

**Universidad Internacional de La Rioja
Máster universitario en Ingeniería de Software y
Sistemas Informáticos**

**Aplicación web para liquidar
procedimientos médicos,
quirúrgicos y hospitalarios en
Colombia.**

Trabajo Fin de Máster

Presentado por: Torres Moreno, Charles Richar

Director/a: Machío Regidor, Francisco

Ciudad: Girardot
Fecha: 08/09/2017

Resumen

En este trabajo de fin de máster, TFM, se encuentra el diseño e implementación de un software desarrollado en un entorno web, para realizar liquidaciones de los procedimientos médicos, quirúrgicos y hospitalarios en Colombia, contemplados en el decreto 2423 de 1996, el acuerdo No. 256 del 2001 y el acuerdo 312 del 2004, conocidos como Manuales Tarifarios SOAT, ISS 2001 e ISS 2004 respectivamente.

Para su elaboración, se utiliza una metodología ágil de desarrollo de software, por ser iterativo e incremental. Las herramientas del lado del servidor que se usan son Apache, PHP para la codificación y MySQL para el almacenamiento en la base de datos. Por su parte, en el lado del cliente se utiliza HTML5, CSS3, JavaScript para la interactividad, AJAX y JSON para la comunicación con el servidor y acceso a los datos.

Palabras Clave: Manual tarifario, SOAT, ISS 2001, ISS 2004, Tarificación sanitaria

Abstract

In this work of end of Master, TFM, is the design and implementation of software developed in a web environment, to make liquidations of the medical, surgical and hospital procedures in Colombia, contemplated in decree 2423 of 1996, the agreement No, 256 of 2001 and agreement 312 of 2004, known as Tariff Manuals SOAT, ISS 2001 and ISS 2004 respectively.

For its elaboration, an agile methodology of software development is used, because iterative and incremental. The server side tools that are used are Apache, PHP for encoding and MySQL for storage in the database. On the client side, HTML5, CSS3, JavaScript for interactivity, AJAX and JSON are used for communication with the server and access to data.

Keywords: Tariff Manuals, SOAT, ISS 2001, ISS 2004, Health Tariff

Índice de contenidos

1. Introducción.....	10
1.1 Justificación	10
1.2 Planteamiento del trabajo	10
1.3 Estructura de la memoria.....	11
2. Contexto y estado del arte.....	13
3. Objetivos concretos y metodología de trabajo	17
3.1. Objetivo general.....	17
3.2. Objetivos específicos	17
3.2. Metodología del trabajo	17
4. Desarrollo específico de la contribución	22
4.1. Tipo 1. Desarrollo práctico	22
4.1.1. Identificación de requisitos	22
4.1.1.1. Requisitos del acuerdo No. 256 del 2001.....	22
4.1.1.2. Requisitos del acuerdo No. 312 del 2004.....	28
4.1.1.3. Requisitos del Decreto 2423 de 1996	32
4.1.2. Descripción del sistema software desarrollado	38
4.1.2.1. Casos de uso manual tarifario ISS 2201	38
4.1.2.2. Casos de uso manual tarifario ISS 2204	41
4.1.2.3. Casos de uso manual tarifario SOAT	44
4.1.2.4. Entorno de Desarrollo Integrado	47
4.1.3. Evaluación.....	71
5. Conclusiones y trabajo futuro	75
5.1. Conclusiones	75
5.2. Líneas de trabajo futuro	77
6. Bibliografía	78
Anexos	82
Artículo	82

Manual de Usuario.....	82
------------------------	----

Índice de tablas

Tabla 1 Plan básico intervenciones quirúrgicas y procedimientos en la especialidad de Neurocirugía.....	23
Tabla 2 Valor UVR manual tarifario ISS 2001 Artículo 59.....	24
Tabla 3 Valor derechos sala de cirugía (quirófanos) y de parto Artículo 77	24
Tabla 4 Materiales de sutura y curación Artículo 85	25
Tabla 5 Especialidades manual tarifario ISS 2001.....	26
Tabla 6 Operaciones del apéndice	27
Tabla 7 Ejemplo liquidación procedimiento quirúrgico	27
Tabla 8 Porcentajes de liquidación cirugías bilaterales	27
Tabla 9 Porcentajes de liquidación mismo especialista, misma y diferente vía de acceso....	28
Tabla 10 Porcentajes de liquidación diferente especialista.....	28
Tabla 11 Procedimientos e intervenciones quirúrgicas.....	29
Tabla 12 Resumen porcentajes de liquidación manual tarifario ISS 2004	30
Tabla 13 Los derechos de sala de curación	31
Tabla 14 Los derechos de sala de recuperación	31
Tabla 15 Materiales según UVR-S	31
Tabla 16 Operaciones del apéndice	32
Tabla 17 Ejemplo liquidación procedimiento quirúrgico	32
Tabla 18 Órganos intracraneales	33
Tabla 19 Servicios profesionales del cirujano o ginecobstetra.....	34
Tabla 20 Servicios profesionales del anestesiólogo	34
Tabla 21 Servicios profesionales de ayudantía quirúrgica	35
Tabla 22 Derechos de sala de cirugía	35
Tabla 23 Materiales.....	36
Tabla 24 Operación del apéndice.....	36
Tabla 25 Ejemplo liquidación procedimiento quirúrgico grupo 7	36
Tabla 26 Resumen porcentajes de liquidación manual tarifario SOAT	37

Tabla 27 Comparativa tuplas manuales tarifarios	52
---	----

Índice de figuras

Figura 1. Indicadores de inflación básica y su variación anual (Fuente: Banco de la República)	15
Figura 2. Modelo de espiral común para el desarrollo de software. (Fuente: Pressman, 2010, p.66).....	18
Figura 3. Funcionamiento del patrón modelo-vista-controlador (Babit, 2011, p.37)	20
Figura 4. Estructura nomenclatura código CUPS (Elaboración propia).....	22
Figura 5. Estructura capítulo I manual tarifario ISS 2001 (Elaboración propia)	23
Figura 6. Estructura capítulo I manual tarifario ISS 2004 (Elaboración propia)	29
Figura 7. Estructura capítulo III manual tarifario SOAT (Elaboración propia)	33
Figura 8. Caso de uso para liquidar un procedimiento en manual tarifario ISS 2001 (Elaboración propia con herramienta CASE Bouml)	38
Figura 9. Caso de uso para liquidar procedimiento bilateral en manual tarifario ISS 2001 (Elaboración propia con herramienta CASE Bouml)	39
Figura 10. Caso de uso para liquidar procedimientos con la misma y diferente vía de acceso en manual tarifario ISS 2001 (Elaboración propia con herramienta CASE Bouml)	40
Figura 11. Caso de uso para liquidar procedimientos con diferente especialista en manual tarifario ISS 2001 (Elaboración propia con herramienta CASE Bouml).....	41
Figura 12. Caso de uso para liquidar un procedimiento en manual tarifario ISS 2004 (Elaboración propia con herramienta CASE Bouml)	42
Figura 13. Caso de uso para liquidar procedimiento bilateral en manual tarifario ISS 2004 (Elaboración propia con herramienta CASE Bouml)	42
Figura 14. Caso de uso para liquidar procedimientos con la misma y diferente vía de acceso en manual tarifario ISS 2004 (Elaboración propia con herramienta CASE Bouml)	43
Figura 15. Caso de uso para liquidar procedimientos con diferente especialista en manual tarifario ISS 2004 (Elaboración propia con herramienta CASE Bouml).....	43
Figura 16. Caso de uso para liquidar un procedimiento en manual tarifario SOAT (Elaboración propia con herramienta CASE Bouml).....	44
Figura 17. Caso de uso para liquidar un procedimiento bilateral en manual tarifario SOAT (Elaboración propia con herramienta CASE Bouml)	45

Figura 18. Caso de uso para liquidar procedimientos con la misma y diferente vía de acceso en manual tarifario SOAT (Elaboración propia con herramienta CASE Bouml)	46
Figura 19. Caso de uso para liquidar procedimientos con la misma vía de acceso y diferente especialista en manual tarifario SOAT (Elaboración propia con herramienta CASE Bouml) .	46
Figura 20. Caso de uso para liquidar procedimientos con diferente vía de acceso y diferente especialista en manual tarifario SOAT (Elaboración propia con herramienta CASE Bouml).	47
Figura 21. Panel de control EasyPHP Devserver (Elaboración propia).....	50
Figura 22. La relación cuenta (Silberschatz, 2006, p. 30)	51
Figura 23. La relación pacientes (Elaboración propia)	53
Figura 24. La relación parametros_tarifarios (Elaboración propia).....	54
Figura 25. La relación liquidador_tarifario (Elaboración propia)	54
Figura 26. Clases y métodos en PHP (Padilla, 2017, p.16)	55
Figura 27. Clases en PHP (Padilla, 2017, p.17)	56
Figura 28. Ejemplo hoja de estilos en cascada CSS (Padilla, 2017, p.25)	57
Figura 29. Ejemplo de un Script en JavaScript (Padilla, 2017, p.10)	57
Figura 30. Proceso de edición y ejecución de un programa JavaScript (Mohedano, 2012, p.11)	58
Figura 31. Ejemplo de código en el propio documento HTML (Ribest, 2011, p.16)	58
Figura 32. Ejemplo de código en documento aparte (Ribest, 2011, p.17).....	59
Figura 33. Ejemplo funcionamiento archivo .htaccess (Elaboración propia)	61
Figura 34. Estructura de archivos patrón MVC (Elaboración propia)	61
Figura 35. Archivo Config.php (Elaboración propia con herramienta CASE Brackets).....	62
Figura 36. Archivo index.php (Elaboración propia con herramienta CASE Brackets)	62
Figura 37. Archivo Database.php (Elaboración propia con herramienta CASE Brackets).....	63
Figura 38. Archivo Model.php (Elaboración propia con herramienta CASE Brackets)	64
Figura 39. Archivo indexController.php (Elaboración propia con herramienta CASE Brackets)	64
Figura 40. Segmento archivo liquidadorModel.php (Elaboración propia con herramienta CASE Brackets)	65

Figura 41. Archivo index.phtml (Elaboración propia con herramienta CASE Brackets).....	65
Figura 42. Renderizado vista index.phtml (Elaboración propia)	66
Figura 43. Segmento archivo estilos.css (Elaboración propia con herramienta CASE Brackets)	66
Figura 44. Interfaz web Liquidador de Cirugías (Elaboración propia)	67
Figura 45. Búsqueda de procedimiento con autocompletado (Elaboración propia)	67
Figura 46. Script autocompletar procedimiento archivo liquidarCirugia.js (Elaboración propia con herramienta CASE Brackets).....	68
Figura 47. Método busca_procedimeinto de liquidadorController (Elaboración propia con herramienta CASE Brackets)	68
Figura 48. Método getProcedimeinto de liquidadorModel (Elaboración propia con herramienta CASE Brackets)	69
Figura 49. Uso de AJAX en la búsqueda de procedimientos (Elaboración propia con herramienta CASE Brackets)	70
Figura 50. Método tabla_procedimiento del controlador liquidadorController (Elaboración propria con herramienta CASE Brackets)	70
Figura 51. Análisis de contraste (Fuente http://wave.webaim.org/).....	72
Figura 52. Estructura de la aplicación web (Elaboración propia)	72
Figura 53. Sistema de comentarios de la aplicación web (Elaboración propia).....	73
Figura 54. Corrección unidades de liquidación Anestesiólogo tarifario SOAT (Elaboración propria).....	74

1. Introducción

En este trabajo se expone el diseño, desarrollo y resultados de una aplicación web para realizar liquidaciones de procedimientos médicos y quirúrgicos en Colombia, tomando como base los decretos y acuerdos, expedidos por las autoridades colombianas, donde se clasifican y aprueban las tarifas que se aplican a los usuarios de las entidades promotoras de salud.

