de negocios por una parte y por la otra, la interfaz de usuario; de esta manera los cambios entre cada componente estén aislados entre sí, así es que si las reglas de negocios cambian, los cambios pueden ser implementados en varios casos sin tener que modificar los Modelos o las Vistas.

### **Extensibilidad**

Esta característica del modelo, permite que el sistema se expanda indefinidamente sin tener que cambiar una estructura previamente diseñada y fijada. Es decir, tener un modelo por separado permite tener varias UI que puedan ser desarrolladas para el mismo Modelo.

### **Reutilización y refactorización**

Un código base limpio también alienta la eficiencia de la programación y la reutilización inteligente de código; cuando hay un solo objeto responsable de una tarea, es más obvio cuando los deberes están siendo manejados el lugar incorrecto.

Con la arquitectura MVC, hay una natural tendencia a reducir la duplicación de código debido a que es tan obvio dónde el código debería estar. De esta manera, la confusión sobre cómo manejar una cierta manipulación de los datos es menos propensa a suceder.

### **Conclusiones**

En general, la arquitectura MVC fomenta el buen diseño, reutilización o reciclado y refactorización desde las primeras fases del proyecto, en vez de dejar que las malas prácticas hagan que el sistema sea innecesariamente complicado desde la parte de diseño.

## **IV. DESARROLLO**

Ya se explicaron anteriormente la estructura y objetivos de las metodologías utilizadas, por lo cual se detallan a continuación los entregables y avances de cada fase, siguiendo el desarrollo completo del sistema, desde la parte de la inyección hasta la transición final.

### **IV.I. Inyección**

Como se mencionó en la sección de metodología, esta primera parte puede entenderse como el análisis del sistema, ya que durante esta etapa se realizó una serie de entrevistas y reuniones con los médicos, ingenieros y pasantes de servicio social, con el fin de establecer las fases del proyecto. A lo largo de estas conversaciones se fue recopilando información sobre qué abarcaría el proyecto, las características principales y otras funcionalidades que se dejarían de lado para esta entrega, pero que podrían implementarse después.

El caso de negocios del proyecto es el entregable producido en esta fase, la cual duró aproximadamente 1 mes, y recopila los puntos más relevantes de las reuniones llevadas a cabo.

Este entregable, se trata de un documento que sirve como justificación para el proyecto, por lo que debe resolverse la duda sobre si el sistema es viable y factible de realizar (que los beneficios compensen los costos), en cuyo caso se procede a la siguiente fase. Sin embargo si dado el análisis, el proyecto presenta demasiadas complicaciones o riesgos, es el momento oportuno para poder cancelar el proyecto antes de una inversión fuerte en recursos.

Al estar comprendido por varias partes, el caso de negocios suele ser un documento bastante complejo. Sin embargo para este caso, en atención a los principios ágiles (mientras menos documentación, mejor avance), se redujo la información a algunos componentes principales (al ser un trabajo académico e individual, en el cual ya se contaban con los recursos necesarios, se omitieron el presupuesto y los costos), se evitó detallar minuciosamente cada punto.

Para este proyecto, el caso de negocios está compuesto de varias secciones que se detallan a continuación.

#### **IV.I.I. Alcance**

La objetivo principal del alcance es que el cliente y el equipo de desarrollo lleguen a un acuerdo sobre qué tanto abarcará el proyecto, con la intención de separar lo que es relevante de lo que no; una especie de caja que limita lo que se hará en el proyecto con el fin de invertir recursos en los

componentes verdaderamente importantes y evitar desperdicios. Podemos separar el alcance en dos partes: entregables y condiciones límites.

Los entregables son todos los objetos que se presentan como producto del desarrollo; se generaron aquellos recomendados por el AUP, omitiendo los que no fueran aplicables (por ejemplo, objetos relacionados con el presupuesto o la integración del equipo). Estos entregables fueron:

- **Sistema:** el software funcionando de acuerdo a los requerimientos, listo para producción.
- **Código fuente:** El código del programa para el sistema.
- **Scripts de instalación:** Códigos para instalar el sistema en algún ambiente, en este caso, el código de la base de datos.
- **Modelo de requerimientos:** Describe los requerimientos con los que el sistema debe cumplir.
- **Modelo del diseño:** Describe el diseño del sistema.

Las condiciones límite, limitaciones indican hasta dónde se va a llegar con el proyecto, se trata de separar lo aplicable de lo que está afuera del límite, hay tres tipos de limitaciones:

#### **Limitaciones de alcance** (indican cuánto se debe lograr en el proyecto)

- El proyecto constará de una aplicación web, la cual debe visualizarse correctamente en varios tamaños de pantalla a manera de formato de página responsiva. Sin embargo, la posibilidad de tener una aplicación dispositivos móviles queda fuera del proyecto, con la posibilidad de implementarse posteriormente.
- El proyecto afectará únicamente a los alumnos de la Facultad de Medicina, específicamente aquellos cursando la materia de Informática Biomédica, con la posibilidad de expandirse en un principio a los pacientes de los alumnos practicantes.
- El proyecto solamente proveerá la posibilidad de intercambio de datos con el ECEU, el cómo manipula este último los datos obtenidos, queda fuera del proyecto.
- El proyecto contempla únicamente los perfiles de usuario y administrador, pudiendo implementarse posteriormente un perfil médico.

#### **Limitaciones de tiempo** (indican la duración del proyecto)

- El límite de desarrollo del proyecto es de 2 años para tener una versión funcional y que se haya probado por los alumnos.

#### **Limitaciones de recursos** (indican cuánto se invertirá y con qué se cuenta en el proyecto)

- El proyecto deberá realizarse con los recursos disponibles del departamento de IB, es decir: computadoras, estudiantes de servicio social, médicos del departamento, etc.

- El proyecto será realizado con software de código libre o de licencias gratuitas.

#### **IV.I.II. Requerimientos**

Primero se tiene que saber qué es lo que el cliente necesita, y así definir lo que el sistema debe incluir para satisfacer sus necesidades. Para este fin, se enuncian los requerimientos; descripciones de lo que el sistema debe hacer, los servicios que provee y las limitaciones en su operación [24]. Para entender mejor los requerimientos, se definen a continuación, los perfiles de uso del sistema:

**Perfil de usuario:** utiliza el sistema para registrar su información y consultarla una vez procesada por la aplicación.

**Perfil de administrador:** utiliza el sistema para darle mantenimiento y actualización, ya sea para gestionar usuarios o modificar información del mismo.

Teniendo los perfiles en cuenta, se agruparon los requerimientos de acuerdo al nivel de detalle; requerimientos de alto nivel, también denominados de usuario y, requerimientos de bajo nivel también denominados del sistema. Esta separación se debe a que tener diferentes niveles de requerimientos es útil porque comunican información del sistema a diferentes tipos de lectores que, de acuerdo al nivel de detalle, pueden utilizarlos de maneras diferentes [24].

**Requerimientos del usuario.** Es la serie de descripciones en lenguaje natural, sobre los servicios que el sistema debe brindar así como sus restricciones. Deben estar expresados de tal manera que sean comprensibles para los usuarios de lo sistema sin conocimiento técnico detallado, tocando solamente el comportamiento externo del sistema, evitando las características del diseño. Debido a su alto nivel de abstracción, un requerimiento de usuario puede ser desglosado en varios requerimientos del sistema [24].

Finalmente, se presentan en la **Fig. 4.1** los requerimientos recopilados en las reuniones mediante un diagrama de casos de uso primario.





Fig. 4.1. Diagrama de casos de uso del sistema

**Requerimientos del sistema.** Son descripciones a detalle sobre cada una de las funciones del sistema, mediante las cuales se detalla exactamente lo que va a implementarse; son objetivos cuantificables.

A continuación se describen los requerimientos definidos de manera general, de acuerdo al orden de prioridad asignado, de mayor a menor importancia. Se encuentran numerados de acuerdo al diagrama de casos de uso mostrado **Fig. 4.1** y se mencionan los actores que intervienen en la tarea.

Primero se enlistan los requerimientos de **alta prioridad**; aquellas características críticas para la misión del proyecto e indispensables para la entrega actual del sistema,

#### 1.1 Registro al sistema:

Cualquier visitante a la aplicación tendrá la posibilidad de crear una cuenta para ingresar al sistema; para este fin se pedirán algunos datos como nombre de usuario y dirección de correo. Una vez realizado lo anterior, se le enviará al usuario un correo en el cual se indicarán sus datos de acceso (se le asignará por defecto una contraseña segura que luego podrá cambiar). Actores: Usuario.

#### 1.2 Acceso al sistema:

Quando el usuario haya creado una cuenta, podrá acceder al sistema utilizando el nombre de usuario que se establezca para el sistema (correo, CURP, etc.) y la contraseña creada por el usuario. Actores: Usuario/Administrador.

#### 2.1 Restauración de contraseña:

En caso de que el usuario haya extraviado la contraseña, podrá pedir que le envíen un enlace a su correo para restablecerla. Hay que mencionar que, por cuestiones de seguridad, los usuarios no podrán recuperar sus contraseñas ya que éstas se encuentran cifradas de manera irreversible [29] en la base de datos evitando así el que otro usuario o incluso un administrador pueda apropiarse de las claves de acceso. Actores: Usuario.

#### 2.2 Cambio de contraseña:

A través de la sección de perfil, los usuarios podrán cambiar su contraseña ingresando su clave actual. Actores: Usuario/Administrador.

#### 3.1 Registro de ficha de identificación:

En esta sección el usuario podrá registrar los datos más básicos sobre su persona como nombre completo, grupo sanguíneo, etc. Actores: Usuario/Administrador.

#### 3.2 Registro de información de medicaciones:

En esta sección el usuario podrá indicar los medicamentos que está tomando, así como las dosis y fecha de duración del tratamiento. Actores: Usuario.

#### 3.3 Registro de mediciones corporales:

En esta sección el usuario podrá registrar los datos que obtenga de su cuerpo, tales como peso altura, presión arterial, etc., independientemente de si son proporcionados por algún profesional del cuidado de la salud, o por sí mismo. Actores: Usuario.

#### 3.4 Registro de antecedentes:

Esta sección es muy amplia y abarcará el registro de muchos conceptos, como enfermedades familiares, hábitos, condiciones de acuerdo al género, etc. Actores: Usuario.

#### 3.5 Registro de observaciones cotidianas

En caso de que el usuario desee apuntar algo más sobre su salud y no haya un espacio específicamente destinado para ello en el sistema, se provee la posibilidad de registrar texto de formato libre. Estos textos se denominan observaciones cotidianas y deberán aparecer de manera fácilmente visible para el usuario, siendo conveniente agruparlos en la pantalla de inicio en orden cronológico descendente. Actores: Usuario.

#### 4.1 Visualización de información registrada:

En todas las secciones en las que el usuario registre información, los datos serán visibles de tal manera que, en posteriores consultas, se muestren las modificaciones realizadas. Actores: Usuario/Administrador.

#### 4.2 Resumen de la información registrada en la pantalla de inicio:

Al momento de iniciar sesión, el usuario será dirigido a la página principal, en la cual aparecerá información procesada a partir de los datos que se hayan registrado previamente. Actores: Usuario/Administrador.

#### 5.1 Gestión de usuarios:

El administrador del sistema podrá eliminar usuarios y ver la información más general de cada uno (ficha de identificación), sin tener posibilidad de ver más detalles. Actores: Administrador.

#### 8.1 Proveer conexión con el ECEU:

Tanto el sistema como la base de datos estarán configurados internamente de tal manera que la compatibilidad con el ECEU pueda darse de manera sencilla y con pocas adecuaciones.

A los requerimientos anteriores les siguen los de **media prioridad**, que asisten a operaciones necesarias del sistema, y son requeridos eventualmente, pero pueden ser relegados a una entrega posterior en caso de ser necesario.

#### 2.3 Cambio de imagen de perfil:

A través de la sección de perfil los usuarios pueden modificar la imagen de perfil proporcionada por alguna otra imagen de su elección, eligiéndola de los archivos locales. Actores: Usuario/Administrador.

#### 7.1 Registro de comentarios sobre sistema:

Los usuarios podrán registrar comentarios sobre el sistema desde la página de inicio, además, será obligatorio la primera vez que cierren sesión.

#### 6.1 Envío de avisos por correo.

Se podrá homologar la comunicación con los usuarios proveyendo una interfaz que permita enviar correo de manera simultánea – sin embargo de forma individual – a todos los usuarios registrados con información relativa al sistema. Actores: Administrador.

#### 6.2 Seguimiento de comentarios hechos por usuarios.

Se deberá tener un módulo mediante el cual se puedan agrupar los comentarios con el fin de emplearlos de mejor manera, teniendo secciones primero generales como comentarios resueltos y no resueltos, y posteriormente otras secciones más específicas como problemas urgentes, problemas resueltos, sugerencias, etc. Actores: Administrador.

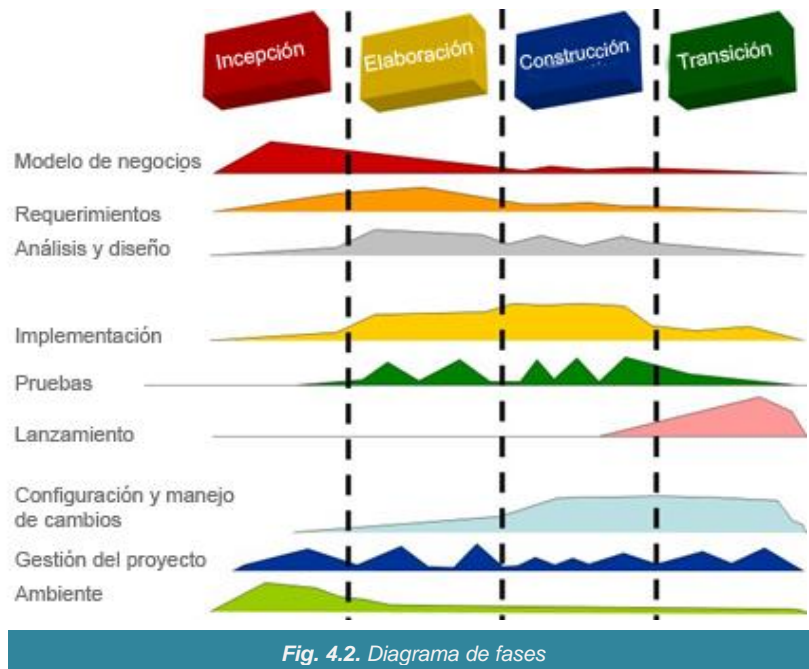
Finalmente, los requerimientos de **baja prioridad** son mejoras funcionales o cualitativas; características que sería bueno agregar en futuras entregas si los recursos lo permiten.

5.1 Las cuentas se eliminarán después de cierto tiempo de inactividad; primero, al año de inactividad se enviará un correo de recordatorio, y a los 5 años se eliminará permanentemente la cuenta.

4.3 Uso de un diagnosticador de enfermedades para ayudar al usuario general a tener una idea sobre enfermedades comunes, y sobre todo de bajo riesgo que pudieran presentársele.

#### IV.I.III. Plan de fases e iteraciones posteriores

Se propone el AUP como metodología de desarrollo del software pues concuerda o es fácilmente adaptable a los tiempos y horarios establecidos para el desarrollo del sistema, así como al tiempo de desarrollo del proyecto mostrado en la siguiente figura (Fig. 4.2).



#### IV.I.IV. Riesgos potenciales asociados al proyecto

Un riesgo es alguna situación que puede suceder a lo largo del proyecto y que, en caso de que suceda, puede afectar el desarrollo proyecto. Para efectos del proyecto de tesis, se omitieron los posibles riesgos positivos que pudieran suceder.