1.1 Justificación

El problema por tratar es que actualmente no se cuenta con un aplicativo, de libre acceso, para realizar las capacitaciones o que sirva de consulta al público en general ante las liquidaciones que se llevan a cabo en las entidades que prestan sus servicios a los afiliados al plan obligatorio de salud.

La causa de esta problemática es que las aplicaciones desarrolladas para este fin son de uso privativo, integradas como parte de software de facturación y adquirido por las empresas que brindan estos servicios.

La importancia de la solución de este problema radica en aportar una herramienta, que pueda ser utilizada para capacitar al personal de las entidades promotoras de salud, en la aplicación práctica de los manuales tarifarios que actualmente se aplican en Colombia.

1.2 Planteamiento del trabajo

En el área de facturación de las instituciones de salud públicas, al ser instituciones con autonomía administrativa, técnica y financiera otorgadas por la ley 100, se maneja información sensible como datos personales de los pacientes, historias clínicas, gestión de cartera y son procesos auditables. Adicionalmente, en un párrafo del artículo 185 la ley 100 (1993) se indica que “toda Institución Prestadora de Servicios de salud contará con un sistema contable que permita registrar los costos de los servicios ofrecidos”. (p. 59)

Es claro que desarrollar un sistema que abarque todos los aspectos de un sistema contable, para un área de facturación de una institución prestadora de salud, requiere más tiempo del disponible para este trabajo, lo que se propone es el desarrollo de un módulo que realice las

liquidaciones de los procedimientos contemplados en los manuales tarifarios, orientado en el ámbito de aplicación educativo, es decir que sirva para capacitar al personal que hace parte del área de facturación, porque para integrarlo en el ámbito comercial son necesarios otros módulos como clientes, proveedores y contables.

La estructura que se presenta en el decreto 2423 de 1996 y los acuerdos 256 y 312 de 2001 y 2004 respectivamente, permite desarrollar un software que agilice las consultas de los más de 14.000 procedimientos que contemplan y que los liquide de forma automatizada.

En este sentido, se obtienen los requisitos a partir del análisis de los tres tarifarios, se crea la base de datos con los procedimientos que describe cada manual y para la codificación de la solución se tienen en cuenta las reglas de liquidación, dependiendo del tipo de procedimiento o los porcentajes según la cantidad de actos quirúrgicos.

De igual forma se realiza un análisis de las tecnologías web disponibles, para seleccionar el servidor de base de datos y servidor web, el lenguaje de codificación de la solución de lado del cliente y del servidor, el entorno de desarrollo local y finalmente el de producción para la publicación en línea.

1.3 Estructura de la memoria

La estructura de la presente memoria está comprendida por los siguientes capítulos:

Capítulo 1: Contiene la introducción, la justificación y el planteamiento del trabajo. Sus antecedentes y cómo la ejecución del proyecto va a ayudar en su posible solución. De igual forma se incluye un resumen de su estructura y contenidos.

Capítulo 2: Expone el contexto y el estado del arte, mostrando un resumen cronológico de la legislatura que da origen a los manuales tarifarios, sus principales características y ámbitos de aplicabilidad. También las fortalezas y debilidades que actualmente presentan.

Capítulo 3: Contiene el objetivo general y objetivos específicos propuestos para realizar el proyecto. Igualmente expone la metodología de desarrollo de proyectos de software aplicada.

Capítulo 4: Explica el desarrollo práctico de la contribución. Iniciando por la identificación de los requisitos del sistema, identificados a partir de cada uno de los tres manuales tarifarios

seleccionados, su estructura, procedimientos y normas de liquidación. Se continua con la descripción del sistema software desarrollado, partiendo de los casos de uso de cada manual, el entorno de desarrollo, los servidores web y de base de datos tanto en el ámbito local como su publicación en internet. Luego se realiza la implementación de la solución desde la base de datos, los lenguajes de programación del lado cliente y del servidor, el patrón arquitectónico MVC y la codificación de la solución. Finalmente, se realiza la evaluación de la solución desde el punto de vista de la usabilidad y la aplicabilidad con el problema propuesto.

Capítulo 5: En el capítulo final se exponen las conclusiones del trabajo realizado y las líneas de trabajo futuro.

Anexos: Incluye la información adicional para complementar los contenidos de este documento, como un manual de usuario, un artículo y la legislatura en que está basado todo el trabajo.

2. Contexto y estado del arte

En Colombia, con la ley 100 de 1993, se crea el sistema de seguridad social con el fin de unificar la normatividad y coordinar las entidades prestadoras de servicios de salud. El Plan Obligatorio de Salud POS se crea en su artículo 162:

El sistema general de seguridad social de salud crea las condiciones de acceso a un plan obligatorio de salud para todos habitantes del territorio nacional antes del año 2001. Este plan permitirá la protección integral de las familias a la maternidad y enfermedad general, en las fases de promoción y fomento de la salud y la prevención, diagnóstico, tratamiento y rehabilitación para todas las patologías, según la intensidad de uso y los niveles de atención y complejidad que se definan. (Ley 100, 1993, p.49)

Con esta ley, surge la necesidad de estandarizar los procedimientos médicos, quirúrgicos y hospitalarios, al igual que sus tarifas, nomenclatura y clasificación de los mismos.

El primero de los tarifarios que publica es el decreto 2423 de 1996 “por el cual se determina la nomenclatura y clasificación de los procedimientos médicos, quirúrgicos y hospitalarios del manual tarifario y se dictan otras disposiciones” (p.1) y es la base de la liquidación utilizada con pacientes víctimas en escenarios como accidentes de tránsito, desastres naturales y atentados terroristas.

El manual tarifario SOAT, decreto 2423 de 1996, en su capítulo 1, describe su campo de aplicación:

Artículo 1º. Campo de aplicación: El presente Decreto será de obligatorio cumplimiento para las Instituciones Prestadoras de Servicios de Salud Públicas. Las entidades privadas deberán aplicarlo obligatoriamente cuando se trate de atención de pacientes víctimas de accidentes de tránsito, desastres naturales, atentados terroristas, atención inicial de urgencias y los demás eventos catastróficos definidos por el Consejo Nacional de Seguridad Social en Salud. (Decreto 2423, 1996, p.1)

Este artículo es modificado por el decreto 887 de 2001 en su artículo 1:

El artículo 1º del Decreto 2423 de 1996, quedará así: “Artículo 1º. Campo de aplicación. El presente decreto será de obligatorio cumplimiento en los casos

originados por accidente de tránsito, desastres naturales, atentados terroristas y los demás eventos catastróficos definidos por el Consejo Nacional de Seguridad Social en Salud; también en la atención inicial de urgencias de otra naturaleza, si no hay acuerdo entre las partes. (Decreto 887, 2001, p.1)

Las tarifas para este manual se definen en Salarios Mínimos Legales Vigentes, lo que permite a un software de liquidación de procedimientos actualizar cada año estos valores. Este manual está dividido en cinco capítulos:

1. Campo de aplicación
2. Definiciones
3. Intervenciones y procedimientos médico-quirúrgicos, nomenclatura y clasificación.
4. Exámenes y procedimientos de diagnóstico y tratamiento, nomenclatura clasificación y tarifas.
5. Servicios intrahospitalarios y ambulatorios, estancias, servicios profesionales, derechos de sala, materiales, suministros y equipos, y paquetes de atención integral. Contenido y tarifas.

Posteriormente, el 19 de diciembre de 2001 se firma el acuerdo No. 256 “por el cual se aprueba el “Manual de Tarifas” de la Entidad Promotora de Salud del Seguro Social EPS-ISS” (p.81) con el que se liquidan todos los procedimientos de los servicios prestados por esta entidad, amparados por la ley 100 en su artículo 275 donde “al Consejo Directivo del Instituto le compete determinar las tarifas que el Instituto aplicará en la venta de servicios de salud.” (p.86)

Tres años después, el 24 de febrero de 2004 se publica el acuerdo 312 “por el cual se aprueba el “Manual de Tarifas” de la Entidad Promotora de Salud del Seguro Social EPS-ISS” (p.1) donde se actualiza la codificación a la Clasificación Única de Procedimientos en Salud (CUPS) y el contenido de los procedimientos del manual ISS 2001.

El inconveniente que actualmente se presenta con los manuales tarifarios, ISS 2001 y 2004, al no ser actualizados cada año, es que sus tarifas continúan invariables desde el momento en que fueron publicados. Actualmente, más de 13 años después, no se tiene en cuenta la devaluación del peso colombiano ni la inflación que oscila entre el 3% y 8% desde el año 2001 al mes de julio del año 2017, según el informe de indicadores de inflación básica y su variación anual del Banco de la República. La siguiente gráfica muestra la variación anual de los indicadores de inflación tomados de los datos de informe del Banco de la República.

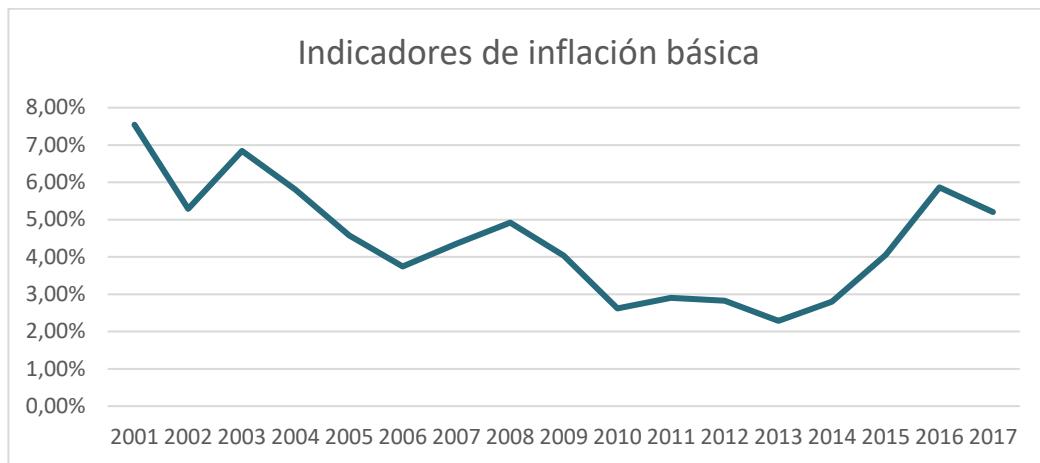


Figura 1. Indicadores de inflación básica y su variación anual (Fuente: Banco de la República)

El 28 de septiembre de 2012, se publica el decreto 2013 “Por el cual se suprime el Instituto de Seguros Sociales ISS, se ordena su liquidación, y se dictan otras disposiciones” (p.1). Esta medida involucra los tarifarios ISS 2001 y 2004 que eran los manuales utilizados por esa entidad pública del estado colombiano. De acuerdo con AFASALUD, “firmar un contrato de prestación de servicios de salud con tarifa iss genera un contrato ilegal [...] usted es el único responsable de la celebración de los contratos de su empresa.” (AFASALUD, 2017, p.4).

Desde esta perspectiva, no es recomendable celebrar contratos de prestación de servicios en salud con tarifas del ISS. Desde el punto de vista legal, el Acuerdo Número 256 de 2001 fue actualizado por el Acuerdo Número 312 de 2004 y con el decreto 2014 del año 2012 se liquidó el ISS y desde el punto de vista económico resulta irracional contratar servicios con precios del año 2001 o 2004.

Aunque por razones de rentabilidad económica, no es recomendable contratar teniendo como base de liquidación los tarifarios del ISS, se incluyen en el presente desarrollo por ser herramientas que actualmente se incluyen en capacitaciones sobre facturación de servicios en salud y también por ser una muy buena oportunidad de aplicar los conocimientos y técnicas adquiridas durante el desarrollo de los contenidos de Master Universitario en Ingeniería del Software y Sistemas Informáticos.

Adicionalmente, estos manuales, son importantes para las áreas de facturación de las entidades prestadoras de servicios en salud. De acuerdo con Leuro y Oviedo (2016), se puede definir facturación como:

1. Es el proceso de registro, valorización y liquidación de los procedimientos, actividades y consumos prestados a un paciente hospitalario o ambulatorio para efectos de cobro.
2. Conjunto de actividades que nos permiten liquidar la prestación de servicios de salud que conlleva a la atención al usuario en el hospital. (p.5)

Cabe resaltar que el proceso de facturación es un concepto muy amplio, porque implica generar facturas por liquidación de productos y servicios, contar con los medios físicos y tecnológicos para capturar la información de los pacientes y actividades, implementar mecanismos de control, sistemas de recaudo, servir como fuente confiable de información de los ingresos para determinar la rentabilidad y permitir a las directivas tomar decisiones, entre otros.

Por este motivo, en el presente proyecto, se toma como base la etapa correspondiente a la liquidación de procedimientos, a partir de los manuales que la legislación en Colombia ha contemplado para este fin. Otras etapas importantes para el proceso, como la generación y almacenamiento de facturas, el control del proceso en sus diferentes etapas o la administración de pacientes con sus respectivas historias clínicas se dejan como posibilidades para líneas de trabajo futuro.

A nivel de software, se enuncian las empresas más representativas del sector. Otras que desarrollan productos similares, no se mencionan a causa de la escasa información disponible. Una de las empresas que cuenta con completos sistemas de información Hospitalarios como es el caso de SIHOS WEB, de la empresa Sinergia S.A.S. <http://www.sinergia.in> que ofrecen un software para las áreas asistenciales y administrativas, con módulos de admisiones, historia clínica, consultas, facturación, entre otras.

Otra empresa es CNT <http://cnt.com.co>, especializados en el sector de la salud, con completos sistemas de software de escritorio y web, para administrar la información asistencial, administrativa y financiera de las entidades prestadoras de servicios de salud.

Teniendo en cuenta lo expuesto en este capítulo, en cuanto a la legislación que rige la tarificación de procedimientos médicos y los complejos sistemas de información que actualmente se encuentran en el mercado, se determina orientar este trabajo de fin de master a la creación de un módulo de liquidación de procedimientos, basado en las reglas establecidas por las normas que los rigen y aplicando las técnicas de desarrollo de software que fueron objeto de estudio en las diferentes asignaturas cursadas.

3. Objetivos concretos y metodología de trabajo

3.1. Objetivo general

Diseñar y codificar una aplicación en entorno web para liquidar procedimientos médicos, quirúrgicos y hospitalarios de los tarifarios ISS 2001, ISS 2004 y SOAT en Colombia.

3.2. Objetivos específicos

- Realizar el estudio de los tres tarifarios, su estructura, sus artículos y condiciones de liquidación.
- Identificar los requerimientos para el desarrollo de la aplicación.
- Establecer el lenguaje de codificación del lado del cliente y del servidor, el servidor de base de datos y el servidor web.
- Implementar la solución, publicar en el servidor web y realizar pruebas de funcionamiento.

3.2. Metodología del trabajo

Para el presente trabajo de fin de Master, se ha optado por el tipo de desarrollo práctico, donde se debe realizar un análisis, identificación de requerimientos, implementación de la solución y su posterior evaluación del funcionamiento.

Jacobson, Booch y Rumbaugh (2000) definen el proceso de desarrollo de software como “el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software” (p.4) mientras que, Sommerville (2005) lo define como “un conjunto de actividades que conducen a la creación de un producto software”. (p.60) Estas actividades es posible organizarlas de forma incremental e iterativa. De acuerdo con Pressman (2010) “el modelo espiral es un modelo evolutivo del proceso del software y se acopla con la naturaleza iterativa de hacer prototipos con los aspectos controlados y sistemáticos del modelo de cascada”. (p. 66)

Como se observa en la siguiente figura, el modelo en espiral permite organizar el desarrollo de una aplicación. Como primer resultado se obtienen las especificaciones del producto a partir de los manuales tarifarios, luego el modelado y codificación para generar un primer prototipo y posteriormente, versiones mejoradas del software de acuerdo con las pruebas y las retroalimentaciones:

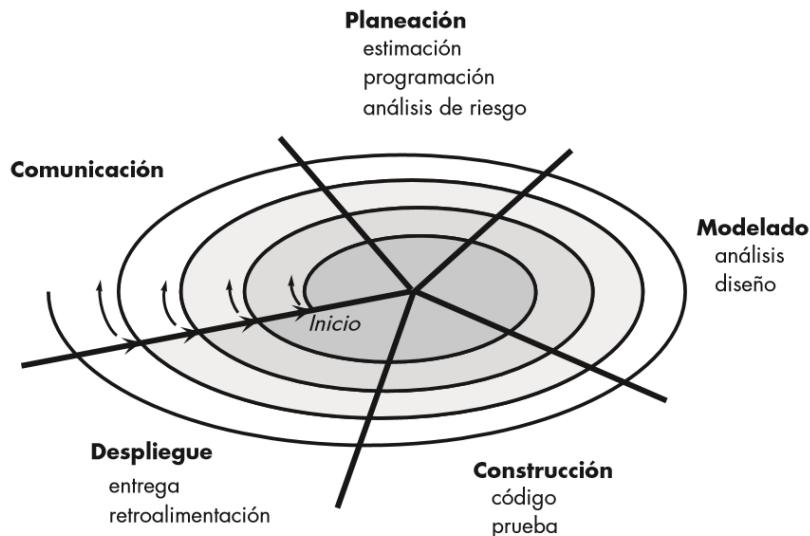


Figura 2. Modelo de espiral común para el desarrollo de software. (Fuente: Pressman, 2010, p.66)

Con el uso del modelo en espiral, se aplica el proceso unificado de desarrollo de software, elaborado por los autores del Lenguaje Unificado de Modelado (UML): Grady Booch, James Rumbaugh e Ivar Jacobson. Este proceso es basado en componentes que se conectan a través de interfaces y para modelar los artefactos del software utiliza UML.

Aplicando los conceptos del proceso unificado, al desarrollo del presente software, después de identificar los requisitos se tiene en cuenta:

Dirigido por casos de uso: Se especifican los casos de uso a partir del análisis de requisitos para cada uno de los tarifarios. Con el uso del Lenguaje Unificado de Modelado UML, se modelan los casos de uso para tener como punto de partida el desarrollo de las demás actividades.

Centrado en la arquitectura: Como recomienda Pressman (2010, 287) “Se elige un estilo arquitectónico [...] que sirve como plantilla para el diseño de la arquitectura del nuevo software”, el estilo arquitectónico que se aplica en este caso es: “**Médicos**: Sistemas que diagnostican, curan o contribuyen a la investigación médica.” Pressman (2010, 237).

Como patrón de arquitectura de software se establece el Modelo Vista Controlador (MVC) para separar la lógica de la aplicación en tres capas.

“El patrón MVC es un patrón de arquitectura de software encargado de separar la lógica de negocio de la interfaz del usuario y es el más utilizado en aplicaciones web, ya que facilita la funcionalidad, mantenibilidad y escalabilidad del sistema, de forma

simple y sencilla, a la vez que permite ‘no mezclar lenguajes de programación en el mismo código.’” (Bahit, 2011, p.36)

“El patrón Modelo-Vista-Controlador (MVC) es el patrón de diseño más adecuado y recomendado para aplicaciones interactivas que distribuyen las funcionalidades de dicha aplicación entre los distintos objetos que la componen, de manera que el grado de acoplamiento entre los objetos de la aplicación sea mínimo” (Martínez, Valderas, Pastor, 2010, p.138)

Como expresan los anteriores autores, el patrón MVC es el más adecuado al ser una aplicación interactiva que se desarrolla en entorno web y sus tres niveles de abstracción son:

Modelo: Hace referencia a la base de datos, que es este caso está almacenando los procedimientos especificados en cada manual tarifario.

Vista: Presenta la información al usuario para que interactúe con la aplicación, le permite seleccionar el tipo de tarifario con el que va a liquidar, buscar el procedimiento por nombre o código, seleccionar la vía de acceso, los especialistas entre otras características definidas en el capítulo de identificación de requisitos.

Controlador: Es el encargado de gestionar todas las peticiones del usuario para realizar las consultas a la base de datos, de forma asíncrona haciendo uso de AJAX, realizar las operaciones matemáticas para mostrar en pesos colombianos los valores de los procedimientos y enviar nuevamente el resultado de la consultas y operaciones a la vista para que el usuario pueda visualizar la información.

En la figura 3 se puede apreciar el funcionamiento de patrón MVC donde en primera instancia el cliente realiza una petición (1) que es capturada por el controlador, se procesa y se invoca (2) al modelo para realizar las consultas en la base de datos (3) y retorna la información al controlador (4). El controlador procesa y envía (5) la información a la vista organiza la información en la interfaz gráfica y la entrega (6) al usuario.

Iterativo e incremental: Si se realiza la división de este trabajo en partes más pequeñas, donde cada parte va a representar una iteración, resulta más fácil la detección de fallas, al tiempo que es menos abrumador ir cumpliendo pequeñas metas para llegar a un resultado final. Se inicia con los requisitos, estudiando cada tarifario para indicar lo que el sistema debe hacer, posteriormente con los casos de uso diagramados en UML, análisis para refinar y

estructurar los requisitos, diseño, implementación para la construcción del software y pruebas para verificar que funciona de manera correcta y que se cumple con los requisitos, culminando con un ciclo de desarrollo de software e iniciando con una nueva iteración.