Se denominaron como principales riesgos negativos asociados al proyecto los siguientes, clasificándose de acuerdo al impacto que pudieran tener:

**Riesgo alto**, aquellas situaciones que en caso de suceder, es muy probable que el proyecto no cumpla los objetivos establecidos, por lo que debe haber un gran enfoque sobre estos eventos.

**Riesgo medio**, aquellas situaciones cuyo suceso impacta de manera parcial el resultado del proyecto y que deben ser vigiladas, pero no supervisadas de manera tan minuciosa como las anteriores.

**Riesgo bajo**, aquellas situaciones que aún en caso de presentarse, no implican una amenaza o un ataque importante para el proyecto.

Habiendo explicado lo anterior, se enlistan de mayor a menor impacto los posibles riesgos, con su clasificación y probabilidad de ocurrencia (probable o improbable):

- a. Poco interés de los usuarios. (**riesgo alto y probable**)
- b. Falta de concordancia entre cliente y desarrollador (medicina e ingeniería). (**riesgo alto y probable**)
- c. Retrasos del proyecto (**riesgo medio e improbable**)
- d. Disminución de la calidad debido a gran duración del proyecto (**riesgo medio e improbable**)
- e. Resistencia al cambio (**riesgo medio y probable**)
- f. Fallas en los servicios básicos (**riesgo bajo e improbable**)

#### **IV.I.V. Visión general de la arquitectura de software propuesta**

Se entiende esta arquitectura, como un modelo abstracto reutilizable que sirve de guía para la construcción de software. Establece la estructura, funcionamiento e interacción entre las partes del programa desde un diseño de alto nivel.

De acuerdo a los requisitos anteriores se decidió en primer lugar, que un único patrón sería suficiente para representar la arquitectura, en vez de complicarse con varios enfoques como sucede en proyectos más grandes.

En segundo lugar, se optó el patrón arquitectónico MVC (Modelo-Vista-Controlador) de acuerdo a las siguientes razones:

- El patrón se adecua perfectamente con la metodología de desarrollo de software seleccionada (AUP) al promover la separación de vistas, lógica de negocios y datos, en capas separadas. Esta separación hace que el desarrollo incrementable e iterable de acuerdo a AUP sea más fácil, pues con un sistema separado en módulos se pueden agregar funcionalidades, que además pueden ser reutilizables, sin afectar necesariamente las características anteriores.

- Se cuenta con experiencia en este patrón al haberse utilizado en proyectos previos, habiendo probado su eficacia y comodidad en términos de desarrollo.
- Es un patrón que siendo abstraído a una mínima expresión, es fácil de comprender y por lo tanto sencillo de explicar, incluso a personas ajenas al ambiente informático.
- Se piensa, en una versión posterior, escalar el proyecto a Angular.js si se cree así conveniente. Para lo cual, la estructura sería muy similar pues este framework (Angular.js) se basa en el MVC.
- El registro de información para cualquier sección debe poder realizarse de manera muy sencilla para no incomodar al usuario, para lo cual, el uso de MVC resulta conveniente, pues en caso de no funcionar alguna vista, ésta se pueda cambiar sin afectar el funcionamiento interno.

#### **IV.I.VI. Requerimientos de hardware y software**

Para la implementación de la aplicación no se requerían características muy específicas del hardware por lo cual, siendo el proyecto un sistema web, podía implementarse en el mismo servidor que almacenaba las otras aplicaciones del laboratorio, el cual ya contaba con el software necesario para correr el sistema. Se utilizó una máquina con las siguientes características:

<b>Tipo de sistema operativo</b>	GNU/Linux Debian
<b>Arquitectura de CPU</b>	x86_64
<b>Tamaño de disco duro (HDD)</b>	1 Tb
<b>Memoria RAM</b>	8 Gb
<b>Nombre del procesador</b>	Intel (R) Xeon (R) CPU E5-2440 v2
<b>Velocidad del procesador</b>	1.90GHz
<b>Núcleos del procesador</b>	16

La forma como obtuvo esta información fue a través del uso de comandos en el servidor; para obtener el espacio de disco duro disponible y total de la máquina se utilizaron los comandos *uname -a* y *free -m*. Para obtener los datos sobre las capacidades de la máquina (procesador, memoria ram, etc.) se usó *cat /proc/cpuinfo*.

#### **IV.I.VII. Entregables**

Concluyendo esta fase, se determinó que el proyecto era factible de realizar y que además podría contribuir hasta cierto punto a mejorar la calidad de vida de la gente. También se determinó que la solución propuesta, es decir el sistema con las especificaciones de desarrollo y planeación analizadas, era un acercamiento que podría funcionar.

## IV.II. Elaboración

Esta fase que duró alrededor de 2 meses (aunque es a partir de este punto que las fases se intercalan e iteran, de manera que el tiempo es estimado), está primordialmente enfocada al diseño; toma como base los requerimientos, características y limitaciones definidas y aprobadas en la fase anterior, y su entregable principal es un primer diseño del sistema. Más específicamente, el entregable es un prototipo del funcionamiento del sistema, habiendo definido ya la arquitectura a emplear y con la estructura general para el sistema.

### IV.II.I. Estructura del sistema

El primer paso en esta fase fue seleccionar las posibles áreas que conformarían el sistema, tomando la estructura del *Interrogatorio* de la 3a edición del *Manual de Historia Clínica Académica de la Facultad de Medicina de la UNAM* [30]. El sistema fue dividido, visto de manera muy general, en las siguientes secciones:

**Ficha de identificación**, la cual contiene datos ordenados como se muestra a continuación.

Datos básicos; implican aquella información inherente a la persona, que como tal permiten identificarlo.

- Nombre completo del paciente: separado en tres campos (nombre[s], apellido paterno y apellido materno) para hacer más fácil su manejo.
- Grupo sanguíneo: se especifica el antígeno o sistema AB y factor o sistema RH, por lo que las posibles opciones a seleccionar son: (O-, O+, A-, A+, B-, B+, AB-, AB+)
- Género: se presentan dos opciones; masculino o femenino.

Nota: es importante resaltar que la sección de antecedentes de género mostrará antecedentes andrológicos o antecedentes gineco-obstétricos dependiendo de la opción que se seleccione aquí.

Fecha y lugar de nacimiento; es muy importante para el médico conocer cuántos años tiene el paciente porque ciertas patologías y medicaciones que prevalecen de acuerdo a la edad.

- Fecha de nacimiento: separada en tres campos (día, mes y año).
- País de nacimiento: se elige de una lista predefinida; de un catálogo registrado en la base de datos con cientos de países y su respectivo código de acuerdo al estándar ISO 3166-1 alpha-2.
- Entidad federativa de nacimiento: se elige de una lista desplegable con los valores del catálogo registrado en la base de datos de acuerdo al listado oficial del INEGI<sup>1</sup>.

---

<sup>1</sup> <http://www.inegi.org.mx/geo/contenidos/geoestadistica/catalogoclaves.aspx> Revisado en octubre de 2016

- Ciudad de nacimiento: este campo está abierto a texto libre debido a la enorme cantidad de ciudades que existen y a la falta de un catálogo completo, en futuras implementaciones podría buscarse un catálogo para la base de datos.
- Delegación o municipio de nacimiento: se elige de una lista desplegable con los valores del catálogo registrado en la base de datos de acuerdo al listado oficial del INEGI<sup>1</sup>.

Domicilio; datos sobre el lugar de residencia del paciente que pueden orientar al médico en el diagnóstico de enfermedades. Estos campos son de texto abierto y son bastante explicativos por lo que no se hará mucho énfasis.

- País: se elige de una lista desplegable con los valores del catálogo registrado en la base de datos de acuerdo al listado oficial del INEGI<sup>1</sup>.
- Entidad federativa: se elige de una lista desplegable con los valores del catálogo registrado en la base de datos de acuerdo al listado oficial del INEGI<sup>1</sup>.
- Ciudad de nacimiento: este campo está abierto a texto libre debido a la enorme cantidad de ciudades que existen y a la falta de un catálogo completo, en futuras implementaciones podría buscarse un catálogo para la base de datos.
- Delegación o municipio: se elige de una lista desplegable con los valores del catálogo registrado en la base de datos de acuerdo al listado oficial del INEGI<sup>1</sup>.
- Colonia: campo de texto libre.
- Calle: campo de texto libre.
- Número de calle: campo de texto libre.
- Código postal: campo de texto libre.

Contacto; en esta sección se encuentra información que puede facilitar al médico la comunicación con el paciente; el correo para comunicación rutinaria, y los teléfonos en caso de urgencia:

- Teléfono: este campo solo acepta números.
- Teléfono móvil: este campo sólo acepta números.
- Correo electrónico: este campo no se puede modificar ya que es el identificador del usuario en el sistema, también a través de este medio se le pueden hacer llegar avisos o anuncios provenientes de la administración del proyecto.

Otros datos; esta última parte de la ficha de identificación recopila algunos otros datos que no fueron englobados en ninguna categoría en particular y que pueden brindar también información importante al médico pueden reflejar un poco de la vida del paciente.

- Estado civil: se puede seleccionar de una lista desplegable; un catálogo con las opciones soltero(a), casado(a), divorciado(a), viudo(a) o unión libre.
- Escolaridad: con las opciones de educación primaria, secundaria, bachillerato, licenciatura, maestría, doctorado o ninguna.
- Religión: se presentan primero la lista de las religiones más comunes, generales (catolicismo, cristianismo, judaísmo, islamismo, hinduismo, budismo u otra).



- Nacionalidad: campo de texto libre.
- Ocupación: se tienen por el momento 6 secciones que se pueden seleccionar: pensionado, desempleado, jubilado, empleado, estudiante u otra. Sería conveniente en un futuro expandir el abanico de opciones.

### **Antecedentes**

Antecedentes heredo-familiares; se comparten genes, hábitos y ciertas características entre familiares, incluso ciertas enfermedades o aspectos corporales, por lo que el conocimiento de la historia familiar puede alertar de posibles riesgos a la salud del paciente.

Se presentan en esta sección los antecedentes agrupados en 4 categorías: relativos a abuelos paternos, a abuelos maternos, al padre, a la madre u otros. Una vez seleccionado el familiar se pueden marcar las siguientes opciones de patologías: enfermedades infectocontagiosas, enfermedades tiroideas, diabetes mellitus, neurológicas, mentales, hipertensión arterial, cardiopatías y nefropatías.

Se pusieron estas opciones por ser enfermedades muy comunes, sin embargo, también se agregó un espacio denominado “Otras enfermedades” (un campo de texto libre), con el fin de poder cubrir un espectro más amplio.

Antecedentes personales patológicos; esta sección contiene las enfermedades y afecciones que ha tenido el paciente a lo largo de toda su vida; evidentemente no puede estar registrada toda la información, pero al menos los eventos más importantes, significativos o con mayores secuelas deben registrarse. Todos los campos son de tipo texto libre: enfermedades de la infancia, alergias, transfusionales, quirúrgicos, hospitalizaciones, traumatismos y fracturas, consumo de tabaco, consumo de alcohol, consumo de drogas.

Antecedentes personales no patológicos; se mencionan los campos junto con una pequeña instrucción de su llenado (misma que aparece en el sistema) tal como fue tomada del manual mencionado anteriormente.

- Alimentación: cantidad y frecuencia en el consumo de alimentos por semana: leche, carne, huevo, verduras, frutas, cereales, leguminosas, etcétera. Balanceada en calidad y cantidad realizando 3 comidas al día, consumo de 1L de agua al día. Texto libre.
- Habitación: habita en casa propia/rentada, tipo de vivienda (jacal, departamento, vecindad, casa sola, tipo urbana que cuenta con todos los servicios intra y extradomiciliarios, construida de materiales perdurables), distribución de la vivienda (número de cuartos y

servicios, número de personas por habitación, convivencia con animales, tipo y número), higiene de la vivienda (iluminación, adecuada ventilación). Texto libre.

- Hábitos higiénicos individuales: aseo personal, baño, cambio de ropa, lavado de manos, aseo dental. Texto libre.
- Ocupación actual y previa: fecha y duración; condiciones del trabajo, horas que labora, higiene laboral, exposición a factores de riesgo laboral. Texto libre.
- Uso de tiempo libre: horario de descanso y recreación, deportes y pasatiempos, vacaciones. Texto libre.
- Inmunizaciones: vacunas y número de dosis (Sabin, DPT, pentavalente, BCG, etcétera), Biológicos (suero antirrábico, antialacrán, anticrotálico, gammaglobulina, anti-Rh. Texto libre.
- Actividad física: texto libre.
- Viajes recientes: texto libre.
- Tatuajes: texto libre.
- Perforaciones: texto libre.
- Exposición tóxica: texto libre.

Antecedentes de género; ciertas enfermedades son inherentes a las características sexuales del individuo por lo que esta sección está inhabilitada hasta que se elija el género. Una vez seleccionada una opción se muestran las siguientes opciones dependiendo de si es masculino o femenino, respectivamente.

- Andrológicos (circuncisión, criptorquidia, poluciones nocturnas, inicio de VSA, número de parejas, enfermedades de transmisión sexual, trastornos de la erección y andropausia).
- Gineco-obstétricos (menarca, ciclo menstrual, inicio de VSA, número de parejas, número de embarazos, número de partos, abortos, cesáreas, método anticonceptivo, fecha de última menstruación, enfermedades de transmisión sexual, menopausia, climaterio, Papanicolau, y lactancia materna).

## **Medicamentos**

En esta sección se hace el registro y control de los medicamentos que el usuario está tomando, para ello deberán tenerse los campos que a continuación se enlistan.

- Medicamento: se elige el nombre del medicamento/principio activo recetado de una lista desplegable.
- Vía de administración: se elige la ruta de entrada del fármaco de una lista desplegable.
- Dosis: campo de texto libre.
- Frecuencia: se registra en dos campos la frecuencia de la dosis; en el primer campo se registra el intervalo de horas, y en el segundo el intervalo de días.

- Fecha de inicio: se registra la fecha en la que se comenzó a tomar el medicamento.
- Fecha de término: este campo se actualiza automáticamente al cambiar el campo de intervalo de días y tomando en cuenta la fecha de inicio.

## **Mediciones**

En esta sección se muestran todas las mediciones registradas por el usuario, por lo que es meramente un histórico de las medidas corporales que el usuario ha registrado desde la página principal. Se muestra cada medición con la fecha de registro y las unidades de medida.

### **IV.II.II. Resolución de riesgos altos**

Poco interés de los usuarios: con el fin de evitar que el proyecto fracasara por este inconveniente, se planeó encuestar constantemente a los usuarios de prueba con el fin de obtener retroalimentación sobre qué aspectos agradan y cuáles no del sistema.

Falta de concordancia entre cliente y desarrollador (medicina e ingeniería): esta es una de las razones por las que se pretendió usar la metodología AUP, ya que esta promueve la participación activa del cliente a lo largo del desarrollo del proyecto. Con este fin, se planearon realizar reuniones semanales para revisar los avances y presentar las características que se fueran agregando.

### **IV.II.III. Herramientas utilizadas**

Se presentan las tecnologías y software necesarios para desarrollar el código del sistema y una breve explicación sobre porqué se utilizó cada herramienta.

## **HTML**

HTML o Hypertext Markup Language (en español, lenguaje de marcado de hipertexto) es la especificación sobre cómo deben ser escritas las páginas web de manera que puedan ser entendidas y correctamente desplegadas por computadoras u otros dispositivos capaces de conectarse a internet [31].

HTML es el rey indiscutible de los lenguajes de marcado de etiquetas, ya que no hay alternativa para páginas web, o por lo menos no una que sea factible de usar en producción. Esto es por cuestiones de estandarización, ya que el etiquetado expresa el tipo de información del texto u de otro contenido de la página.