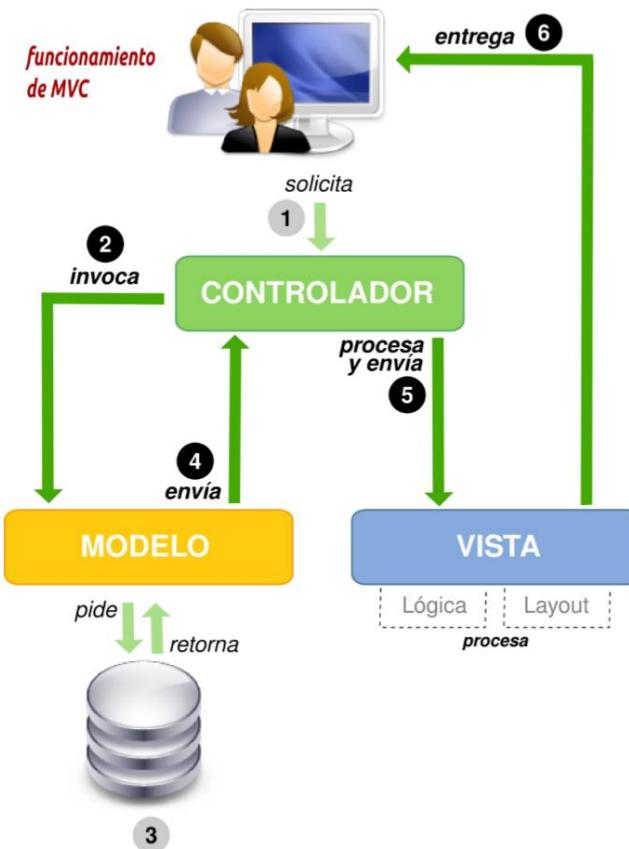


Figura 3. Funcionamiento del patrón modelo-vista-controlador (Babit, 2011, p.37)

Para facilitar el trabajo, es necesario el uso de otras herramientas de software especializadas que permiten diagramar en UML, codificar la aplicación y documentar, conocidas como herramientas CASE.

Herramientas CASE: CASE significa Computer Aided Software Engineering, Campderrick (2003, 31) las define como “herramientas CASE son software de apoyo al desarrollo, mantenimiento y documentación informatizados de software”.

En este tipo de herramientas se destacan Bouml para los diagramas implementados con el lenguaje unificado de modelado y el editor de código Brackets.

Desde el punto de vista, de lo expuesto es este capítulo, se opta por una metodología de desarrollo de software ágil, con ciclos iterativos e incrementales, generando prototipos para

concluir con una versión mejorada del software soportada por las pruebas y retroalimentaciones del ciclo anterior.

Como lo menciona Pressman, una metodología de desarrollo ágil va más allá de las herramientas, los ciclos iterativos y el producto de software, teniendo en cuenta el equipo de trabajo y un estrecho contacto con el cliente:

La ingeniería de software ágil combina una filosofía con un conjunto de lineamientos de desarrollo. La filosofía pone el énfasis en: la satisfacción del cliente y en la entrega rápida de software incremental, los equipos pequeños y muy motivados para efectuar el proyecto, los métodos informales, los productos del trabajo con mínima ingeniería de software y la sencillez general en el desarrollo. (Pressman, 2010, p.83)

4. Desarrollo específico de la contribución

Este desarrollo se realiza en un ambiente web, con tecnologías html5, CSS3 y JavaScript del lado del cliente, PHP del lado del servidor, con el patrón de arquitectura de software Modelo Vista Controlador (MVC), MySQL en las bases de datos y se publica la solución en un servidor Apache con el dominio <http://www.tarificador.hol.es/> para libre acceso. La selección de estas tecnologías se justifica en los capítulos a continuación.

4.1. Tipo 1. Desarrollo práctico

4.1.1. Identificación de requisitos

En un manual tarifario se contemplan actividades, procedimientos e intervenciones, identificados por códigos y nomenclaturas que permiten clasificarlos e implementarlos en un sistema de información.

En la resolución 5261 de 1994 el ministerio de salud establece el Manual de Actividades, Intervenciones y Procedimientos del Plan Obligatorio de Salud. En esta solución se utilizaron códigos y nomenclaturas no estandarizados y es por este motivo que con la resolución 2333 de 2000 y su posterior actualización en la resolución 1896 de 2001, se adopta la CLASIFICACIÓN ÚNICA DE PROCEDIMIENTOS EN SALUD (CUPS), organizando los procedimientos e intervenciones que se realizan en Colombia y los identifica con un código y una nomenclatura validada con la Clasificación Internacional de Enfermedades, versión española y es adaptada para Colombia por los expertos del sector de la salud. La estructura de su nomenclatura se muestra a continuación:

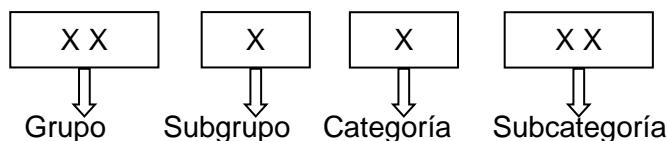


Figura 4. Estructura nomenclatura código CUPS (Elaboración propia)

4.1.1.1. Requisitos del acuerdo No. 256 del 2001

Tomando como base la Clasificación Única de Procedimientos en Salud (CUPS), en el 2001 se aprueba el acuerdo No. 256. En este manual tarifario, en adelante denominado manual tarifario ISS 2001, se presenta una equivalencia entre los códigos establecidos inicialmente

con los nuevos CUPS y su calificación en Unidades de Valor Relativo (UVR), como muestra en la siguiente figura:

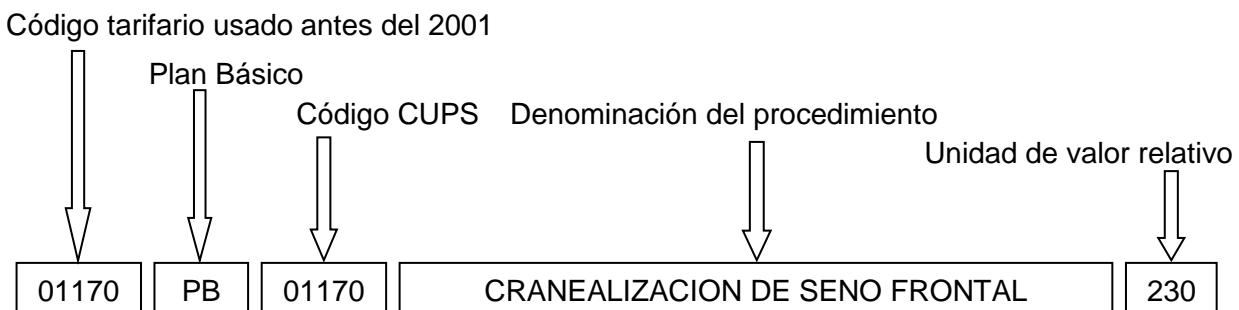


Figura 5. Estructura capítulo I manual tarifario ISS 2001 (Elaboración propia)

En la siguiente tabla se observa un segmento del Capítulo I, Artículo 1°, del acuerdo No. 256 que se toma como referencia para crear la base de datos que almacena los procedimientos del manual tarifario ISS 2001.

Tabla 1 Plan básico intervenciones quirúrgicas y procedimientos en la especialidad de Neurocirugía.

CRANEO				
MALFORMACIONES CONGENITAS				
Ref.	Código	Descripción		UVR
01100	PB	020101	CORRECION DE CRANEO SINOSTOSIS, POR CRANIECTOMIA SIN AVANCES	380
01101	PB	020102	CORRECION DE CRANEO SINOSTOSIS, POR CRANIECTOMIA CON AVANCE FRONTAL	380
01101	PB	020104	CORRECCION DE CRANEO ESTENOSIS MULTIPLE CON BRAQUICEFALIA, CON AVANCE FRONTAL	380
01102	PB	020106	CORRECCION DE CRANEO ESTENOSIS CON ALTERACION CRANEO FACIAL COMPLEJA, CON AVANCE FRONTO-ORBITARIO	620
01102	PB	020107	CORRECCION DE CRANEO ESTENOSIS CON ALTERACION CRANEO FACIAL COMPLEJA, CON AVANCE FRONTO NASAL (LEFORT III)	620
01102	PB	020108	CORRECCION DE CRANEO ESTENOSIS CON ALTERACION CRANEO FACIAL COMPLEJA, CON AVANCE FRONTO MAXILAR (LEFORT II)	620

01102	PB	021212	CORRECION DE MENINGOCELE DE FOSA ANTERIOR, CON AVANCE FRONTO ORBITARIO Y PLASTIA DE MENINGE	700
-------	----	--------	---	-----

Diario Oficial, Año CXXXVII No. 44.662, ACUERDO NUMERO 256 DE 2001, p.81

Para la liquidación de un procedimiento, se debe tener en cuenta las Unidades de Valor Relativo. En la siguiente tabla se presenta el valor de la UVR para Especialista, Anestesiólogo, Ayudante y Odontólogo general, según el Artículo 59:

Tabla 2 Valor UVR manual tarifario ISS 2001 Artículo 59

Ref.	Código	Descripción	Valor
39101	PB S41101	ESPECIALISTAS DE CLÍNICAS QUIRÚRGICAS O GINECOOBSTÉTRICAS	1.270
39102	PB S41201	ESPECIALISTAS EN ANESTESIOLOGÍA	960
39103	PB S41301	MÉDICO AYUDANTE QUIRÚRGICO	360
39145	PB S41401	MÉDICO U ODONTÓLOGO GENERAL	810

Diario Oficial, Año CXXXVII No. 44.662, ACUERDO NUMERO 256 DE 2001, p.140

El siguiente aspecto para tener en cuenta cuando se realiza una liquidación son los derechos de sala, que se liquidan según el número de UVR de la cirugía o procedimiento, como se muestra en esta tabla del Artículo 77:

Tabla 3 Valor derechos sala de cirugía (quirófanos) y de parto Artículo 77

Ref.	Código	Descripción	Valor
39209	PB S23101	HASTA 20 UVR	12.890
39210	PB S23102	DE 21 HASTA 30 UVR	26.790
39211	PB S23201	DE 31 HASTA 40 UVR	44.270
39212	PB S23202	DE 41 HASTA 50 UVR	55.605
39213	PB S23203	DE 51 HASTA 60 UVR	81.175
39214	PB S23204	DE 61 HASTA 70 UVR	96.520
39215	PB S23205	DE 71 HASTA 80 UVR	114.830
39216	PB S23301	DE 81 HASTA 90 UVR	129.655
39217	PB S23302	DE 91 HASTA 100 UVR	144.645
39218	PB S23303	DE 101 HASTA 110 UVR	148.545
39219	PB S23304	DE 111 HASTA 130 UVR	153.075

39220	PB	S23305	DE 131 HASTA 150 UVR	186.410
39221	PB	S23306	DE 151 HASTA 170 UVR	204.700
39222	PB	S23307	DE 171 HASTA 200 UVR	246.970
39223	PB	S23308	DE 201 HASTA 230 UVR	279.405
39224	PB	S23309	DE 231 HASTA 260 UVR	318.255
39225	PB	S23310	DE 261 HASTA 290 UVR	356.455
39226	PB	S23311	DE 291 HASTA 320 UVR	401.015
39227	PB	S23312	DE 321 HASTA 350 UVR	445.560
39228	PB	S23313	DE 351 HASTA 380 UVR	471.015
39229	PB	S23314	DE 381 HASTA 410 UVR	503.460
39230	PB	S23315	DE 411 HASTA 450 UVR	548.020

Diario Oficial, Año CXXXVII No. 44.662, ACUERDO NUMERO 256 DE 2001, p.142

Para intervenciones superiores a 450 UVR, se debe multiplicar las unidades por \$1.410, según el Parágrafo 1°.