## **PHP**

PHP es el acrónimo recursivo de Hypertext Preprocessor (en español, preprocesador de hipertexto) es un lenguaje programación (más específicamente un lenguaje de scripting) embebido, lo que

significa que puede ser intercalado dentro de HTML y hacer así el desarrollo de sitios web dinámicos bastante accesible.

Esta herramienta es un lenguaje de secuencias de comandos y es diseñado para hacer algo solamente hasta que un evento ocurre – por ejemplo, cuando un usuario envía un formulario que va hacia una URL [32].

PHP tiene un **excelente desempeño**, una estrecha integración con casi cualquier base de datos, disponibilidad, estabilidad, portabilidad, y una casi ilimitada función debido a su extensibilidad [32].

Además, PHP es **un lenguaje de código abierto** (lo que implica ningún costo) y con una curva de aprendizaje muy manejable, lo que le ha llevado a tener crecimiento exponencial desde su concepción. Se mencionan finalmente algunas de las principales ventajas de este lenguaje de acuerdo a [33]:

Al ser un lenguaje interpretado, es hasta cierto punto, **independiente del entorno** donde se desarrolle, por lo que puede usarse prácticamente en cualquier plataforma.

Aunque no dispone de soporte comercial, hay una **extensa comunidad en línea** que puede guiar y ayudar en la resolución de problemas en el desarrollo con esta herramienta.

Existe una **variedad de recursos** en la web que pueden facilitar el desarrollo de aplicaciones, ya que existen bibliotecas de clases que resuelven los problemas más frecuentes con los que se puede encontrar el programador.

## **PDO**

PDO son la siglas de *PHP Data Object* (en español, objeto de datos de PHP) y es una extensión de PHP que utiliza controladores para acceder a los datos de motores de bases de datos, actualmente de hasta 12 tipos distintos de bases como MySQL, PostgreSQL, Microsoft SQL Server, Oracle, mSQL, SQLite, etc [34].

La **generalidad** de PDO provee una capa de abstracción de datos, con la cual, la codificación depende menos de la base de datos que se esté usando específicamente. Esto se debe a que los métodos tienen el mismo nombre para todas las bases de datos que soporta [34].

También, una de las principales ventajas de PDO es la **seguridad** con el uso de sentencias preparadas (prepared statements). Esta característica da la posibilidad de vincular parámetros, con

lo cual se evita tener que poner entre comillas los parámetros y por consiguiente, se protege el sistema contra inyecciones SQL u otros problemas similares.

### **JavaScript**

JavaScript es uno de varios lenguajes de programación que están en uso actualmente, y el cual es principalmente conocido como la tecnología que permite a las páginas web volverse “dinámicas” o “interactivas” [35].

### **CSS**

CSS, como se conoce más comúnmente al término *Cascading Style Sheets* (en español, hojas de estilo en cascada) es un lenguaje que usa reglas, llamadas estilos, para determinar la presentación visual de los sitios web [31]. Hay que mencionar que, para fines prácticos, actualmente CSS es la única especificación utilizada para dar estilo a páginas web.

Este lenguaje da control creativo sobre la estructura y diseño de la página, y trabaja mano a mano con el navegador para hacer que el HTML luzca agradable y sea cómodo para el usuario [36].

Permite separar el contenido de la presentación; por un lado la estructura se define en el documento HTML y por otro lado, la representación pasa a ser controlada por las hojas de estilos ligadas a la página [37].

Permite homogeneizar el diseño gracias a la definición de criterios que deberán respetarse en todo el sitio. Esto es posible gracias a las *clases*, que pueden declararse de manera muy general y aplicarse a elementos similares sobre cualquiera de los archivos del proyecto.

Promueve la reutilización y el reciclado de código pues ofrece la posibilidad de que se definan clases para evitar la necesidad de rescribir código [37].

Permite la herencia de propiedades, lo que significa que muchas propiedades son transmitidas, de manera inherente, de un elemento padre a un elemento hijo, esto implica que se ahorre mucho tiempo de desarrollo y código al no tener que declarar el estilo para cada uno de los elementos del documento.

Permite también, gracias a la especificidad, dar estilo a un elemento siendo lo más concreto posible, evitando así el que pueda haber ambigüedades o equivocaciones sobre a qué elementos se va a aplicar el formato.

Es precisamente la unión de estos dos últimos conceptos lo que da al CSS la característica de **estilo en casada**, y que se refiere, de manera muy básica, a que el navegador irá aplicando propiedades y sobrescribiendo las anteriores, de tal manera que al final el diseño será el más adecuado a nuestras necesidades. Esto permite crear reglas para que el sitio se represente de una manera correcta en diferentes dispositivos.

## SQL

SQL es la abreviación de *Structured Query Language* (lenguaje estructurado de consultas), el lenguaje más comúnmente utilizado para, a través del DBMS, consultar y modificar la base de datos.

Este lenguaje sirve tanto para la manipulación como para la definición de datos [38]; incluye las sentencias para modificar una base de datos mediante funciones; funciones como: insertar y eliminar tuplas, formar relaciones y declarar el esquema de la base, por mencionar algunas.

SQL usa una sintaxis libre de forma que le da al usuario la flexibilidad de estructurar sentencias en la manera que mejor se le adapte, además, cada petición es analizada por el DBMS antes de la ejecución, para verificar la sintaxis y, en caso de ser posible, optimizar la misma petición [39]. Algunas de las mayores ventajas de SQL de acuerdo a son:

A diferencia de los lenguajes por procedimientos o lenguajes procedurales, los cuales requieren instrucciones de navegación para recuperar los datos, SQL está codificado sin datos embebidos. Esto implica que el programador sólo necesita especificar qué dato es requerido y cómo recuperarlo, de lo cual ya se ocupa el DBMS [39].

Aplicaciones escritas en SQL pueden ser fácilmente portadas entre sistemas. Esta portabilidad puede ser requerida cuando los DBMS subyacentes necesitan ser actualizados o cambiados [39].

SQL como lenguaje provee un mayor grado de abstracción y es independiente de la manera en la que se implementa internamente; una consulta regresa el mismo resultado sin importar si la optimización ha sido hecha con índices o no.

El lenguaje es simple y fácil de aprender, pero también puede resolver situaciones complejas, ya que los resultados esperados están bien definidos y el lenguaje tiene una sólida base teórica. Con estas bases, hay ambigüedad sobre la manera en la que una consulta interpretará los datos y producirá el resultado.

## **Apache**

Apache es el servidor web (servidor HTTP) dominante en internet hoy en día, ya que está ampliamente extendido a través sitios web que requieren manejar bases de datos o algún tipo de programación ya que se ajusta a sitios de todos los tamaños y tipos.

Unas de sus principales características es que es freeware (software gratuito), pero también es un programa de código abierto, lo que significa que el código fuente puede ser examinado por cualquiera que así lo desee.

Además, al igual que con PHP, este servidor cuenta con un amplio respaldo en línea a pesar de no tener soporte de tipo comercial, de manera que si hay errores, miles de usuarios pueden analizar el código para encontrar errores.

Gracias a su alto grado de configuración, Apache puede correr desde un solo sitio personal hasta un enorme sitio sirviendo a millones de visitantes regulares. Puede usarse para servir archivos estáticos sobre la web o como front-end a aplicaciones que generan respuestas personalizadas a los visitantes [40].

Es tan común este servidor que muchos desarrolladores usan Apache como servidor de prueba en sus máquinas, probando el código en un ambiente local, antes de publicarlo a una audiencia más extensa [40].

## **Notepad++**

Notepad++ es un editor de texto gratuito para la plataforma Windows, cuya interface es bastante similar a la del bloc de notas regular pero con características avanzadas como sintaxis codificada por colores, indentación de código, y numeración de líneas [41], lo que lo convierte en una aplicación extremadamente útil y práctica para escribir y editar código.

Esta herramienta es muy simple, liviana y efectiva, por lo que es comúnmente utilizada, y cuenta con buena reputación, en ambientes de desarrollo web.

Además, Notepad++ se enmarca bajo la licencia GNU (es una aplicación gratuita) y brinda la posibilidad de escribir código fuente de una gran variedad de lenguajes como CSS, JavaScript, PHP, Ruby, Perl y SQL, entre otros, por lo que es comúnmente utilizada para proyectos de programación de baja escala de casi cualquier índole.

Otras características de este programa son: envoltura de sintaxis, niveles de zoom, soporte para macro, disponibilidad de utilizar extensiones y la disponibilidad en diferentes idiomas, incluso en español [37].

## **MySQL**

Se utilizó en un principio MySQL, debido a la facilidad de uso, sin embargo, para usos posteriores, se planea utilizar PostgreSQL debido al buen soporte que proporciona a bases de datos con grandes volúmenes de información y transacciones pesadas.

### **IV.II.IV. Entregables**

Teniendo ya una idea de las secciones que abarcaría el sistema y de las tecnologías que se utilizarían para el desarrollo e implementación, se presentó un diseño a mano, primer prototipo de carácter ilustrativo, en el cual se proponían las distintas vistas para el proyecto así como también los posibles nombres y logos para el proyecto.

### **IV.III. Construcción**

Teniendo una visión general del problema y de cómo sería la posible solución - es decir, el diseño generado en la fase de elaboración – se llega a la fase más robusta (en cuanto a tiempo, avances y recursos) del proyecto; la construcción del sistema.

El principal entregable de esta fase (de este conjunto de fases iterativas) es el sistema funcionando pues, a partir de este punto, ya se cuenta con información suficiente para comenzar con la configuración del entorno de desarrollo y la posterior programación del sistema en sí. A continuación se detalla el proceso que se siguió para tener el servidor con las herramientas necesarias para poder ejecutar la aplicación.

Primero hay que mencionar que, dado que se decidió desarrollar el proyecto en el servidor del departamento y que éste ya corría aplicaciones con requerimientos similares a los de este sistema, la parte de instalación de software del lado del servidor no fue necesaria. También, el departamento contaba con un administrador de sistemas y fue quien creó las cuentas a utilizar en la máquina, tanto de la base como del sistema en sí.

Dado lo anterior, lo que se hizo inicialmente en esta etapa fue por una parte, revisar que las versiones del software y las capacidades del hardware fueran útiles para el sistema, y se crearon los usuarios para la base de datos y para el sistema operativo, con los permisos y características necesarias, y por otra parte, se configuró y se instaló el software necesario en las máquinas donde se desarrollaría el proyecto, es decir, los servidores de desarrollo.



### IV.III.I. Configuración

Para configurar el entorno, primero se obtuvo la información de las características del servidor del departamento de IB y desde una computadora del mismo departamento, tal como se muestra en la **Tabla 4.1**.

**Tabla 4.1**

*Comandos usados para obtener las características del servidor*

Comando	Uso	Información obtenida
uname	Imprime información relativa al sistema actual.	Se obtuvieron el nombre, entrega y versión del kernel, el nombre del equipo (hostname), nombre y plataforma del hardware, tipo de procesador, nombre del sistema operativo.
df	Muestra información relativa al espacio del disco duro.	Se obtuvo la cantidad de memoria disponible en disco duro, así como los puntos de montaje y el sistema de ficheros.
free	Muestra la cantidad de memoria RAM y SWAP, disponible y en uso.	Se obtuvo la cantidad de memoria con la que cuenta la máquina, así como una muestra de qué tanto se usa actualmente para otros proyectos.
php	Administración de PHP.	Se utilizó para comprobar la versión de PHP y que este servicio estuviera funcionando adecuadamente.
mysql	Administración de MySQL	Se utilizó para comprobar la versión de MySQL y que este servicio estuviera funcionando adecuadamente. También con este comando pudieron revisarse las dos bases de datos (producción y desarrollo) del proyecto.

También se consultaron varios archivos de configuración con el propósito de obtener la mayor cantidad de información posible (**Tabla 4.2**).

Finalmente, dos carpetas fueron asignadas a este usuario como propietario, una carpeta para el sistema de producción, y otra carpeta para el sistema de pruebas. Ambas partes se conectaban a MySQL, pero cada una a una base diferente.

**Tabla 4.2***Comandos usados para obtener las características del servidor*

Comando	Uso	Información obtenida
<code>/proc/cpuinfo</code>	Muestra información sobre el procesador y los núcleos de la máquina.	Se obtuvo nombre, velocidad, marca y modelo del procesador, número de núcleos físicos y número de núcleos lógicos.
<code>/etc/passwd</code>	Muestra información de los usuarios registrados en el host.	Se utilizó este comando para comprobar que el usuario se hubiera creado bien, así como para conocer los permisos que poseía.
<code>/etc/group</code>	Muestra información de los grupos registrados en el host.	Este archivo se revisó para comprobar que el grupo creado para los usuarios relacionados con el sistema, estuviera correcto.

Esta fue la parte de configuración de lado del servidor de producción y el de pruebas, a continuación se describe brevemente la parte de configuración del lado de los servidores de desarrollo.

Para la programación, se estuvieron intercalando tres máquinas del departamento como servidores de desarrollo debido a que eran las mismas computadoras usadas por los prestadores de servicio social y había que emplear las máquinas que estuvieran desocupadas. Se usaron también otras dos máquinas propias; una pc de escritorio para el desarrollo desde casa y una Netbook, para tener movilidad y para probar la página desde una resolución menor.

Además, en cada computadora se instalaron hasta un máximo de 5 navegadores (Google Chrome, Mozilla Firefox, Opera, Safari e Internet Explorer; los navegadores de internet más comunes y usados [REF]) en diferentes versiones. Lo anterior implicaba en sí, una muestra general del sistema en varias situaciones con el fin de que éste pudiera verse correctamente sin importar el sistema operativo, los recursos de la máquina o el navegador que se usara para visualizar la página.

A cada una de estas computadoras se le instaló principalmente Notepad++, SSH Secure Shell y Mendeley Desktop (para la parte de la investigación) desde ambientes gráficos por lo que no hubo mayor complicación.

### **IV.III.II. Programación**

Una vez instalado el software necesario para el desarrollo, se hizo una primera página con un menú que estuvo compuesto por las áreas tomadas del Manual de Historia Clínica Académica (como se mencionó en la parte de Diseño); tomando en cuenta estas secciones para proceder a generar cada una de las vistas para el sistema.

Fue así que se creó la primera versión del sistema; un primer sitio funcional, el cual fue revisado por algunos usuarios y se encontró que la interfaz para moverse entre áreas, específicamente el menú, no era adecuado. Este primer sitio incluía también una muestra de cómo sería la interfaz de inicio de sesión para el sistema.

A pesar de que este primer acercamiento no terminó de convencer, sirvió para asegurarse de que las áreas en las cuales estaría dividido el sistema, serían las adecuadas, procurando que el sitio no fuera muy complicado de navegar, ya fuera por tener pocas secciones pero con demasiado contenido y subsecciones o por ser muchas secciones con poco contenido, lo cual resultaría en que el usuario se esforzara inútilmente al querer cambiar de sección.

Hubo una segunda versión, en la cual ya se diseñó, implementó y conectó la base de datos de acuerdo a las áreas que se habían establecido, y se comenzó la etapa de pruebas del sistema.

También en esta fase, se creó una sección específica que funcionaría para administración del sistema, en la cual hay varias opciones como el ver los comentarios recibidos, la posibilidad de enviar correos de aviso a todos los usuarios del sistema y un apartado para poder eliminar usuarios (mismo que sólo se utilizó como parte de las fases de prueba durante el desarrollo).

Finalmente se llegó a la tercera y última versión actual del sistema, la cual se ajustó de acuerdo a la realimentación recibida; se corrigieron algunos fallos graves, faltas de ortografía y bugs. Con esto se dio por terminada la fase de desarrollo y la primera versión oficial del sistema.

### IV.III.III. Navegación del sistema

A continuación en la **Fig. 4.3**, se presenta el diagrama general de navegación de las principales interfaces del sistema y posteriormente, se explica detalladamente cada vista y su funcionamiento.

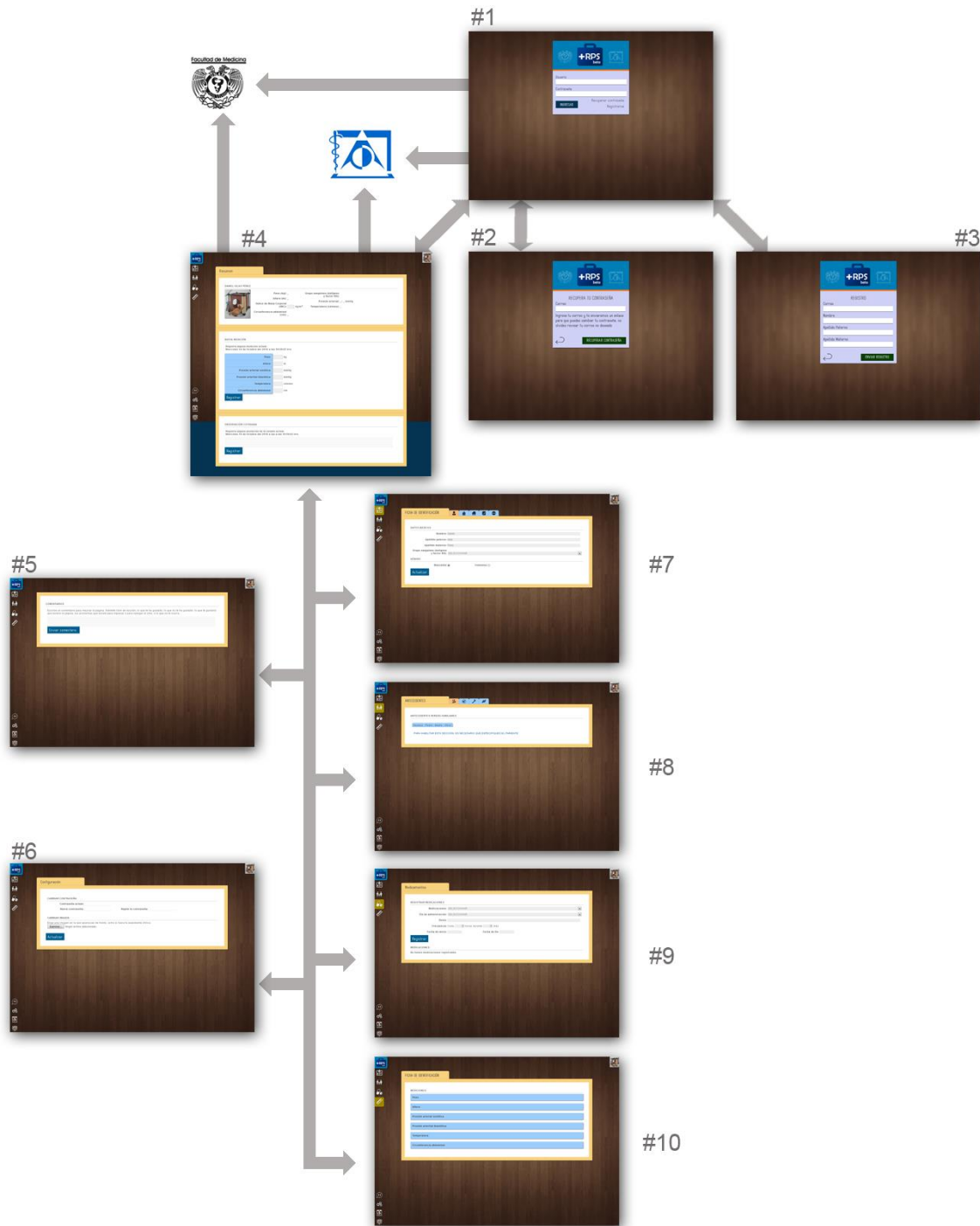

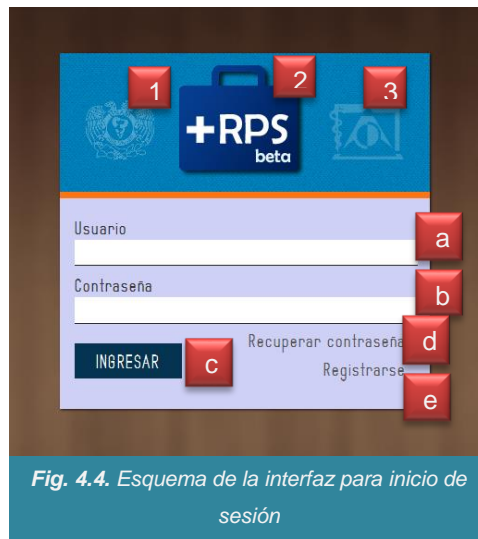


Fig. 4.3. Diagrama de navegación del sistema

Se muestran impresiones de pantalla de las tareas del sistema, además, sobre cada vista o interfaz se ponen indicadores, secuenciados mediante letras como por ejemplo  para señalar ciertos elementos en específico y explicar características del funcionamiento.

### Interfaz #1. Inicio de sesión (login)

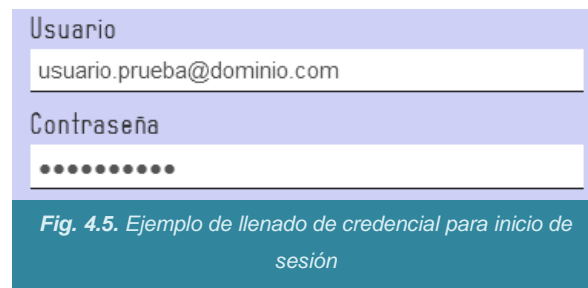


En la **Fig. 4.4** se muestra la primera interfaz que el usuario visualiza. Se observa un contenedor dividido en dos partes: una superior en la cual se encuentran los logos de sitios relacionados al proyecto, y una parte inferior que contiene un formulario para el inicio de sesión.

Los logos de la parte superior corresponden, de izquierda a derecha, al escudo de la Facultad de Medicina de la UNAM (1), el logo del RPS (2) y el logo del departamento de Informática Biomédica (3) de la facultad anteriormente mencionada.

El formulario de inicio de sesión contiene un campo de texto, un campo de contraseña (los caracteres se muestran como círculos no como letras), un botón de envío y dos enlaces.

Los campos son para el nombre de usuario (a) y la contraseña (b) - también referida como clave de acceso - respectivamente; en el primero, el usuario escribe su identificador (actualmente el correo) y en un campo inferior, la contraseña (**Fig. 4.5**).

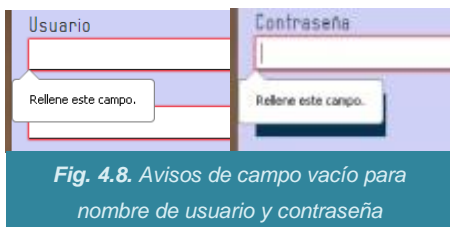




*Fig. 4.6. Aviso de inicio de sesión incorrecto*

en la **Fig. 4.6.**

Para evitar que usuario envíe el formulario sin haberlo completado, tanto el campo de usuario como el de contraseña se han definido como requeridos, por lo que el navegador mostrará alertas de campos vacíos al cambiar de campo (**Fig. 4.7**) y al apretar el botón de envío (**Fig. 4.8**). En caso de que la información esté correcta se procederá a la **Interfaz #4** que se explicará más adelante.



*Fig. 4.8. Avisos de campo vacío para nombre de usuario y contraseña*

Nota: este tipo de validaciones se encuentran a lo largo de todo el sistema, por lo que se omitirá su explicación en futuras ocasiones.

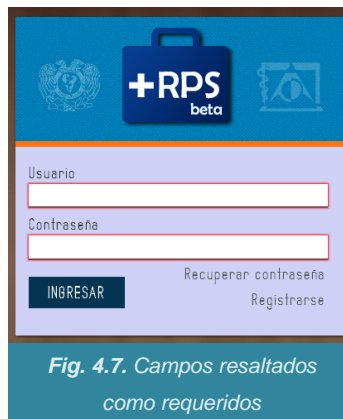
Los enlaces que aparecen en la parte inferior de este contenedor son para recuperar contraseña (d), en caso de usuarios registrados, y para registrarse al sistema (e), en caso de usuarios no registrados.

Al hacer clic sobre el primer enlace, se navega a la **Interfaz #2** y al hacer clic sobre el segundo enlace, se procede a la **Interfaz #3**.

Un botón inferior (c) sirve para enviar la información a otro archivo que comprueba que el nombre exista en la base de datos y de ser así, se comprobará que la contraseña esté asociada con el usuario.

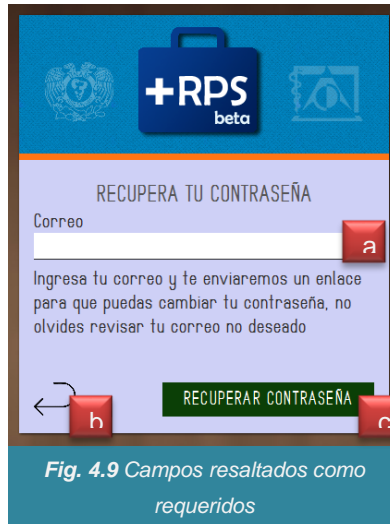
Las contraseñas se encuentran encriptadas con varios algoritmos para evitar el robo y posterior uso inapropiado de éstas.

Si las credenciales son válidas se pasará a la **Interfaz #4**, en caso de haber algún error en la identificación del usuario, se devolverá el flujo a esta interfaz y se mostrará algún aviso como se muestra



*Fig. 4.7. Campos resaltados como requeridos*

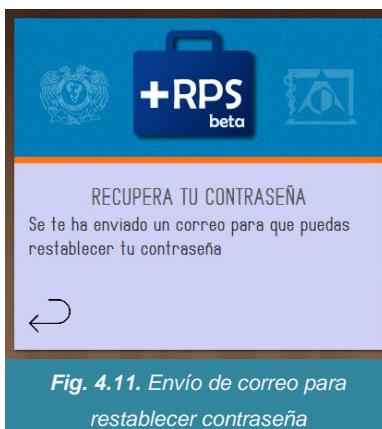
## Interfaz #2. Recuperar contraseña



En esta interfaz (**Fig. 4.9**), la parte superior del contenedor se mantiene igual y la inferior se simplifica por un formulario muy sencillo.

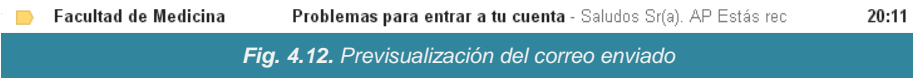
Este nuevo formulario está compuesto por un campo de texto (a), un enlace (b) y un botón de envío (c). El enlace simplemente regresa al usuario a la vista anterior (**Interfaz #1**) mientras que en el campo de texto se escribe el identificador del usuario, y con el botón de se envía la información, hay dos posibles situaciones cuando se envía la información.

Primero, si el identificador de usuario proporcionado no está registrado, se retornará a esta misma vista y se mostrará un aviso indicando que hay un error, (**Fig. 4.10**). El usuario deberá escribir su correo nuevamente, o puede regresar a la página de inicio de sesión con el enlace (b).



Por otra parte, si el identificador ingresado es correcto, es decir existe en la base de datos como usuario registrado, se redirigirá la navegación a una vista similar a la anterior. La diferencia de esta nueva interfaz es que en lugar de aparecer el campo para ingresar el correo, simplemente se mostrará un texto informando al usuario que se ha enviado un enlace a su correo para restablecer la clave de acceso (**Fig.4.11**), el cual contiene las instrucciones necesarias para este proceso.

Este mensaje tiene como remitente “**Facultad de Medicina**”, como asunto “**Problemas para entrar a tu cuenta**” y como contenido, instrucciones sobre cómo restablecer la contraseña (**Fig. 4. 12**).



En este correo, tal como se muestra en la **Fig.4.13**, se le indicará al usuario que hubo una petición de restablecer la contraseña de su cuenta. Se le pedirá al mismo que haga clic sobre un enlace, el cual lo llevará a otra vista (**Fig.4.14**) para configurar su nueva clave de acceso al sistema.

## Saludos Sr(a). AP

Estás recibiendo un correo del Registro Personal de Salud de la Facultad de Medicina porque recientemente se solicitó restablecer la contraseña de tu cuenta. Para continuar con el proceso sigue el enlace que aparece al final del mensaje.

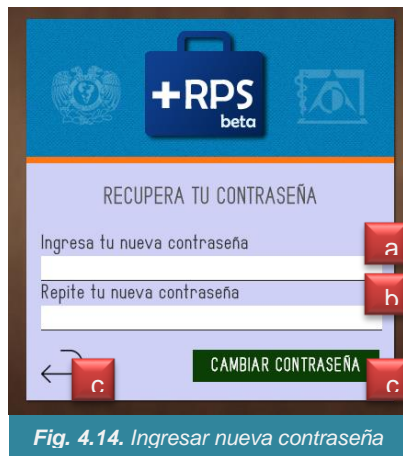
En caso de que no hayas solicitado restablecer tu contraseña no hagas caso de este correo.

Sigue el siguiente enlace para cambiar tu contraseña.  
[http://lab3d.facmed.unam.mx/PHR/CONTROLADORES/recuperar\\_contraseña.php?urs=bc2e26b6ce0817d918cf23eac35d90b51ef249fd&ky=1cf84e2b0e33e1d2ccce1cc1cb25036744585509](http://lab3d.facmed.unam.mx/PHR/CONTROLADORES/recuperar_contraseña.php?urs=bc2e26b6ce0817d918cf23eac35d90b51ef249fd&ky=1cf84e2b0e33e1d2ccce1cc1cb25036744585509)



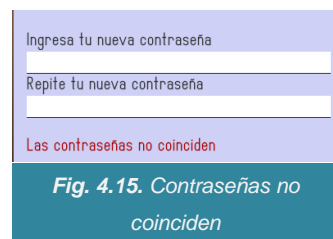
Esta nueva vista está compuesta de dos campos (a y b), un botón de envío (c) y un enlace (d). Tanto el campo superior (a) como el inferior (b) son de tipo contraseña, y se tienen que llenar con la misma información; la nueva clave de

acceso del usuario.



Una vez llenados ambos campos, el usuario deberá usar el botón de envío para mandar a procesar información anterior y, a partir de aquí, pueden ocurrir dos situaciones dependiendo de si los datos fueron correctos o incorrectos.

Si las contraseñas escritas no coinciden, se regresará al usuario a esta misma pantalla, se le mostrará un mensaje de error como en



Por el contrario, si el contenido de ambos campos coincide, se registrará la nueva contraseña y se mostrará al usuario la página de inicio de sesión, desplegando un aviso en la parte inferior, como se muestra en la **Fig.4.16**.





### Interfaz #3. Registro

**Fig. 4.17. Formulario de registro**

En la siguiente vista, **Fig. 4. 17**, aparece el formulario para registro de nuevos usuarios, compuesto por un campo de tipo correo (a), tres campos de tipo texto (b, c y d), un botón de envío (e) y un enlace (f) que regresa al usuario a la **Interfaz #1** (inicio de sesión).

Los tres primeros campos (correo, nombre y apellido paterno) están validados como requeridos y el campo de correo está validado para aceptar solamente cadenas que coincidan con el formato establecido para las direcciones de correo, es decir que coincidan con la expresión regular:

$\wedge[a-zA-Z0-9]+(\.[a-zA-Z0-9])*\@[a-zA-Z0-9][a-zA-Z0-9]*\$$ .