Luego de los derechos de sala, se debe proceder a liquidar los materiales de sutura y curación según la tabla que se muestra a continuación:

Tabla 4 Materiales de sutura y curación Artículo 85

Ref.	Código	Descripción	Valor
39301	PB	S55101	HASTA 20 UVR
39302	PB	S55102	DE 21 HASTA 30 UVR
39303	PB	S55103	DE 31 HASTA 40 UVR
39304	PB	S55104	DE 41 HASTA 50 UVR
39305	PB	S55105	DE 51 HASTA 60 UVR
39306	PB	S55106	DE 61 HASTA 70 UVR
39307	PB	S55107	DE 71 HASTA 80 UVR
39308	PB	S55108	DE 81 HASTA 90 UVR
39310	PB	S55109	DE 91 HASTA 100 UVR
39311	PB	S55110	DE 101 HASTA 110 UVR
39312	PB	S55111	DE 111 HASTA 130 UVR
39313	PB	S55112	DE 131 HASTA 150 UVR
39314	PB	S55113	DE 151 HASTA 170 UVR

Diario Oficial, Año CXXXVII No. 44.662, ACUERDO NUMERO 256 DE 2001, p.143

En el manual tarifario ISS 2001 los procedimientos se encuentran organizados de acuerdo con una región anatómica, una especialidad y el consecutivo del código CUPS, empezando por los procedimientos aplicados a la cabeza descendiendo hasta los pies. En primera instancia se encuentra la especialidad de Neurocirugía, seguidas por las cirugías de oftalmología, otorrinolaringología, como se muestra en la siguiente tabla:

Tabla 5 Especialidades manual tarifario ISS 2001

Artículo	Especialidad
1	Neurocirugía
2	Oftalmología
3	Otorrinolaringología
4	Cirugía de cuello
5	Cirugías cardiovasculares
6	Cirugía de tórax
7	Cirugía abdominal
8	Proctología
9	Urología y nefrología
10	Cirugía de mama
11	Ginecología
12	Obstetricia
13	Ortopedia y traumatología
14	Cirugía de mano
15	Cirugía plástica
16	Cirugía oral, maxilofacial y Dental
17	Toma de Biopsias
18	Endoscopia diagnóstica y terapéutica

Fuente: Elaboración propia, a partir de los artículos del acuerdo 256 de 2001

Para realizar el proceso de liquidación, se debe tener en cuenta las UVR de las tablas anteriores, es decir, para un procedimiento quirúrgico son necesarios servicios profesionales, compuestos de un Cirujano, un anestesiólogo y un ayudante, más los derechos de sala y materiales.

Para ilustrar el procedimiento se realiza la liquidación de una apendicectomía en las tablas 6 y 7:

Tabla 6 Operaciones del apéndice

Ref.	Código	Descripción	UVR
07731	PB	471100 APENDICECTOMÍA SOD	80

Diario Oficial, Año CXXXVII No. 44.662, ACUERDO NUMERO 256 DE 2001, p.94

Tabla 7 Ejemplo liquidación procedimiento quirúrgico

Art 59 Cirujano	\$ 1.270=	80 UVR	\$101.600=
Art 59 Anestesiólogo	\$ 960=	80 UVR	\$ 76.800=
Art 59 Ayudantía	\$ 360=	80 UVR	\$ 28.800=
Art 77 Derechos de sala	\$114.830=	De 71 hasta 80 UVR	\$114.830=
Art 85 Materiales	\$ 88.610=	De 71 hasta 80 UVR	\$ 88.610=
Total:			\$410.640=
Aprox. a la centena:			\$410.700=

Fuente: Elaboración propia, a partir de los artículos del acuerdo 256 de 2001

Otros cuatro casos de liquidación se presentan cuando se realizan múltiples cirugías. En estos casos se deben identificar los siguientes aspectos contemplados en los Artículos 65 – 66 – 67 y 68:

- Si el procedimiento es bilateral. Ver Tabla 8
- La vía de acceso, misma vía de acceso o diferente vía de acceso. Ver tabla 9
- Si las cirugías son realizadas por el mismo o diferente especialista. Ver tabla 10

Tabla 8 Porcentajes de liquidación cirugías bilaterales

Tipo Cirugía	Servicios Profesionales			Derechos de Sala	Materiales
	Cirujano	Anestesiólogo	Ayudantía		
Bilateral	100%	100%	100%	100%	100%
	75%	75%	75%	75%	75%

Fuente: Elaboración propia, a partir de los artículos del acuerdo 256 de 2001

En la tabla 9 el primer procedimiento sobre el que se realiza el cobro del 100% debe ser el de más alto costo.

Tabla 9 Porcentajes de liquidación mismo especialista, misma y diferente vía de acceso

Vía de Acceso	No.	Servicios Profesionales			Derechos de Sala	Materiales
		Cirujano	Anestesiólogo	Ayudantía		
Misma Vía	1	100%	100%	100%	100%	100%
	2	60%	60%	60%	50%	50%
	3	No se cobra	No se cobra	No se cobra	No se cobra	No se cobra
Diferente Vía	1	100%	100%	100%	100%	100%
	2	75%	75%	75%	50%	50%
	3	75%	75%	75%	50%	50%

Fuente: Elaboración propia, a partir de los artículos del acuerdo 256 de 2001

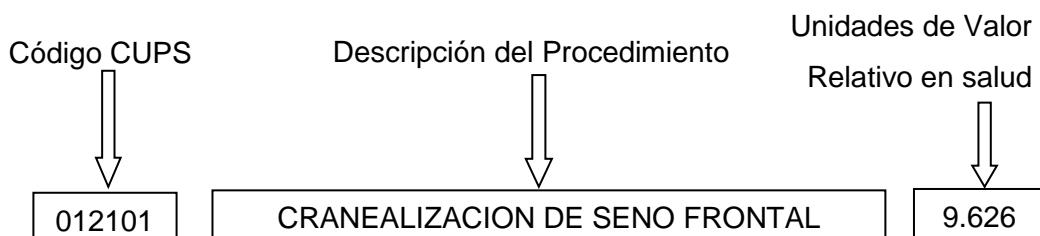
Tabla 10 Porcentajes de liquidación diferente especialista

Cirujano	No.	Servicios Profesionales			Derechos de Sala	Materiales
		Cirujano	Anestesiólogo	Ayudantía		
A	1	100%	100%	100%	100%	100%
	2	60%	No se cobra	No se cobra	50%	50%
	3	No se cobra	No se cobra	No se cobra	No se cobra	No se cobra
B	1	100%	75%	50%	100%	100%
	2	75%	No se cobra	No se cobra	50%	50%
	3	No se cobra	No se cobra	No se cobra	No se cobra	No se cobra

Fuente: Elaboración propia, a partir de los artículos del acuerdo 256 de 2001

4.1.1.2. Requisitos del acuerdo No. 312 del 2004

En marzo de 2004 se publica el acuerdo No. 312 (manual tarifario ISS 2004), con el fin de actualizar el Manual de Tarifas en aspectos como: el uso exclusivo de los códigos CUPS, actualización de procedimientos y el factor de liquidación en UVR-S Unidades de Valor Relativo en Salud.

**Figura 6.** Estructura capítulo I manual tarifario ISS 2004 (Elaboración propia)

En este manual tarifario se establece una forma de liquidación a partir de un solo valor para las Unidades de Valor Relativo en Salud, UVR-S = \$100, como ejemplo, para obtener la tarifa en pesos del procedimiento de la figura anterior, las 9.626 UVR-S por el factor de multiplicación \$100, la tarifa será de \$962.600.

Tabla 11 Procedimientos e intervenciones quirúrgicas

CUPS	DESCRIPCION_CUPS	UVR-S (incluye los servicios descritos en numeral 2)	
		REALIZADO POR MEDICO ESPECIALISTA	REALIZADO POR MEDICO U ODONTOLOGO GENERAL
Artículo 01. SISTEMA NERVIOSO			
01	PROCEDIMIENTOS EN CRANEO, CEREBRO Y MENINGES CEREBRALES		
010	INCISION DE CRANEO (PUNCION CRANEA)		
0101	PUNCION CISTERNAL		
010101	PUNCION CISTERNAL, VIA LATERAL	978	
010102	PUNCION CISTERNAL, VIA MEDIAL	978	
ARTICULO 03. SISTEMA VISUAL			
08	PROCEDIMIENTOS EN PARPADOS		
080	INCISION DE PARPADO		
080100	DRENAJE DE COLECCION POR BLEFAROTOMIA SOD	714	562

Diario Oficial, Año CXXXIX No. 45.478, ACUERDO NUMERO 312 DE 2004, p.5 – p.11

La tabla 11 se toma como referencia para crear la base de datos que almacena los procedimientos del manual tarifario ISS 2004, donde se debe tener en cuenta si el procedimiento es efectuado por un Médico Especialista o por un Médico u Odontólogo General.

Para liquidar cuando se realiza más de un procedimiento, es necesario tener en cuenta los porcentajes de la tabla 12, para procedimientos bilaterales, múltiples con misma o diferente vía de acceso y mismo o diferente cirujano.

Tabla 12 Resumen porcentajes de liquidación manual tarifario ISS 2004

PROCEDIMIENTOS BILATERALES		
	UVR-S	MATERIALES
BILATERAL	1	100
	2	75
MULTIPLES: MISMO ESPECIALISTA		
UNICA VIA (1)	1	100
	2	55
	3	-----
DIFERENTE. VIA (2)	1	100
	2	65
	3	65
MULTIPLES: DIFERENTES ESPECIALIDADES EN UNICA VIA		
CIRUJANO A	1	100
	2	40
	3	-----
CIRUJANO B	1	100
	2	40
	3	-----
MULTIPLES: DIFERENTES ESPECIALIDADES EN DISTINTA VIA		
CIRUJANO A	1	100
	2	40
	2	-----
CIRUJANO B	1	100
	2	40
	3	-----

Diario Oficial, Año CXXXIX No. 45.478, ACUERDO NUMERO 312 DE 2004, p.5

Igual que en el manual anterior, se aplican derechos de sala según las tablas 13 y 14 del Artículo 26.