Se ha decidido que el último campo (apellido materno), quede como opcional debido a que no toda la población cuenta con ambos apellidos.

Un vez llenados los campos y enviada la información con el botón proporcionado (e) se enviará al usuario a la **Interfaz #1**

desplegando un mensaje (**Fig. 4.18**) indicándole que ha sido registrado satisfactoriamente y pidiéndole que revise el mensaje enviado a su correo (**Fig. 4.19**) con instrucciones sobre su cuenta.

**Fig. 4.18. Registro correcto**

**Fig. 4.19. Previsualización del mensaje de usuario registrado**

Este correo de bienvenida, tal como se muestra en la **Fig. 4.20**, le muestra al usuario los datos necesarios para iniciar sesión; muestra el nombre de usuario o identificador y una contraseña o clave de acceso provisional, así como el enlace a la página del sistema.

## **Bienvid@ Sr(a). Ap, al Registro Personal de Salud de la Facultad de Medicina**

Sus datos de inicio de sesión son los siguientes:

Usuario: [usuario.prueba@gmail.com](mailto:usuario.prueba@gmail.com)

Contraseña: [LJxufAobSgp](#)

Puedes iniciar sesión desde la siguiente página: <http://lab3d.facmed.unam.mx/PHR>

*Fig. 4.20. Correo de bienvenida*

Usando la URL mostrada en este correo, se puede obtener la **Interfaz #1** en el navegador y posteriormente iniciar sesión con los datos proporcionados, de la manera que se explicó anteriormente en la primera vista.

Si en esta interfaz, el usuario intenta registrarse con un correo que ya existe u ocurre algún error durante el procesamiento del registro, se regresará a la **Interfaz #1** y se mostrará un aviso de error (**Fig. 4.21.**)



The screenshot shows a web interface for 'RPS beta'. At the top, there is a blue header with a white medical cross icon and the text '+RPS beta'. Below the header, there are two input fields: 'Usuario' and 'Contraseña'. To the right of the 'Contraseña' field, there are two links: 'Recuperar contraseña' and 'Registrarse'. Below these fields is a dark blue button labeled 'INGRESAR'. At the bottom of the form, there is an orange banner with the text 'REGISTRO FALLIDO' and 'Intenta nuevamente' below it.

*Fig. 4.21. Registro fallido*

## Interfaz #4. Página principal

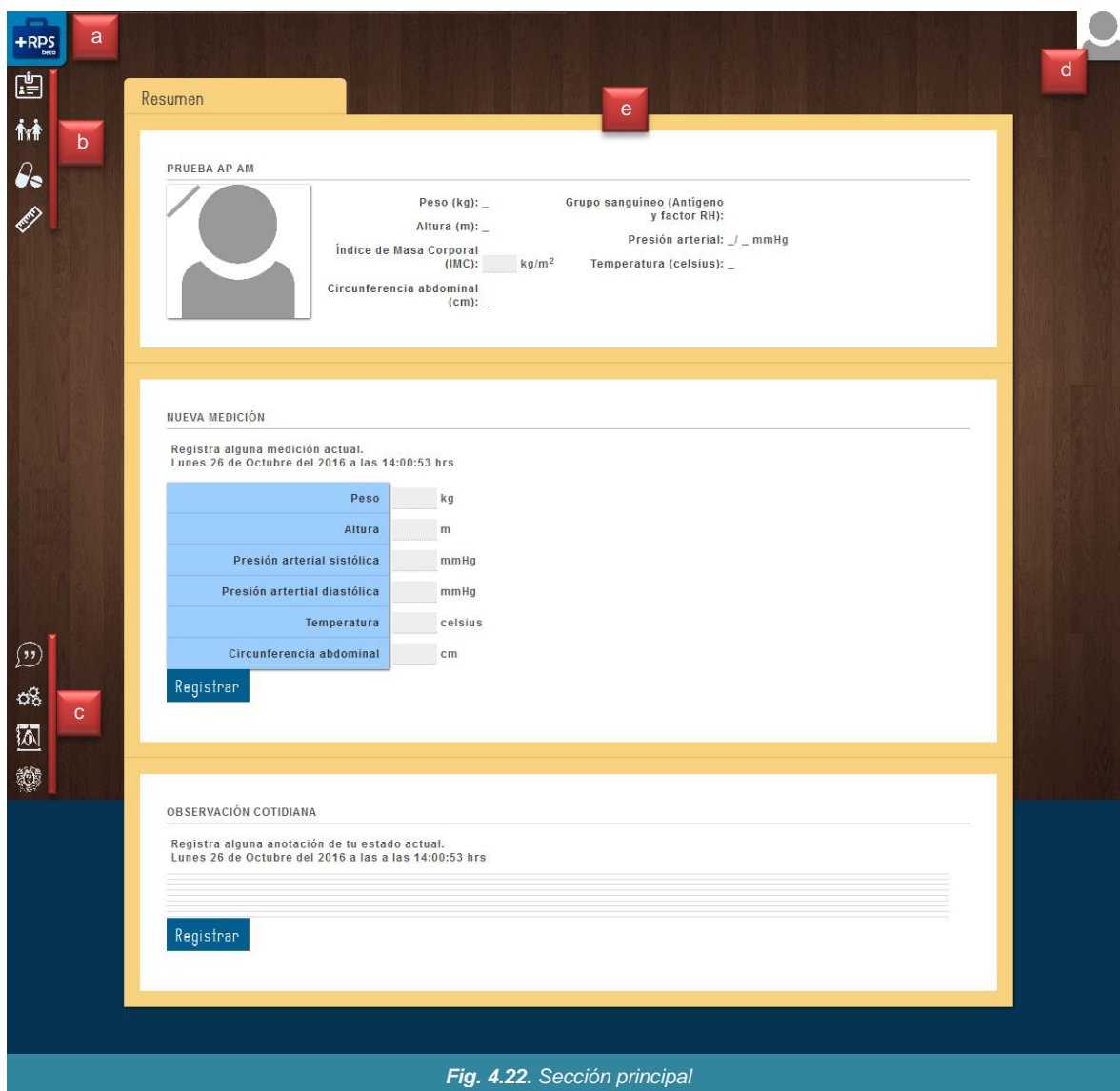


Fig. 4.22. Sección principal

Habiendo ingresado exitosamente al sistema mediante el inicio de sesión, el usuario visualizará la **Interfaz #4**, misma que se muestra en la **Fig. 4.22** y en la cual se pueden apreciar los siguientes elementos: logo del sistema (a), menú superior (b), menú inferior (c), imagen del usuario (d) y contenedor principal (e), se explican a detalle cada una de estas secciones a continuación.

### Logo del sistema (a)

En la esquina superior del sistema está situado un pequeño contenedor en el cual se encuentra el logo del sistema; esta imagen es en realidad un enlace que redirecciona hacia la página principal. Al pasar el cursor encima de esta imagen, se despliega un contenedor con el nombre del sistema, tal como se muestra en la **Fig. 4.23**.





Fig. 4.24.  
Menú superior

### Menú superior (b)

Este menú, situado en el lado izquierdo superior de la página, es el que se utiliza para navegar a través de las principales interfaces del sistema, ya que contiene enlaces hacia las pantallas principales: **Interfaz #7** Ficha de identificación (1), **Interfaz #8** Antecedentes (2), **Interfaz #9** Medicaciones (3), **Interfaz #7** Mediciones (4).

Cada uno de estos enlaces cambia de diseño al pasarse el cursor por encima para resaltar el ícono a seleccionar tal como se indica en la **Fig. 4.24** y se regresa al estilo original al quitar el cursor. También, una vez hecho clic sobre algún enlace del menú y haber pasado al área correspondiente, este estilo se mantiene para resaltar la sección actual.

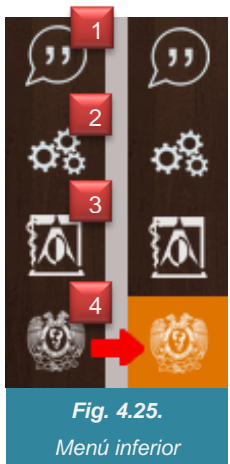


Fig. 4.25.  
Menú inferior

### Menú inferior (c)

Este menú (**Fig. 4.25**) es similar al anterior, tanto en funcionalidad, como en diseño y ubicación (salvo que éste se encuentra en la parte inferior), y contiene enlaces hacia a la **Interfaz #5** Comentarios (1), a la **Interfaz #6** Configuración (2) y a sitios relacionados con el proyecto; a la página web de la Facultad de Medicina de la UNAM (4) y a la página web del departamento de Informática Biomédica de esta facultad (3).

Estos cambios de estilo pueden apreciarse también en la **Fig. 4.3**. Diagrama de navegación del sistema.

Cabe mencionar que se diseñó un estilo web adaptable para el sistema, por lo que en caso de que la resolución sea pequeña (de acuerdo a lo definido en las hojas de estilo), los dos menús mencionados anteriormente, dejarán de estar anclados del lado izquierdo y con formato vertical. Pasarán a tener formato horizontal y a ubicarse en el lado superior (Menú superior) y en el lado inferior (Menú inferior). Un ejemplo de este cambio de visualización puede apreciarse en la **Fig. 4.26**, en la que se compara una vista con el diseño original (arriba) y esa misma vista pero con el diseño compacto, responsivo o adaptable (abajo).

También, a partir de este punto, con el fin de ahorrar espacio en este documento y que el lector pueda observar mejor las imágenes, las vistas presentadas se muestran con el formato compacto debido a que así se hace mejor uso del espacio en el archivo. Además, con este formato se mantiene el contexto de los menús, en lugar de cortar la impresión de pantalla y mostrar solamente el contenedor general, perdiendo así una visión integral del sistema.

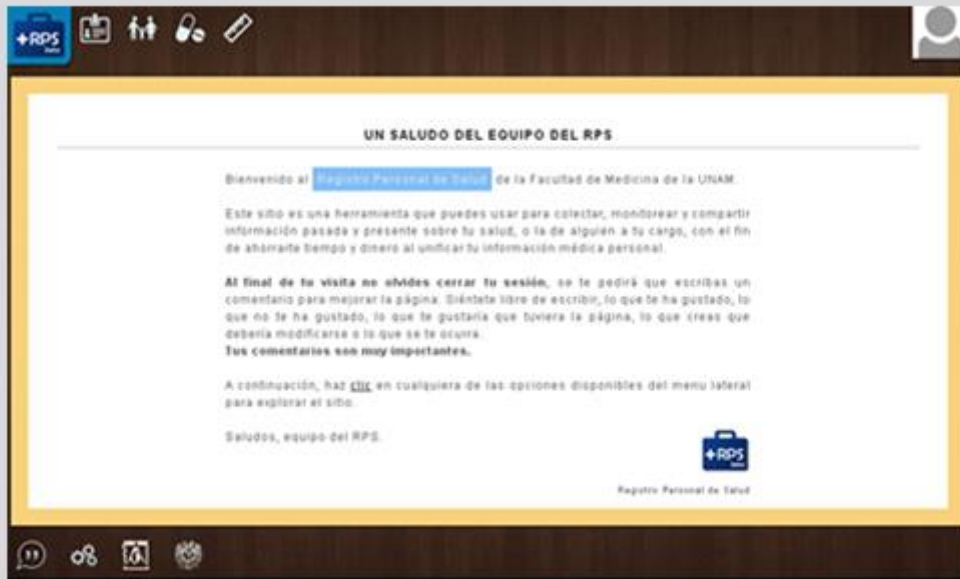


Fig. 4.26. Comparación de vista normal y vista compacta

Fig. 4.27. Estructura del contenedor general I

Resumen

DIEGO SALAS ANDRADE



Peso (kg): 72	Grupo sanguíneo (Antígeno y factor RH): A+
Altura (m): 1.71	Presión arterial: 108/ 62 mmHg
Índice de Masa Corporal (IMC): 24.62 kg/m <sup>2</sup>	Temperatura (celsius): 27
Circunferencia abdominal (cm): 89	

La siguiente sección sirve para insertar nuevas mediciones ya que contiene los tipos de medidas que se pueden registrar en el sistema (esta parte es escalable) y al lado de cada tipo de medida hay un campo de texto para poder ingresar el valor obtenido del examen físico correspondiente.

También, si se ha ingresado el peso y la altura, se calcula el Índice de Masa Corporal (IMC) y se despliega con un color verde (**Fig. 4.27**), naranja o rojo, dependiendo de qué tan saludable es el valor. Finalmente en la parte inferior de estas medidas, hay un botón para enviar a procesar los datos (**Fig. 4.28**).

NUEVA MEDICIÓN

Registra alguna medición actual.  
Lunes 10 de Noviembre del 2016 a las 17:43:09 hrs

Peso	<input type="text"/>	kg
Altura	<input type="text"/>	m
Presión arterial sistólica	<input type="text"/>	mmHg
Presión arterial diastólica	<input type="text"/>	mmHg
Temperatura	<input type="text"/>	celsius
Circunferencia abdominal	<input type="text"/>	cm

Registrar

Fig. 4.28. Estructura del contenedor general II

Nota: las unidades de medida para cada medición están definidas de acuerdo a los tipos de medición registrados en la base de datos.

La siguiente sección, mostrada en la **Fig. 4.29**, es utilizada para insertar observaciones cotidianas; son campos de texto abiertos en los cuales, ya sea por comodidad o por haber un espacio específicamente dedicado en el sistema, el usuario puede registrar cualquier tipo de anotación que tenga en mente. Tiene un respectivo botón para el envío de información.

**Fig. 4.29.** Estructura del contenedor general III

OBSERVACIÓN COTIDIANA

Registra alguna anotación de tu estado actual.  
Lunes 10 de Noviembre del 2016 a las 17:43:09 hrs

Registrar

Debido a que generalmente este tipo de anotaciones, las observaciones cotidianas, son más prácticas para el usuario y/o contienen información muy importante y relevante para el médico, el listado de éstas aparece en el siguiente contenedor (en realidad un conjunto de contenedores) y se van apilando una tras otra, de manera que las últimas anotaciones realizadas, serán las primeras en aparecer. Se muestran de ejemplo un par de anotaciones en la **Fig. 4.30**.

**Fig. 4.30.** Estructura del contenedor general IV

2016-04-02 10:11:01

Circunferencia de biceps midió 27.5 cm

2016-03-26 13:44:27

Circunferencia de biceps midió 27 cm

Nota: esta interfaz será la que visualizará el usuario cada vez que inicie sesión, excepto la primera vez que ingrese en el sistema, en cuyo caso aparecerá en el contenedor general, un aviso de bienvenida y una breve instrucción sobre cómo usar el sistema, como se muestra en la **Fig 4.31**.



Las siguientes dos secciones (**Interfaz #7**. Ficha de identificación e **Interfaz #8**. Antecedentes) son principalmente las que se diseñaron a partir del Manual de Historia Clínica Académica y cuya estructura y diseño se definió anteriormente en **Estructura del sistema** en la fase de elaboración, por lo que simplemente se mostrarán las vistas de cómo quedó finalmente esta parte del sistema.

Los formularios no necesitan mayor explicación, pues están compuestos en su mayoría por campos de texto libres y botones de envío, los cuales se han explicado con anterioridad. Siendo así, se obvian las explicaciones sobre funcionamiento de estas dos vistas (**Interfaz #7** e **Interfaz #8**), no así de las últimas dos pantallas (**Interfaz #9** e **Interfaz #10**) principales.



## Interfaz #7. Ficha de identificación

FICHA DE IDENTIFICACIÓN

DATOS BÁSICOS

Nombre: Prueba

Apellido paterno: Ap

Apellido materno: Am

Grupo sanguíneo (Antígeno y factor RH): SELECCIONAR

GÉNERO

Masculino  Femenino

Actualizar

Fig. 4.32. Ficha de identificación I (Datos básicos)

En la **Fig. 4.32** se muestra la primera parte de esta sección; cabe resaltar que para esta vista y las siguientes, cuando el usuario no ha registrado información en los campos éstos estarán rodeados de una banda roja para indicarle que necesitan ser llenados. Lo anterior se puede apreciar en esta última imagen - en el área de género - pues al ser la primera vez que se visualiza la información, no se ha especificado un valor y está remarcado de color.

Después de enviar la información de un formulario el sistema la procesa e imprime un mensaje en la misma sección, junto al botón de envío, informando al usuario que los datos se actualizaron de manera correcta (**Fig. 4.33**) o, en caso de que llegue a pasar, que hubo algún error al actualizar la información, (**Fig. 4.34**).

Fig. 4.33. Datos actualizados correctamente

Masculino  Femenino

Actualizar Los datos se actualizaron correctamente

Fig. 4.34. Error al actualizar datos

Masculino  Femenino

Actualizar Ha ocurrido algún error, vuelva a intentarlo

A continuación se visualizan las demás secciones (**Fig. 4.35** Fecha y lugar de nacimiento, **Fig. 4.36** Domicilio, **Fig. 4.37** Contacto y **Fig. 4.38** Otros datos).

Fig. 4.35. Ficha de identificación II (Fecha y lugar de nacimiento)

The screenshot shows a web interface for a 'Ficha de identificación II' (Identification Card II) focused on birth information. At the top, there is a navigation bar with a '+RPS beta' logo and several icons representing different services. Below this is a header with the title 'FICHA DE IDENTIFICACIÓN' and a set of navigation icons. The main content area is titled 'FECHA Y LUGAR DE NACIMIENTO' and contains the following fields:

- Fecha de nacimiento: Día 0, Mes 0, Año 0 (with dropdown arrows)
- Pais de nacimiento: SELECCIONAR (dropdown menu)
- Entidad federativa de nacimiento: (text input field)
- Ciudad de nacimiento: (text input field)
- Delegación o municipio de nacimiento: (text input field)

An 'Actualizar' (Update) button is located at the bottom left of the form area. The interface also includes a user profile icon in the top right and a footer with communication and settings icons.

The screenshot shows a web interface for a 'Ficha de identificación III' (Identification Card III) focused on address information. It features the same top navigation bar and header as the previous form. The main content area is titled 'DOMICILIO:' and contains the following fields:

- Pais: SELECCIONAR (dropdown menu)
- Entidad federativa: (text input field)
- Ciudad: (text input field)
- Delegación o municipio: (text input field)
- Colonia: (text input field)
- Calle: (text input field)
- #: (text input field)
- Código postal: (text input field)

An 'Actualizar' (Update) button is located at the bottom left of the form area. The interface also includes a user profile icon in the top right and a footer with communication and settings icons.

Fig. 4.36. Ficha de identificación III (Domicilio)

**FICHA DE IDENTIFICACIÓN**

**CONTACTO:**

Teléfono:

Teléfono móvil:

Correo electrónico: correo.prueba@gmail.com

Actualizar

Fig. 4.37. Ficha de identificación IV (Contacto)

**FICHA DE IDENTIFICACIÓN**

**OTROS DATOS**

Estado civil SELECCIONAR

Escolaridad SELECCIONAR

Religión SELECCIONAR

Nacionalidad

Ocupación SELECCIONAR

Actualizar

Fig. 4.38. Ficha de identificación V (Otros datos)

## Interfaz #8. Antecedentes

The screenshot shows a web interface for recording medical history. At the top, there is a navigation bar with icons for RPS, a clipboard, a family, a pill, and a document. Below this is a header with the word 'ANTECEDENTES' and several icons. The main content area is titled 'ANTECEDENTES HEREDO-FAMILIARES' and contains a sub-header with buttons for 'Abuelos', 'Padre', 'Madre', and 'Otros'. Below these buttons is a message: 'PARA HABILITAR ESTA SECCIÓN, ES NECESARIO QUE ESPECIFIQUES EL PARIENTE'. At the bottom of the interface, there are icons for chat, settings, a person, and a group.

Fig. 4.39. Antecedentes heredo familiares I

La siguiente sección es el área de antecedentes (**Fig. 4.39**) y, de manera similar a la ficha de identificación, ya se definió la estructura anteriormente por lo que se omite la explicación de los campos. Sin embargo, es necesario hacer algunas anotaciones sobre el funcionamiento del registro de antecedentes heredo familiares, ya que éste se realiza por partes.

Para este proceso, y tal como se indica en la propia vista, primero se selecciona el tipo de pariente (abuelo, padre, madre u otro) al que haremos referencia. De aquí, si se selecciona abuelos, tendremos que elegir si son paternos o maternos, y finalmente nos aparecerán las opciones posibles que se pueden registrar en el sistema (**Fig. 4.40**). Estas últimas opciones aparecerán de manera inmediata si, en vez de abuelos, se selecciona padre o madre, como se muestra en la **Fig. 4.41**.

This screenshot shows the detailed selection options for recording family history. It features a grid of buttons for selecting the type of relative: 'Abuelos', 'Padre', 'Madre', and 'Otros'. Below these, there are buttons for 'Paternos' and 'Maternos'. The main area contains a list of medical conditions with checkboxes: 'E. infectocontagiosas', 'Hipertensión arterial', 'E. tiroideas', 'Cardiopatías', 'Diabetes mellitus', 'Nefropatías', and 'Neurológicas mentales'. At the bottom, there is a section for 'Otras enfermedades' with several horizontal lines for text entry, and an 'Actualizar' button.

Fig. 4.40. Antecedentes heredo familiares II

**Fig. 4.41.** Antecedentes heredo familiares III

Finalmente, en caso de que se opte por otra opción de pariente, primero se seleccionará el tipo de relación, desde una lista desplegable llena con un catálogo de la base de datos (**Fig. 4.42**). Una vez seleccionado el parentesco, se muestran las opciones de registro disponibles para las patologías como en los casos anteriores.

**Fig. 4.42.** Antecedentes heredo familiares IV

A continuación se visualizan las demás secciones (**Fig. 4.43** Antecedentes personales patológicos, **Fig. 4.44** Antecedentes personales no patológicos, **Fig. 4.45** Antecedentes referentes al género, **Fig. 4.46** Antecedentes andrológicos y **Fig. 4.47** Antecedentes gineco-obstétricos).

Fig. 4.43. Antecedentes personales patológicos

The image shows a screenshot of a medical software interface. At the top, there is a dark blue header with the text "Fig. 4.43. Antecedentes personales patológicos". Below the header is a navigation bar with several icons: a blue box with "+RPS" and "RPS", a document icon, a family icon, a pill icon, and a "RPS" icon. To the right of the navigation bar is a user profile icon. Below the navigation bar is a yellow tab labeled "ANTECEDENTES". To the right of the tab are four icons: a group of people, a prohibition sign, a key, and a lock. The main content area is a white form with a yellow border. The form is titled "ANTECEDENTES PERSONALES PATOLÓGICOS" and contains several sections, each with a title and a series of horizontal lines for text entry: "Enfermedades de la infancia", "Alergias", "Transfusionales", "Quirurgicos", "Hospitalizaciones", "Traumatismos y fracturas", "Consumo de tabaco", "Consumo de alcohol", and "Consumo de drogas". At the bottom left of the form is a blue button labeled "Actualizar". At the bottom of the interface is a dark blue footer with several icons: a speech bubble, a gear, a person, and a crown.

Fig. 4.44. Antecedentes personales no patológicos

The image shows a digital medical history form. At the top, there is a dark blue header with the text 'Fig. 4.44. Antecedentes personales no patológicos'. Below the header is a navigation bar with several icons: a first aid kit, a document, a family, a pill, and a tooth. The main content area is titled 'ANTECEDENTES PERSONALES NO PATOLÓGICOS' and contains several sections with text prompts and horizontal lines for input:

- Alimentación:** cantidad y frecuencia en el consumo de alimentos por semana: leche, carne, huevo, verduras, frutas, cereales, leguminosas, etcétera. Balanceada en calidad y cantidad realizando 3 comidas al día, consumo de 1L de agua al día.
- Habitación:** habita en casa propia/rentada, tipo de vivienda( jacal, departamento, vecindad, casa sola, tipo urbana que cuenta con todos los servicios intra y extradomiciliarios, construida de materiales perdurables ), distribución de la vivienda( número de cuartos y servicios, número de personas por habitación, convivencia con animales, tipo y número ), higiene de la vivienda( iluminación, adecuada ventilación )
- Hábitos higiénicos individuales:** aseo personal, baño, cambio de ropa, lavado de manos, aseo dental.
- Ocupación actual y previa:** fecha y duración; condiciones del trabajo, horas que labora, higiene laboral, exposición a factores de riesgo laboral.
- Uso de tiempo libre:** horario de descanso y recreación, deportes y pasatiempos, vacaciones
- Inmunizaciones:** vacunas y número de dosis (Sabin, DPT, pentavalente, BCG, etcétera), Biológicos (suero antirrábico, antialacrán, anticrotático, gammaglobulina, anti-Rh).
- Actividad física**
- Viajes recientes**
- Tatuajes**
- Perforaciones**
- Exposicion tóxica**

At the bottom left of the form is a blue button labeled 'Actualizar'. The bottom of the screen features a dark blue footer with several icons: a speech bubble, a gear, a person, and a crown.



Fig. 4.45. Antecedentes referentes al género

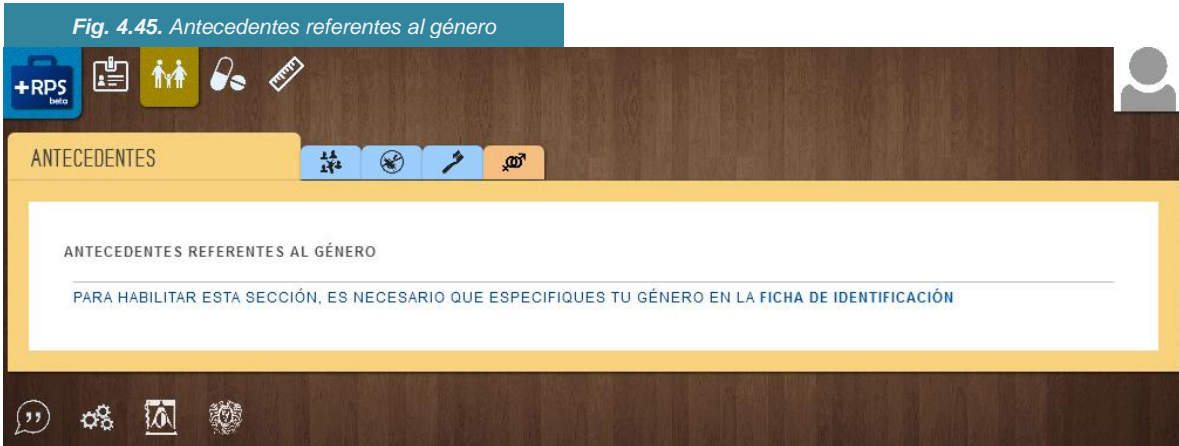
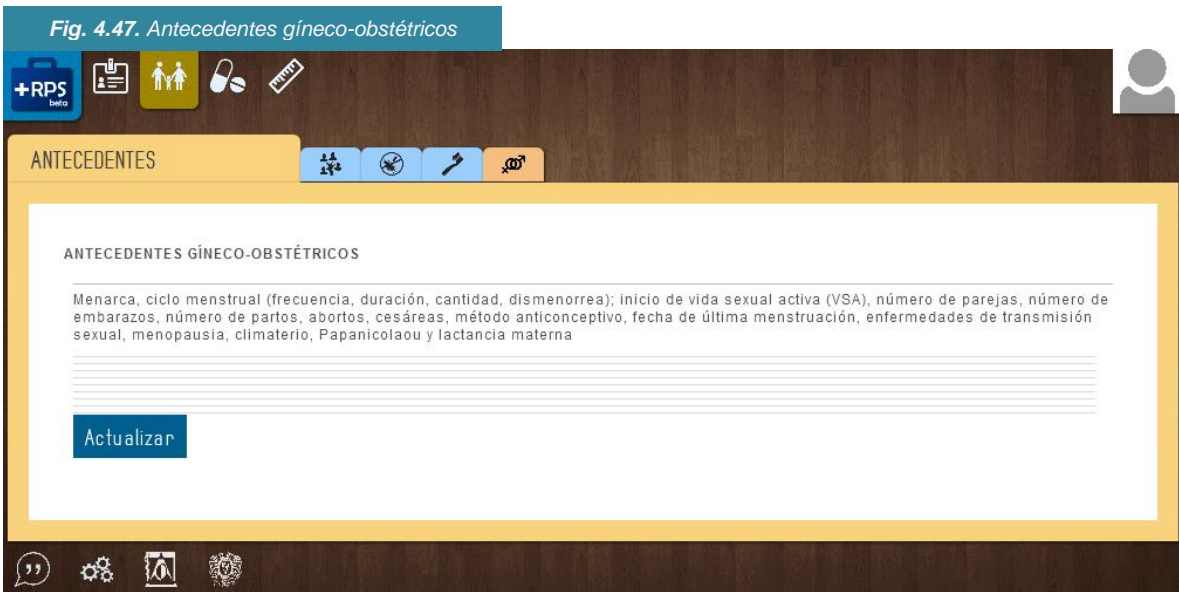


Fig. 4.46. Antecedentes andrológicos



Fig. 4.47. Antecedentes gineco-obstétricos





## Interfaz #9. Medicaciones

**Fig. 4.48. Medicaciones**

Medicamentos

REGISTRAR MEDICACIONES

Medicamento: SELECCIONAR **a**

Via de administración: SELECCIONAR **b**

Dosis: **c**

Frecuencia: Cada **c** horas durante **c** días **c**

Fecha de inicio: 12/10/2016 **d** Fecha de término: **e**

Registrar **f**

MEDICACIONES:

No tienes medicaciones registradas

La sección ilustrada en la **Fig. 4.48** muestra la interfaz para registrar medicaciones, está formada por: dos listas desplegadas, una para seleccionar el medicamento (a), y otra para seleccionar la vía de administración (b); un campo de texto para escribir la dosis (c), dos campos de tipo número para seleccionar la frecuencia dos campos de tipo fecha que sirven para definir la fecha de inicio (d) y la fecha de fin de la medicación (e); y un botón de envío (f).

Para registrar la medicación se seleccionan las opciones deseadas de los primeros dos campos, los cuales son listas que se rellenan con catálogos obtenidos de la base de datos, se escribe la dosis anotada por el médico, se escribe cada cuándo debe tomarse el medicamento y por cuánto tiempo, y finalmente se envía la información.

Cabe mencionar que el campo de fecha de inicio está puesto por defecto en la fecha actual, y al cambiar el número de días en que se vaya tomar el medicamento, se actualiza automáticamente la fecha de término tomando en cuenta este factor. Además, las fechas de inicio y de fin despliegan un calendario respectivamente para facilitar la selección de fechas.

Una vez registradas las medicaciones, éstas van apareciendo debajo del formulario, tal como se ejemplifica en las **Fig. 4.49** (ejemplo de llenado del formulario) y **Fig. 4.50** (como se muestra la medicación una vez registrada).