Tabla 13 Los derechos de sala de curación

CUPS	DESCRIPCION CUPS	UVR-S
S22101	SALA DE CURACIONES	72
LOS DERECHOS DE SALA DE OBSERVACION EN EL SERVICIO DE URGENCIAS		
S20001	SERVICIO DE COMPLEJIDAD BAJA SOD	258
S20101	SERVICIO DE COMPLEJIDAD MEDIANA SOD	310
S20201	SERVICIO DE COMPLEJIDAD ALTA SOD	403

Diario Oficial, Año CXXXIX No. 45.478, ACUERDO NUMERO 312 DE 2004, p.87

Tabla 14 Los derechos de sala de recuperación

CUPS	DESCRIPCION CUPS	UVR-S
S24101	SERVICIO DE COMPLEJIDAD BAJA	338
S24102	SERVICIO DE COMPLEJIDAD MEDIANA	443
S24103	SERVICIO DE COMPLEJIDAD ALTA	533

Diario Oficial, Año CXXXIX No. 45.478, ACUERDO NUMERO 312 DE 2004, p.87

Ahora se continua el proceso de liquidación con los materiales, según el artículo 27, como se muestra en la tabla a continuación:

Tabla 15 Materiales según UVR-S

CUPS	DESCRIPCION CUPS	UVR-S
S55101	HASTA 421 UVR-S	326
S55102	DE 422 HASTA 810 UVR-S	336
S55103	DE 811 HASTA 1103 UVR-S	348
S55104	DE 1104 HASTA 1577 UVR-S	476
S55105	DE 1578 HASTA 1933 UVR-S	603
S55106	DE 1934 HASTA 3056 UVR-S	864
S55107	DE 3057 HASTA 3542 UVR-S	930
S55108	DE 3543 HASTA 3990 UVR-S	998
S55109	DE 3991 HASTA 4440 UVR-S	1147
S55110	DE 4441 HASTA 4768 UVR-S	1295
S55111	DE 4769 HASTA 5388 UVR-S	1377
S55112	DE 5389 HASTA 6324 UVR-S	1471
S55113	DE 6325 HASTA 7095 UVR-S	1606

Los materiales de UVR-S mayor a 7095 y en caso de cirugías múltiples con UVR-S menores, todos los materiales se pueden facturar según su consumo, hasta por el precio de adquisición más el 5%.

Diario Oficial, Año CXXXIX No. 45.478, ACUERDO NUMERO 312 DE 2004, p.87

Para ilustrar el procedimiento se realiza la liquidación de una apendicectomía en las siguientes tablas:

Tabla 16 Operaciones del apéndice

CUPS	DESCRIPCION CUPS	UVR-S (incluye los servicios descritos en numeral 2)
	REALIZADO POR	REALIZADO POR MEDICO
	MEDICO	U ODONTOLOGO
	ESPECIALISTA	GENERAL
471100	APENDICECTOMIA SOD	3.542

Diario Oficial, Año CXXXIX No. 45.478, ACUERDO NUMERO 312 DE 2004, p.30

Tabla 17 Ejemplo liquidación procedimiento quirúrgico

APENDICECTOMIA SOD	\$100=	3.542 UVR-s	\$354.200=
Art 27 Materiales	\$100=	De 3057 hasta 3542 UVR-S 930	\$ 93.000=
Total:			\$ 447.200 =

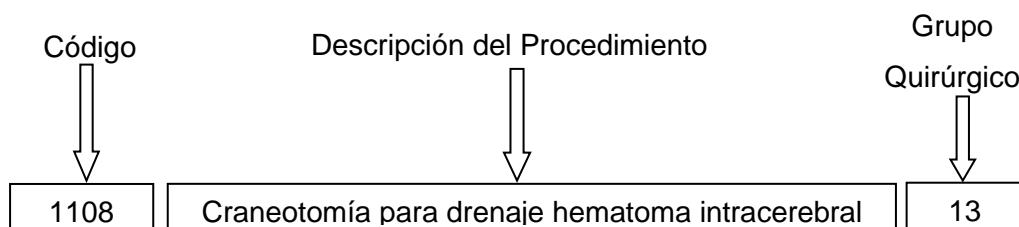
Fuente: Elaboración propia, a partir de los artículos del acuerdo 312 de 2004

Este valor del procedimiento clínico incluye: los servicios profesionales, el personal técnico y auxiliar, uso de los equipos, derechos de sala, consumo de cualquier material o elemento (reactivos, medios de contraste, película o papel fotográfico, material de sutura).

4.1.1.3. Requisitos del Decreto 2423 de 1996

El 31 de diciembre de 1996, el Gobierno Nacional Colombiano aprobó el decreto 2423, (manual tarifario SOAT) de obligatorio cumplimiento cuando se presenten pacientes víctimas de accidentes de tránsito, en Colombia el SOAT es un Seguro Obligatorio de Accidentes de Tránsito y por disposición de decreto 2423 también deben atender pacientes víctimas de desastres naturales, atentados terroristas y otras catástrofes que defina el Consejo Nacional de Seguridad Social en Salud.

En este manual tarifario, se observa el manejo de los procedimientos mediante grupos quirúrgicos, como se muestra en la figura 7.

**Figura 7.** Estructura capítulo III manual tarifario SOAT (Elaboración propia)

En la siguiente tabla, que se toma como referencia para crear la base de datos, se puede apreciar que, a diferencia de los tarifarios de los capítulos anteriores, los procedimientos médicos se encuentran clasificados según el grupo quirúrgico y nomenclatura.

Tabla 18 Órganos intracraneales

CRANEOTOMÍAS PARA TRATAMIENTO QUIRÚRGICO DE LESIONES INTRACRANEALES

CÓDIGO	DESCRIPCIÓN DEL PROCEDIMIENTO	GRUPO QUIRÚR.
1101	Craneotomía para extracción cuerpo extraño; incluye esquirlectomía	9
1102	Craneotomía para drenaje hematoma epidural o subdural	20
1103	Craneotomía para extracción secuestro	8
1104	Craneotomía para drenaje de hematoma de fosa posterior	20
1106	Craneotomía para ruptura de senos de duramadre	20
1107	Trepanación para monitoreo de presión intracraneana	12
1108	Craneotomía para drenaje hematoma intracerebral	13

Diario Oficial, Año CXXXII No. 42.961, DECRETO NUMERO 2423 DE 1996, p.2

Para la liquidación de un procedimiento, en este manual tarifario, se utiliza como base el Salario Mínimo Legal Vigente (SMLV), lo que garantiza que los precios de los procedimientos se actualicen automáticamente cada año. Por ejemplo, para el año 2017 el SMLV en Colombia es de \$737.717,00 y corresponde a un Salario Mínimo Legal Diario Vigente (SMLDV) de \$24.590,56.

Los conceptos para tener en cuenta en el proceso de liquidación son el cirujano, el anestesiólogo, el ayudante quirúrgico, los derechos de sala y los materiales. Los valores en SMLDV según el grupo quirúrgico se muestran en las siguientes tablas del artículo 48:

Tabla 19 Servicios profesionales del cirujano o ginecobiobstetra

CÓDIGO	GRUPO QUIRÚRGICO	SMLDV
39000	Grupo 02	2.93
39001	Grupo 03	3.57
39002	Grupo 04	4.31
39003	Grupo 05	5.86
39004	Grupo 06	7.68
39005	Grupo 07	9.00
39006	Grupo 08	10.44
39007	Grupo 09	12.76
39008	Grupo 10	15.71
39009	Grupo 11	17.62
39010	Grupo 12	19.63
39011	Grupo 13	21.48
39012	Grupo especial 20	25.43
39013	Grupo especial 21	33.10
39014	Grupo especial 22	38.67
39015	Grupo especial 23	60.73

Diario Oficial, Año CXXXII No. 42.961, DECRETO NUMERO 2423 DE 1996, p.106

Tabla 20 Servicios profesionales del anestesiólogo

CÓDIGO	GRUPO QUIRÚRGICO	SMLDV
39100	Grupo 02	2.09
39101	Grupo 03	2.53
39102	Grupo 04	3.10
39103	Grupo 05	3.83
39104	Grupo 06	4.56
39105	Grupo 07	5.30
39106	Grupo 08	6.17
39107	Grupo 09	7.30
39108	Grupo 10	9.02
39109	Grupo 11	10.08
39110	Grupo 12	11.44
39111	Grupo 13	12.72
39112	Grupo especial 20	14.85
39113	Grupo especial 21	20.12
39114	Grupo especial 22	26.82

39115	Grupo especial 23	37.95
39116	Parto normal o intervenido (forceps o espátulas) y revisión de cavidad uterina	4.96

Diario Oficial, Año CXXXII No. 42.961, DECRETO NUMERO 2423 DE 1996, p.107

El ayudante quirúrgico, se debe liquidar a partir de los procedimientos que sean del grupo 6 en adelante, en los grupos 2 al 5 no es necesario este servicio profesional debido a la menor complejidad de los procedimientos.

Tabla 21 Servicios profesionales de ayudantía quirúrgica

CÓDIGO	GRUPO QUIRÚRGICO	SMLDV
39117	Grupo 06	2.01
39118	Grupo 07	2.37
39119	Grupo 08	2.74
39120	Grupo 09	3.49
39121	Grupo 10	4.28
39122	Grupo 11	4.83
39123	Grupo 12	5.36
39124	Grupo 13	6.17
39125	Grupo especial 20	6.94
39126	Grupo especial 21	9.03
391127	Grupo especial 22	10.55
39128	Grupo especial 23	16.57

Diario Oficial, Año CXXXII No. 42.961, DECRETO NUMERO 2423 DE 1996, p.107

En el artículo 49 se presentan los derechos de sala que se muestran en la siguiente tabla:

Tabla 22 Derechos de sala de cirugía

CÓDIGO	GRUPO QUIRÚRGICO	SMLDV
39204	Grupo 02	4.84
39205	Grupo 03	5.97
39206	Grupo 04	7.61
39207	Grupo 05	10.45
39208	Grupo 06	15.13
39209	Grupo 07	16.88
39210	Grupo 08	18.51
39211	Grupo 09	21.10

39212	Grupo 10	28.08
39213	Grupo 11	29.95
39214	Grupo 12	31.47
39215	Grupo 13	33.16
39216	Grupo especial 20	34.82
39217	Grupo especial 21	36.51
39218	Grupo especial 22	38.34
39219	Grupo especial 23	48.07

Diario Oficial, Año CXXXII No. 42.961, DECRETO NUMERO 2423 DE 1996, p.109

En el artículo 55 tenemos los materiales a liquidar como se aprecia en la siguiente tabla:

Tabla 23 Materiales

CÓDIGO	GRUPO QUIRÚRGICO	SMLDV
39301	GRUPOS 02- 03	2.28
39302	GRUPOS 04- 05-06	4.27
39303	GRUPOS 07-08-09	9.92
39304	GRUPOS 10-11-12-13	15.72

Diario Oficial, Año CXXXII No. 42.961, DECRETO NUMERO 2423 DE 1996, p.110

Los materiales de los grupos especiales del 20 al 23 se deben liquidar según el precio comercial de catálogo para venta al público.