**Fig. 4.49. Ejemplo de llenado de medicaciones**

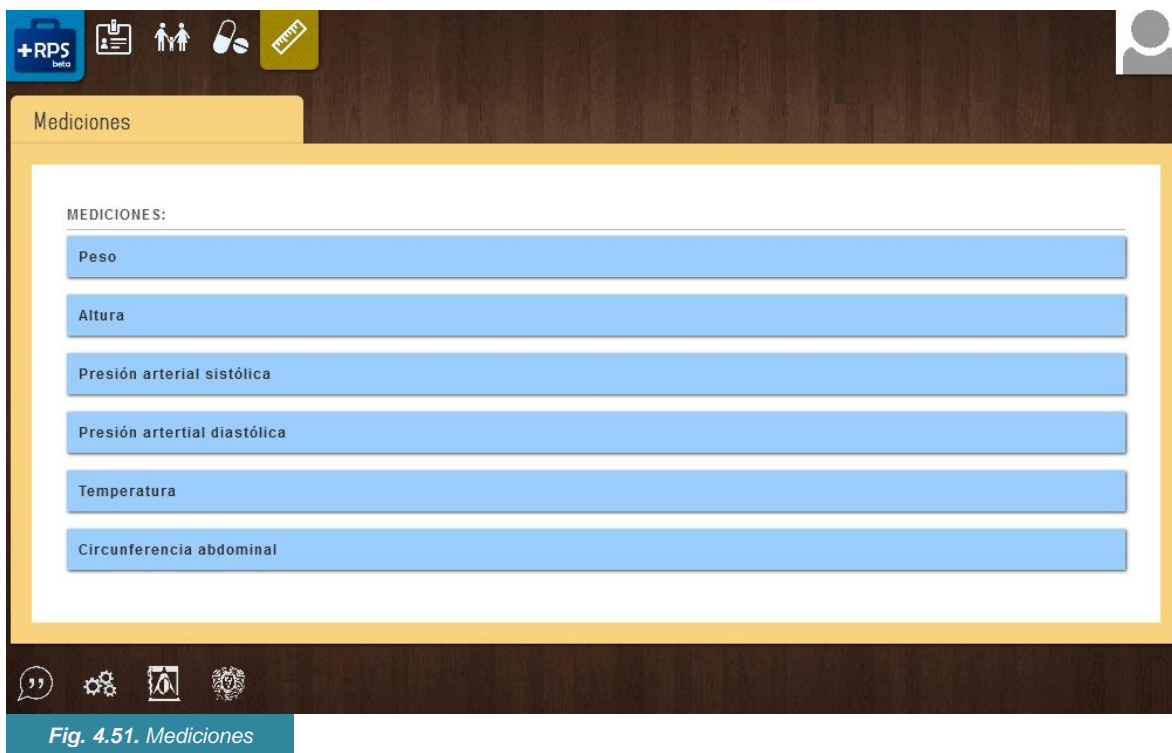
The screenshot shows a web interface for registering medications. At the top left, there is a navigation bar with icons for '+RPS', a document, a family, a pill, and a 'NUEVO' button. A user profile icon is in the top right. Below the navigation bar is a yellow header with the text 'Medicamentos'. The main content area is a white box with a yellow border. It contains a form titled 'REGISTRAR MEDICACIONES' with the following fields: 'Medicamento:' with a dropdown menu showing 'Diltiazem'; 'Via de administración:' with a dropdown menu showing 'Pastillas'; 'Dosis:' with a text input field containing '180mg'; 'Frecuencia:' with a dropdown menu showing 'Cada 24' and a text input field containing 'horas durante 30' and a dropdown menu showing 'días'; 'Fecha de inicio:' with a date picker showing '12/10/2016'; and 'Fecha de término:' with a date picker showing '11/11/2016'. Below the form is a blue 'Registrar' button. Underneath the form is a section titled 'MEDICACIONES:' with the text 'No tienes medicaciones registradas'. At the bottom of the interface are icons for a speech bubble, a gear, a person, and a flower.

**Fig. 4.50. Muestra de medicaciones registradas**

The screenshot shows the same web interface as Fig. 4.49, but with the registration process completed. The 'Medicamento:' dropdown now shows 'SELECCIONAR', and the 'Via de administración:' dropdown also shows 'SELECCIONAR'. The 'Dosis:' field is empty. The 'Frecuencia:' dropdown shows 'Cada' and the text input field shows 'horas durante' and the dropdown shows 'días'. The 'Fecha de inicio:' date picker shows '12/10/2016' and the 'Fecha de término:' date picker is empty. The blue 'Registrar' button is now highlighted in green, and a green message box below it says 'Los datos se actualizaron correctamente'. Below the form is a section titled 'MEDICACIONES:' with a list of registered medications. The first entry is 'Diltiazem', 180mg suministrado mediante Pastillas en intervalos de 24 horas cada 30 días. Comenzando desde el día 2016-12-10 hasta el día 2016-11-11. At the bottom of the interface are icons for a speech bubble, a gear, a person, and a flower.

## Interfaz #10. Mediciones

En la última de las secciones principales (**Fig. 4.51**) se despliegan los valores que haya registrado el usuario a partir de exámenes físicos. Estas mediciones se irán acumulando y ordenando de más reciente a más antiguo, tal como se muestra en la **Fig. 4.52**.



Se recuerda que los últimos valores que se tengan registrados, son los que aparecen en el resumen de la sección principal del sistema.

Hasta aquí terminan las principales y más importantes interfaces del sistema, faltan por mencionar brevemente, las interfaces de comentarios y configuración de la cuenta. También, por seguridad, se hace omisión de las interfaces de administrador generadas.

Fig. 4.52. Listado de mediciones

MEDICIONES:

Peso	
2015-04-13 12:38:38	66.4 (kg)
2015-04-22 12:07:17	66.2 (kg)
2015-04-26 14:34:23	64.6 (kg)
2015-06-28 15:38:11	65.9 (kg)
2015-11-20 22:10:45	68.9 (kg)
2015-12-03 21:27:31	68.3 (kg)
2015-12-05 11:59:24	67.4 (kg)
2015-12-09 21:00:05	66.3 (kg)
2015-12-12 11:48:13	68.9 (kg)
2016-01-07 21:22:37	69.9 (kg)
2016-07-28 00:04:05	72 (kg)

Altura	
2015-04-13 12:38:38	1.71 (m)

Presión arterial sistólica	
2015-04-23 10:22:31	80 (mmHg)
2016-02-20 16:43:44	108 (mmHg)

Presión arterial diastólica	
2015-04-23 10:22:31	120 (mmHg)
2016-02-20 16:43:44	62 (mmHg)




  

Temperatura	
2015-04-23 10:22:53	27 (celsius)

Circunferencia abdominal	
2016-01-14 18:25:05	89 (cm)

## Interfaz #5. Comentarios

COMENTARIOS

Escribe un comentario para mejorar la página. Siéntete libre de escribir, lo que te ha gustado, lo que no te ha gustado, lo que te gustaría que tuviera la página, los problemas que tuviste para ingresar o para navegar el sitio, o lo que se te ocurra.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Enviar comentario**


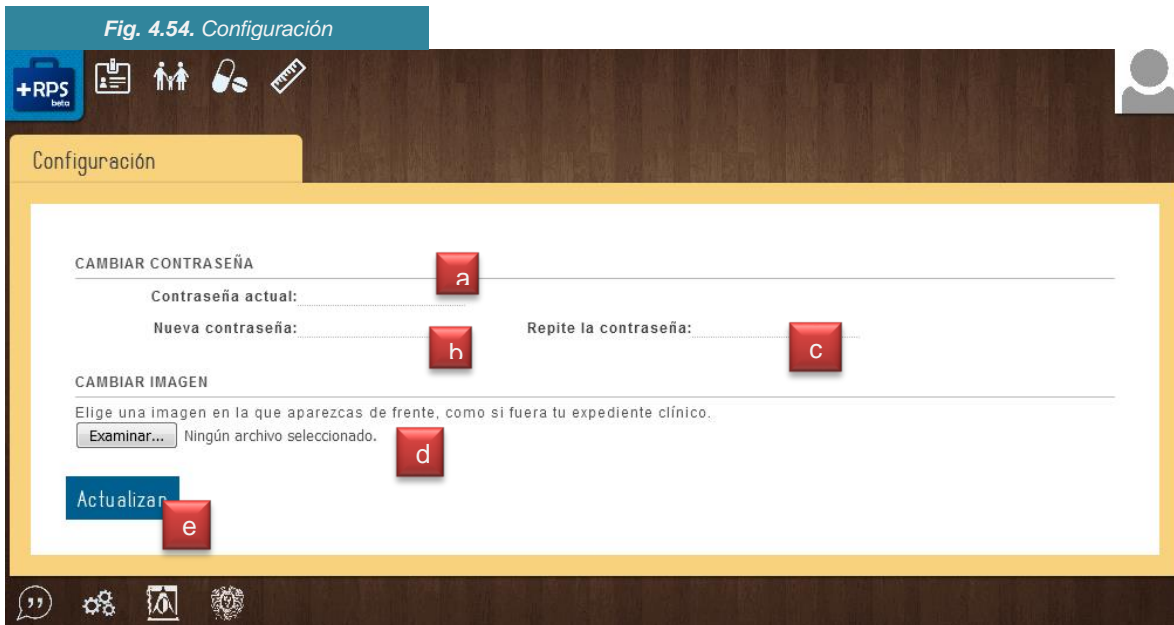


Fig. 4.53. Comentarios

Esta vista (**Fig. 4.53**) se trata simplemente de un pequeño formulario en el cual, el usuario puede escribir texto de formato libre y enviarlo a los administradores del sistema, con el fin de realimentar y mejorar así el proyecto. Estos comentarios son recibidos y administrados a través del sistema, por el usuario administrador.

## Interfaz #6. Configuración



Es la última pantalla que se presenta en el documento; sirve para actualizar la contraseña y la imagen del usuario (**Fig. 4.54**).

Para la primera parte (actualizar contraseña) se cuenta con tres campos de tipo contraseña (a, b y c); en el primer campo se tiene que escribir la contraseña actual, y en los siguientes dos (b y c), se escribe la nueva contraseña. En caso de que haya algún error al procesar la información (la nueva clave no coincida en ambos campos o la clave actual ingresada sea incorrecta), se mostrará un aviso como en la **Fig. 5.55**.

**Fig. 4.55. Error al actualizar contraseña**

The screenshot shows a web interface for changing a password. At the top, there is a section titled "CAMBIAR CONTRASEÑA" with a horizontal line below it. Below this line are three input fields: "Contraseña actual:", "Nueva contraseña:", and "Repite la contraseña:". A red error message box is displayed below the input fields, containing the text "Ha habido algún error al actualizar la contraseña". Below the error message is another section titled "CAMBIAR IMAGEN" with a horizontal line below it. Below this line is the text "Elige una imagen en la que aparezcas de frente, como si fuera tu expediente clínico." followed by a button labeled "Examinar..." and the text "Ningún archivo seleccionado."

En la segunda parte (actualizar imagen), el usuario deberá seleccionar una imagen con el botón proporcionado (d) y enviar la información. De igual manera, si ocurre algún problema, se notificará con un aviso (**Fig. 4.56**). Para ambas partes se usa el mismo botón de envío (e).

**Fig. 4.56. Error al actualizar imagen**

The screenshot shows a web interface for uploading an image. At the top, there is a section titled "CAMBIAR IMAGEN" with a horizontal line below it. Below this line is the text "Elige una imagen en la que aparezcas de frente, como si fuera tu expediente clínico." followed by a button labeled "Examinar..." and the text "Ningún archivo seleccionado." A red error message box is displayed below the input field, containing the text "Ha habido algún error al actualizar la imagen". Below the error message is a blue button labeled "Actualizar".

### Notas finales de las interfaces

Las validaciones mostradas en las vistas son del lado del cliente, sin embargo, también se tiene las mismas del lado del servidor, con el fin de tener mayor seguridad.

## V. RESULTADOS

El producto final del proyecto fue un sistema en línea a través del cual, personas sin conocimientos previos de medicina, pueden registrar datos relacionados con su salud o la de algún pariente para poder consultarla posteriormente, con el fin de tener un registro organizado por consiguiente, una presencia más activa en el cuidado de su salud. Este proyecto es accesible actualmente desde el servidor del departamento de Informática Biomédica mediante la siguiente dirección:

<http://lab3d.facmed.unam.mx/PHR>

Nota: la navegación y estructura finales del sistema ya se comentaron ampliamente en la parte de desarrollo, de manera que no se volverán a presentar.

El conteo final de usuarios registrados fue de 649 personas, y los tres meses con mayor tráfico registrado fueron, en orden descendente, febrero, abril y marzo.

Además, se recibieron 263 comentarios para el sistema (38.82% de los usuarios totales), a partir de los cuales se pudieron conocer ciertos datos sobre la experiencia de uso, principalmente sobre si gustaba o no el sistema (Tabla 5.1). Se agruparon de la siguiente manera:

- **Agradable:** Los usuarios que expresaron específicamente que el diseño era apropiado, cómodo y/o le gustaba por alguna razón.
- **No dice:** Los usuarios que no comentaron específicamente si les gustaba o disgustaba el sistema.
- **Desagradable:** Los usuarios que pensaban que el sistema era inadecuado, tedioso y/o no les gustaba por alguna razón.
- **No interesa:** Los usuarios que ingresaron comentarios sin sentido (espacios en blanco, letras sin coherencia, etc.)

**Tabla 5.1**

*Características a resaltar de los comentarios respecto al diseño y/o estructura general del sistema*

Los usuarios en general piensan que el sistema es ...

Agradable	Desagradable	Sin percepción explícita	No interesa	TOTAL
169 (64.27%)	2 (0.76%)	68 (25.85%)	24 (9.12%)	263

Notas:

También se recopilaron algunas características o cambios que mencionaban los usuarios y que pudieran implementarse en futuras ocasiones **Tabla 5.2**. Nótese que por facilidad, se muestra el recuento total de las personas que presentaron ese tipo de comentarios agrupados en 3 categorías, las cuales se presentan a continuación.

- **Ajuste de diseño y/o estructura:** Se sugirieron cambios, ajustes o adiciones al sistema; desde errores ortográficos, tamaño de letra y esquema de colores, hasta nuevas áreas completas para el sistema o plataformas de uso (por ejemplo una aplicación móvil).
- **Útil para la práctica médica:** Mencionaron explícitamente que podría serles útil o que veían el propósito del proyecto.
- **Problemas con el registro de información o con el acceso al sistema:** Mencionaron que hubo problemas al momento de registrar datos en el sistema o cuando quisieron acceder al sistema por primera vez.

**Tabla 5.2**

*Características a resaltar y más comunes de los comentarios del sistema*

Algunos usuarios piensan sobre el sistema que...

Ajuste o cambio del diseño y/o estructura	Útil para la práctica médica	Problemas en registro de información o con acceso al sistema
46	58	34

Notas: La suma total de estos comentarios no abarca el 100% de los que se recibieron, además, algunos comentarios aparecen agrupados bajo dos o incluso las 3 categorías.

A partir de lo anterior, se pudo comprobar que en general, el sistema era bastante agradable (adecuado como herramienta, cómodo de usar y con buena presentación) para los usuarios y que se creía era una aplicación que podría llegar a ser de utilidad médica.

Se contestaron los comentarios de los usuarios que necesitaban más ayuda (no podían o no sabían cómo entrar al sistema, tenían errores al ingresar información, etc.) y se les dio seguimiento. A los usuarios restantes, se les respondió a sus comentarios con un correo de agradecimiento por haber participado en la prueba del sistema.

También, de las pruebas realizadas (Apache Bench) se comprobó que el sistema es capaz de manejar sin problema 25 usuarios y hasta un promedio de 40 usuarios concurrentes bajo estrés, con un tiempo máximo de respuesta de 30 segundos.

Sin embargo, dado que la arquitectura de esta aplicación se divide en tres capas (el servidor web, el servidor de aplicación y el servidor de la base de datos), el máximo de usuarios concurrentes



depende en gran medida de cómo se optimicen las tres capas a la vez. Lo anterior posibilita en un futuro la optimización del tiempo de respuesta del sistema.

Finalmente, se presenta el porcentaje de uso de cada una de las áreas de registro de información del sistema en la **Tabla 5.3**.

**Tabla 5.3**

*Porcentaje de uso de cada área*

Porcentaje de usuarios que registraron información al menos una vez en cada área

Ficha de ID	Mediciones	Antecedentes	Medicaciones
41%	27%	14%	2%

Notas: Estos porcentajes solamente reflejan cuántos usuarios registraron información al menos una vez; sin embargo no toma en cuenta las visualizaciones de la información registrada, ni el grado de uso de las secciones por cada usuario.

Como último resultado, hay que mencionar que en el desarrollo del proyecto, se logró hacer un diagnosticador de enfermedades y fue probado con una base de datos, tomando como referencia varias guías de práctica clínica del gobierno.

Sin embargo, debido a la complejidad del sistema, se creyó conveniente dejarlo para un trabajo aparte como un modelo de espacio vectorial para la toma de decisiones en el diagnóstico clínico, mismo que se planea seguir desarrollando y mejorando.

## VI. CONCLUSIONES

La información personal debería estar disponible en todo momento, sobre todo si se trata de datos médicos o relacionados con la salud; por esta razón debe impulsarse una campaña en México para que la sociedad se adueñe de su propia información, de manera que no dependa totalmente de instituciones de salud o de un médico en particular para manejar sus datos.

Un mal empleo y manejo de los datos puede implicar que la información del paciente se monopolice por parte de instituciones o profesionales del cuidado de la salud, con el inconveniente de los registros médicos generalmente se encuentren desorganizados, incompletos o incluso perdidos. Esto es posible sólo porque los pacientes deciden por cuenta propia adoptar un perfil pasivo en su cuidado; la gente necesita gestionar su propio registro de salud e ir acumulando su información para acceder a ésta cuando se requiera.

Se observa que el uso y desarrollo de este tipo de herramientas, los Registros Personales de Salud, es todavía incipiente y requiere tiempo para llegar a un nivel aceptable de cobertura y calidad; no hay una estructura definida o estandarizada, y muchos usuarios aún no entienden el concepto de un RPS, en parte debido a que todavía existe una barrera mental para la adopción de registros personales, bien sea por parte del paciente o por parte del médico.

Por parte del paciente, se tiene la concepción de que el sistema puede implicar una carga al tener que actualizarlo constantemente, incluso a veces hay problema en entender los términos médicos o lo que se debe registrar en cada área. Por parte del médico, estos sistemas pueden representar trabajo extra al tener que entender el funcionamiento del mismo, para llegar a darle un uso. En ambos casos, se tiene que superar la fase inicial de uso, para entender los beneficios a largo plazo.

Respecto a los objetivos de la tesis hay que mencionar que el objetivo principal fue crear un sistema capaz de registrar, manipular y procesar la información de salud más común y sencilla que se maneja en la relación médico-paciente, objetivo que se cumplió satisfactoriamente.

Además se demostró que la participación de alguna autoridad que impulse el sistema, en este caso los profesores de las materias de IB y algunos otros médicos, realmente hace una diferencia en el grado de aceptación de un registro de salud.

En ese sentido, si se les acostumbra a los estudiantes de medicina a usar el sistema, es muy probable que lleguen a utilizarlo con sus pacientes y de esa manera ir poco a poco aumentando el

número de usuarios con cada generación de alumnos que cursen la materia de IB. Por lo tanto, en versiones futuras, se seguirá trabajando en conjunto con los médicos, para impulsar no sólo la adopción, sino también en la difusión del proyecto.

Se identificó también que hay una necesidad de pasar la aplicación a plataformas móviles, si bien esto no se reflejó de manera directa en los comentarios recibidos.

Además, como cualquier estudio, este trabajo tiene una limitante cualitativa pues el tamaño de la muestra estadística es pequeño y concentrado; el perfil de usuario percibido fue el de estudiantes interesados en el cuidado de la salud. Los usuarios fueron personas que está en contacto cercano con la tecnología y sobre todo con las tecnologías móviles, lo cual hace que se integre fácilmente al sistema. Esto último también implicaría un factor social que ayudaría en la integración del sistema a otro tipo de herramientas de carácter de redes sociales, lo que impulsaría notablemente su difusión y por consiguiente, su uso.

Finalmente, este trabajo se suma a los trabajos sobre las herramientas personales de salud dirigidas específicamente hacia los pacientes mexicanos, haciendo hincapié en los usuarios pues sus problemas y singularidades médicas, son diferentes de las del resto del mundo. Así, este trabajo apoya evitar utilizar sistemas enfocados o concebidos para el uso de ciudadanos extranjeros (principalmente estadounidenses), solución a la que se recurre frecuentemente en cuestiones de sistemas.

### **Recomendaciones para versiones futuras**

En un proyecto tan amplio como éste, siempre se desea que haya una mejora continua del mismo; por lo tanto una recomendación para futuros desarrollos es el especializar el sistema en grupos concretos de pacientes, sobre todo los que tienen mayor abundancia en México (diabéticos, hipertensos, personas con sobrepeso, etc.)

Otra recomendación más sería que, por cuestiones de seguridad, se omita en siguientes versiones el nombre real o al menos los apellidos del usuario propietario de la cuenta y/o que se manejen nombres de usuario en lugar del correo.

Se presentan a continuación algunas perspectivas generales para mejorar el sistema:

- Incluir recordatorios electrónicos (alertas de resultados de laboratorio críticos, recordatorios de vacunación, re-portal de interacciones de drogas).
- Incluir material de educación al paciente (dietas, información sobre prevención de enfermedades).

- Integrar al programa una interface que permita a los usuarios mediante dispositivos móviles, acceder al sistema de historias clínicas.
- Integrar sistemas eficientes de captura, procesamiento y almacenamiento de imágenes.
- Convertir o complementar el sistema con una aplicación personal de salud específicamente para plataformas móviles.

Debido a que se implementó la arquitectura MVC, valdría la pena considerar migrar el sistema en un futuro a un modelo MEAN (MongoDb, ExpressJS, AngularJS, NodeJS), en vez del modelo LAMP (Linux, Apache, MySQL, PHP) utilizado. Esto es posible debido a que una de las tecnologías base del modelo MEAN es AngularJS, un framework que se fundamenta en la separación de vistas, modelos y controladores para la implementación front-end, por lo que sería bastante adecuado. Aun así, habría que volver a revisar las ventajas de cambiar la implementación.

\*\*\*

Como comentario final, es indudable que se tiene que hacer un enorme esfuerzo para llegar a implementar de manera satisfactoria este tipo de herramientas, y posteriormente, los usuarios deben aceptarlas y acostumbrarse a ellas. Sin embargo, y a pesar de lo anterior, se planea continuar con el proyecto y seguir desarrollando el sistema con el fin de lograr la mejor experiencia y la mayor utilidad posibles para el usuario.

Vale la pena seguir impulsando este proyecto, con el fin de mejorar y agilizar el sistema de salud mediante la tecnología, y que de esa manera se beneficie la sociedad, sobre todo ciertos sectores que requieren una mayor atención en sus cuidados.

## VII. BIBLIOGRAFÍA

- [1] A. R. Ramírez, "Tratamiento jurídico de los datos clínicos en México (información y límites de acceso)," pp. 327–349, 2005.
- [2] C. G. A. Tenorio, "Los datos personales en México. Perspectivas y retos de su manejo en posesión de particulares," *Rev. Mex. Derecho Const.*, p. 258, 2013.
- [3] W. H. Curioso, J. A. Saldías, and R. Zambrano, "Historias clínicas electrónicas. Experiencia en un Hospital Nacional. Satisfacción por parte del personal de salud y pacientes.," vol. 15, no. 1, pp. 1–60, 2002.
- [4] J. A. G. Rodas, R. D. M. Hernández, L. P. M. Vélez, A. P. Moreno, P. A. A. Pérez, E. C. Gutiérrez, M. M. Cadavid, and A. R. Hamid, "Evaluación de la historia clínica sistematizada en la relación médico paciente de las IPS adscritas a Susalud, Medellín 2002.," *CES Med.*, vol. 17, no. 2, pp. 17–30, 2003.
- [5] A. H. Krist and S. H. Woolf, "A vision for patient-centered health information systems," *JAMA*, vol. 305, pp. 300–301, 2011.
- [6] M. Kyazze, J. Wesson, and K. Naude, "The design and implementation of a ubiquitous personal health record system for South Africa," *Glob. Telehealth 2014*, pp. 29 – 41, 2014.
- [7] S. Wells, R. Rozenblum, A. Park, M. Dunn, and D. W. Bates, "Personal Health Records for Patients with Chronic Disease," *Appl. Clin. Inform.*, vol. 5, no. 2, pp. 416–429, 2014.
- [8] G. L. Gaskin, C. a Longhurst, R. Slayton, and A. K. Das, "Sociotechnical Challenges of Developing an Interoperable Personal Health Record: Lessons Learned.," *Appl. Clin. Inform.*, vol. 2, no. 4, pp. 406–419, 2011.
- [9] A. Baird, F. North, and T. S. Raghu, "Personal Health Records (PHR) and the future of the physician-patient relationship," *Proc. 2011 iConference '11*, pp. 281–288, 2011.
- [10] S. L. Mora, "Programación de aplicaciones web: historia, principios básicos y clientes web," *Editor. Club Univ.*, vol. 53, no. 9, pp. 1689–1699, 2013.
- [11] A. C. Yera, *Programar desde un punto de vista científico*. Madrid: VISION NET, 2009.
- [12] S. Haldar, *SQLite Database System: Design and Implementation*, 2nd ed. Sibsankar Haldar - Self-Publishing, 2015.
- [13] A. C. Yera, *Diseño y programación de bases de datos*. VISION LIBROS, 2007.
- [14] J. Sánchez, "Diseño Conceptual de Bases de Datos," *Creat. Commons*, pp. 1–25, 2004.

- [15] A. Silberschatz, H. F. Korth, and S. Sudarshan, *Fundamentos de bases de datos*. 2002.
- [16] F. L. O. Rivera, *Bases de datos relacionales. Teoría y práctica*. Instituto Tecnológico Metropolitano, 2008.
- [17] M. J. Hernandez, *Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design*, 3rd ed. Addison-Wesley, 2013.
- [18] Compilación, *Redes avanzadas*. USERSHOP, 2007.
- [19] R. E. Yeager, Nancy J. & McGrath, *Web Server Technology: The Advanced Guide for World Wide Web Information Providers*. 1996.
- [20] Carlos G. González Salamea, "La Informática Médica y los Sistemas de Información.," p. 14, 2003.
- [21] L. Morell Baladrón, *Indicadores de actividad de las organizaciones sanitarias*, 1st ed. Díaz de Santos, 2014.
- [22] Secretaria de Salud, "Manual del Expediente Clínico Electrónico," p. 52, 2010.
- [23] Y.-T. Song, S. Hong, and J. Pak, "Empowering patients using cloud based personal health record system," *2015 IEEE/ACIS 16th Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel/Distributed Comput. (SNPD). Proc.*, pp. 293 – 8, 2015.
- [24] I. Sommerville, *Software Engineering*, 9th ed. Addison-Wesley, 2006.
- [25] U. of Alberta, "Software Processes and Agile Practices," in *Diplomado en Software Product Management*, 2015.
- [26] K. Beck, *Extreme Programming Explained*, 2nd ed. 2005.
- [27] C. Pitt, *Pro PHP MVC*. Springer Verlag GmbH, 2012.
- [28] P. McFarland, Ian; Harrison, *Mastering Tomcat Development*. Wiley Publishing, 2002.
- [29] D. Todorov, *Mechanics of User Identification and Authentication: Fundamentals of Identity Management*. Auerbach Publications, 2007.
- [30] F. de Medicina, *Historia Clínica Académica*, 3rd ed. UNAM, 2005.

- [31] V. DeBolt, *Mastering Integrated HTML and CSS*. 2007.
- [32] L. Ullman, *PHP and MySQL for dynamic web sites*, 4th ed. Peachpit Press, 2011.
- [33] P. M. Rodríguez, *Mantenimiento de portales de información*. VISION LIBROS, 2009.
- [34] U. O. Capuñay, *Desarrollo web con PHP*. Ocapunay, 2013.
- [35] J. D. Toal, Ray; Dionisio, *The JavaScript Programming Language*. Jones & Bartlett Learning, 2009.
- [36] D. S. McFarland, *CSS: The missing manual*. O'REILLY, 2009.
- [37] D. De Luca, *HTML5: entienda el cambio, aproveche su potencial*. USERSHOP, 2011.
- [38] H. Garcia-Molina, D. Ullman, and J. Widom, *Database Systems: The Complete Book*. 2002.
- [39] I. Group, *Introduction to Database Management Systems*. Tata McGraw-Hill, 2005.
- [40] B. Laurie and P. Laurie, *Apache: The Definitive Guide*, 2nd ed. O'REILLY, 2000.
- [41] M. Sabin-Wilson, Lisa; Miller, Cory; Palmer, Kevin; Rennick, Andrea; Torbert, *WordPress All-In-One*, vol. 29, no. 6. Wiley Publishing, 2000.

## **Agradecimientos:**

Son tantas las personas que participaron en este proyecto que no es una tarea sencilla enumerarlos a todos. Sin embargo, se hará mención a quienes fueron parte vital en la terminación de este proyecto y a quienes han trabajado conmigo desde pequeño para ayudarme a terminar una carrera profesional.

Muchos de ellos me apoyaron aún sin saber que estaba realizando la tesis, el servicio social y el examen de idiomas, pues se los escondí para darles la sorpresa hasta pocas semanas antes del examen profesional.

Agradezco a mi padre y a mi madre, por haber sido el principal apoyo hasta la fecha y durante todo el proceso de estudios; nunca me pidieron otra cosa más que dar lo mejor de mí y ser el mejor en lo que hacía. Esta tesis aparece está dedicada a ellos en las primeras páginas del escrito, simplemente con las palabras “A mis padres...” porque no puedo expresar correctamente ni completamente todas las razones por las que les dedico este trabajo.

Agradezco a mi hermano, pues fue la razón de que no perdiera la cabeza y pusiera los pies en la tierra; me hizo relajarme y divertirme en un momento en el que la tensión me hubiera destrozado de no haber liberado estrés y haber tenido a alguien con quien hablar sobre otras cosas que no fueran de la carrera. Es mi mejor amigo y compañero.

Agradezco a mi familia en general; especialmente a mis abuelos, personas maravillosas y cálidas con una percepción diferente de la vida; y a mis tíos y primos, con los que crecí y con los que me distraigo de vez en cuando del trabajo. Todos ellos han sabido soportarme y tenerme paciencia durante lo que llevo de vida, en especial estos últimos dos años y sin saber que estaba trabajando en la tesis.

Agradezco a mis amigos, quienes también han pasado o están pasando por este proceso y en quienes he encontrado consejos, guías y muchas anécdotas para poder terminar esta última etapa de estudios. Varios de ellos me han acompañado a lo largo de la carrera también.

Agradezco mi director de tesis quien aportó mucho al proyecto, tanto de conocimiento técnico para el desarrollo, como de conocimiento en redacción para la parte escrita. Más allá de ser solamente mi asesor, fue un buen compañero que me dirigió en este proceso y quien (no está de más ponerlo), me tuvo mucha paciencia durante el desarrollo del mismo.

Agradezco a todos los profesores que he tenido, no sólo en la formación académica o en el trabajo, sino a todas aquellas personas de las que he aprendido algo a lo largo de toda la vida.

A todos ellos y a muchas personas más les agradezco su cooperación, y espero que de alguna manera este trabajo haya tenido, o tenga en el futuro, algún impacto en sus vidas.