Para ilustrar el procedimiento se realiza la liquidación de una apendicectomía en las siguientes tablas:

Tabla 24 Operación del apéndice

CÓDIGO	DESCRIPCIÓN DEL PROCEDIMIENTO	GRUPO QUIRÚR.
7730	Apendicectomía	7

Diario Oficial, Año CXXXII No. 42.961, DECRETO NUMERO 2423 DE 1996, p.23

Tabla 25 Ejemplo liquidación procedimiento quirúrgico grupo 7

Art 48 Cirujano	\$24.590,56=	9,00	\$ 221.315,04=
Art 48 Anestesiólogo	\$24.590,56=	5,30	\$ 130.329,97=
Art 48 Ayudantía	\$24.590,56=	2,37	\$ 58.279,63=
Art 49 Derechos de sala	\$24.590,56=	16,88	\$ 415.088,65=

Art 55 Materiales	\$24.590,56=	9,92	\$ 243.938,36=
		Total:	\$1'068.915,64=
		Aprox. a la centena	\$1'069.000,oo=

Fuente: Elaboración propia, a partir de los artículos del decreto 2423 de 1996

Otros casos de liquidación se presentan cuando se realizan múltiples cirugías:

Tabla 26 Resumen porcentajes de liquidación manual tarifario SOAT

PROCEDIMIENTOS BILATERALES						
		Servicios Profesionales			Derechos	Materiales
		Cirujano	Anestesiólogo	Ayudantía	de Sala	
BILATERAL	1	100%	100%	100%	100%	100%
	2	75%	75%	75%	50%	75%
MULTIPLES: MISMO ESPECIALISTA						
UNICA VIA	1	100%	100%	100%	100%	100%
	2	50%	50%	50%	-----	-----
	3	50%	50%	50%	-----	-----
DIFERENTE VIA	1	100%	100%	100%	100%	100%
	2	75%	75%	75%	50%	75%
	3	75%	75%	75%	50%	75%
MULTIPLES: DIFERENTES ESPECIALIDADES EN UNICA VIA						
CIRUJANO A	1	100%	100%	100%	100%	100%
	2	50%	75%	-----	50%	-----
	3	50%	75%	-----	50%	-----
CIRUJANO B	1	100%	100%	50%	50%	-----
	2	50%	75%	-----	50%	-----
	3	50%	75%	-----	50%	-----
MULTIPLES: DIFERENTES ESPECIALIDADES EN DISTINTA VIA						
CIRUJANO A	1	100%	100%	100%	100%	100%
	2	50%	75%	-----	50%	75%
	3	50%	75%	-----	50%	75%
CIRUJANO B	1	100%	100%	50%	50%	75%
	2	50%	75%	-----	50%	75%
	3	50%	75%	-----	50%	75%

Fuente: Elaboración propia, a partir de los artículos del decreto 2423 de 1996

4.1.2. Descripción del sistema software desarrollado

4.1.2.1. Casos de uso manual tarifario ISS 2201

Diagramas UML para Liquidar Procedimientos Quirúrgicos.

El diagrama de la figura 8, representa la forma en que se debe liquidar un procedimiento en el manual tarifario ISS 2001, iniciando por la consulta y posteriormente se realiza la liquidación en UVR del cirujano, el anestesiólogo, el ayudante, los derechos de sala y los materiales.

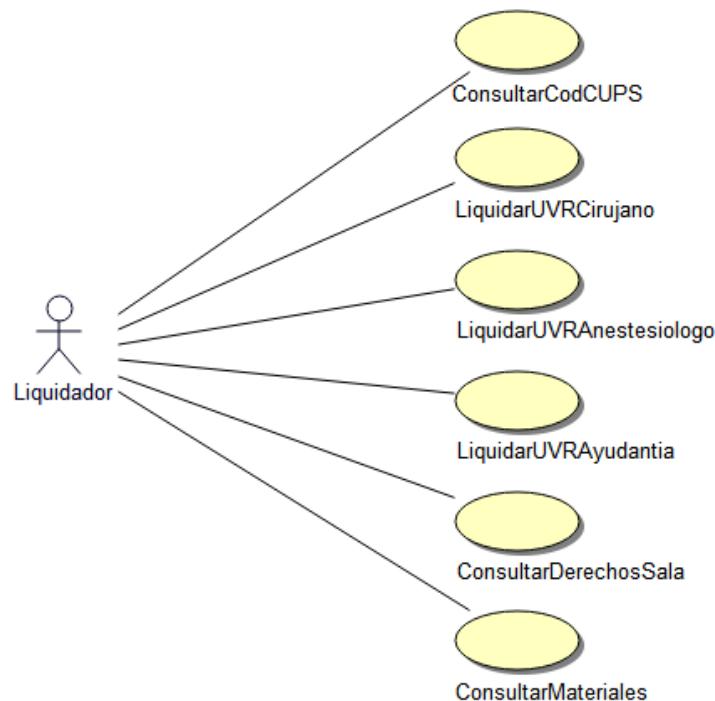


Figura 8. Caso de uso para liquidar un procedimiento en manual tarifario ISS 2001 (Elaboración propia con herramienta CASE Bouml)

El diagrama de la figura 9, representa la forma en que se debe liquidar un procedimiento bilateral en el manual tarifario ISS 2001, iniciando por la consulta de cada procedimiento y posteriormente se realiza la liquidación en UVR del cirujano, el anestesiólogo, el ayudante,

los derechos de sala y los materiales al 100%. Para el segundo procedimiento los porcentajes de liquidación se manejan al 75%.

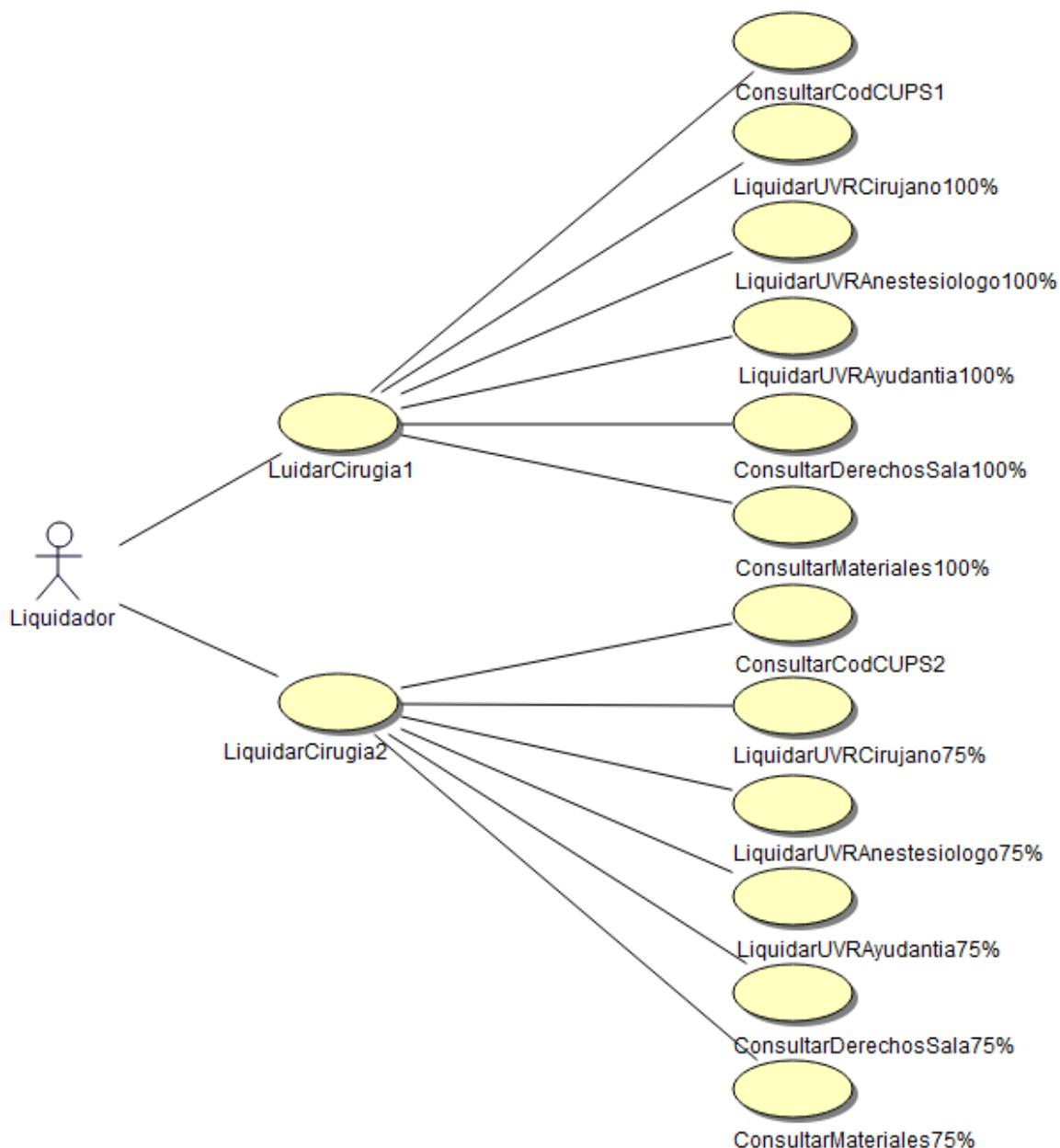


Figura 9. Caso de uso para liquidar procedimiento bilateral en manual tarifario ISS 2001 (Elaboración propia con herramienta CASE Bouml)

El diagrama de la figura 10, representa la forma en que se debe liquidar los procedimientos con la misma y diferente vía de acceso en el manual tarifario ISS 2001, iniciando por selección de la vía, la consulta de cada procedimiento y posteriormente se realiza la liquidación en UVR del cirujano, el anestesiólogo, el ayudante, los derechos de sala y los materiales al 100%. Para el segundo procedimiento los porcentajes de liquidación se manejan como se indica en

la figura y en el tercer procedimiento según el caso no se cobran o se liquidan como se observa a continuación:

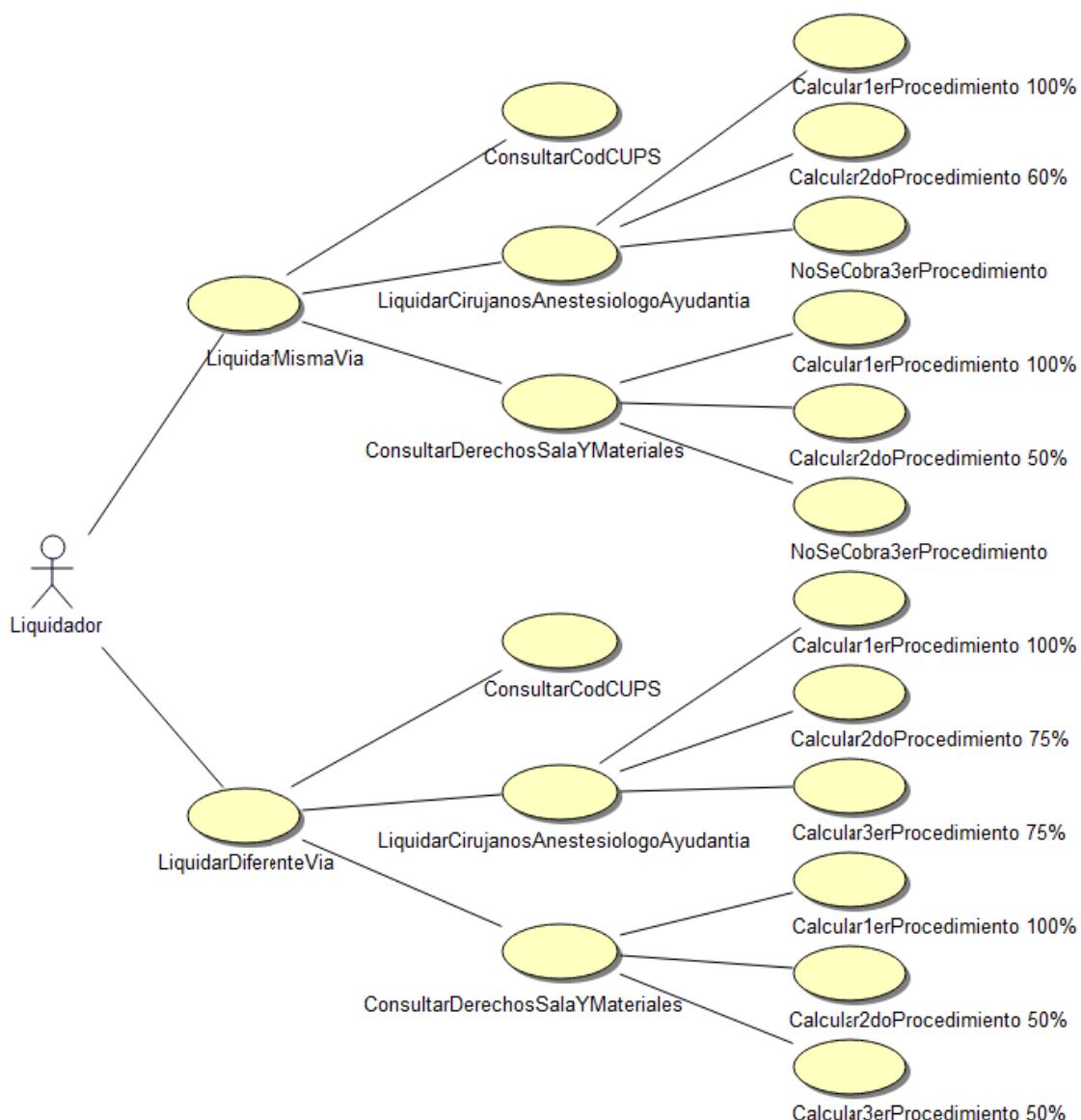


Figura 10. Caso de uso para liquidar procedimientos con la misma y diferente vía de acceso en manual tarifario ISS 2001 (Elaboración propia con herramienta CASE Bouml)

El diagrama de la figura 10, representa la forma en que se debe liquidar los procedimientos con diferente especialista en el manual tarifario ISS 2001, iniciando la consulta de cada procedimiento y posteriormente se realiza la liquidación en UVR del cirujano, el anestesiólogo, el ayudante, los derechos de sala y los materiales al 100%. Para el segundo y tercero procedimiento los porcentajes de liquidación se manejan como se indica en la figura:

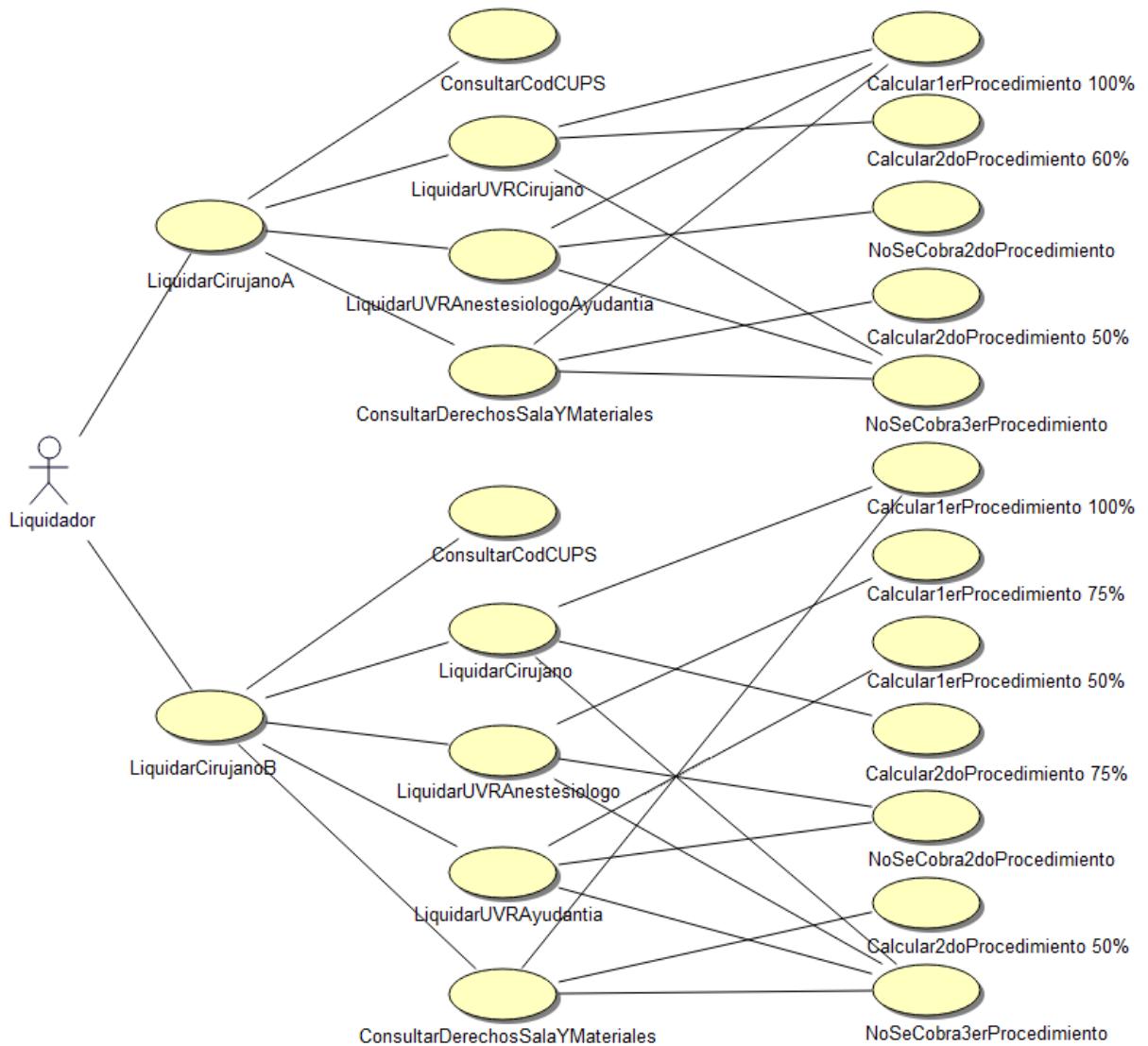


Figura 11. Caso de uso para liquidar procedimientos con diferente especialista en manual tarifario ISS 2001 (Elaboración propia con herramienta CASE Bouml)

4.1.2.2. Casos de uso manual tarifario ISS 2204

Diagramas UML para Liquidar Procedimientos Quirúrgicos.

El diagrama de la figura 12, representa la forma en que se debe liquidar un procedimiento en el manual tarifario ISS 2004, iniciando por la consulta y posteriormente se realiza la liquidación en UVR-S del procedimiento y los materiales. La figura 13 representa los porcentajes de liquidación para procedimientos bilaterales.

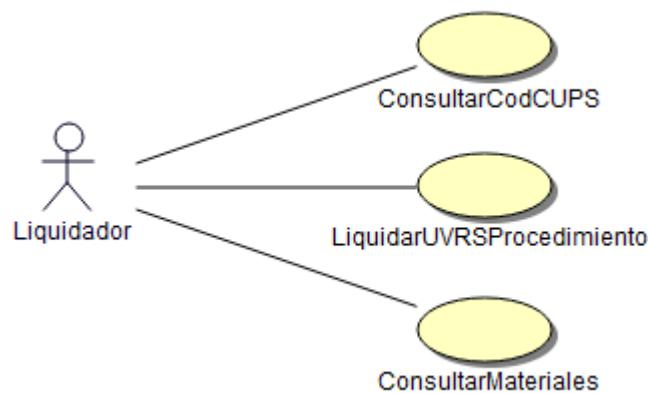


Figura 12. Caso de uso para liquidar un procedimiento en manual tarifario ISS 2004 (Elaboración propia con herramienta CASE Bouml).

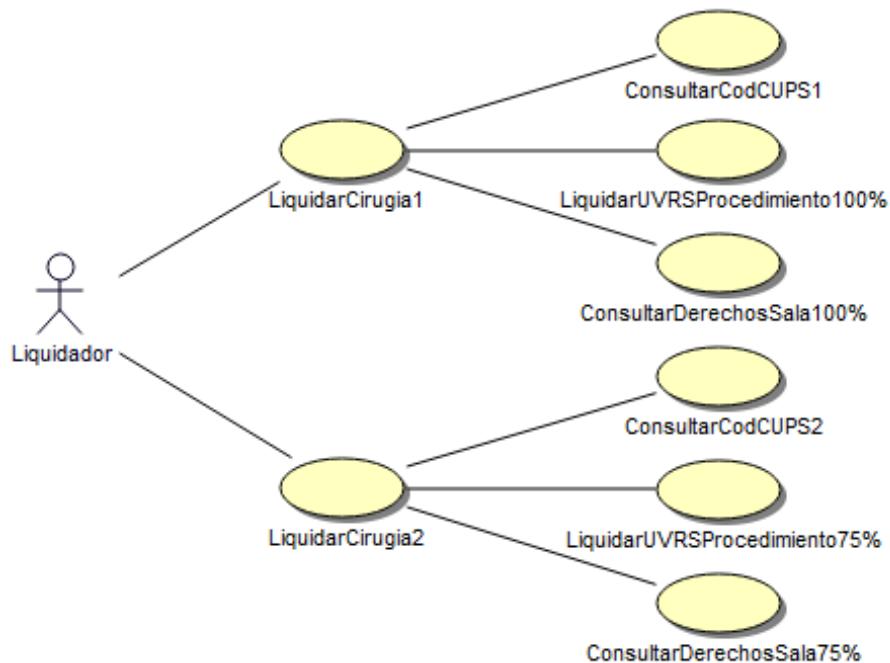


Figura 13. Caso de uso para liquidar procedimiento bilateral en manual tarifario ISS 2004 (Elaboración propia con herramienta CASE Bouml)

Los diagramas de las figuras 14 y 15 se presentan los porcentajes para procedimientos con única y diferente vía de acceso, al igual que diferente especialista.

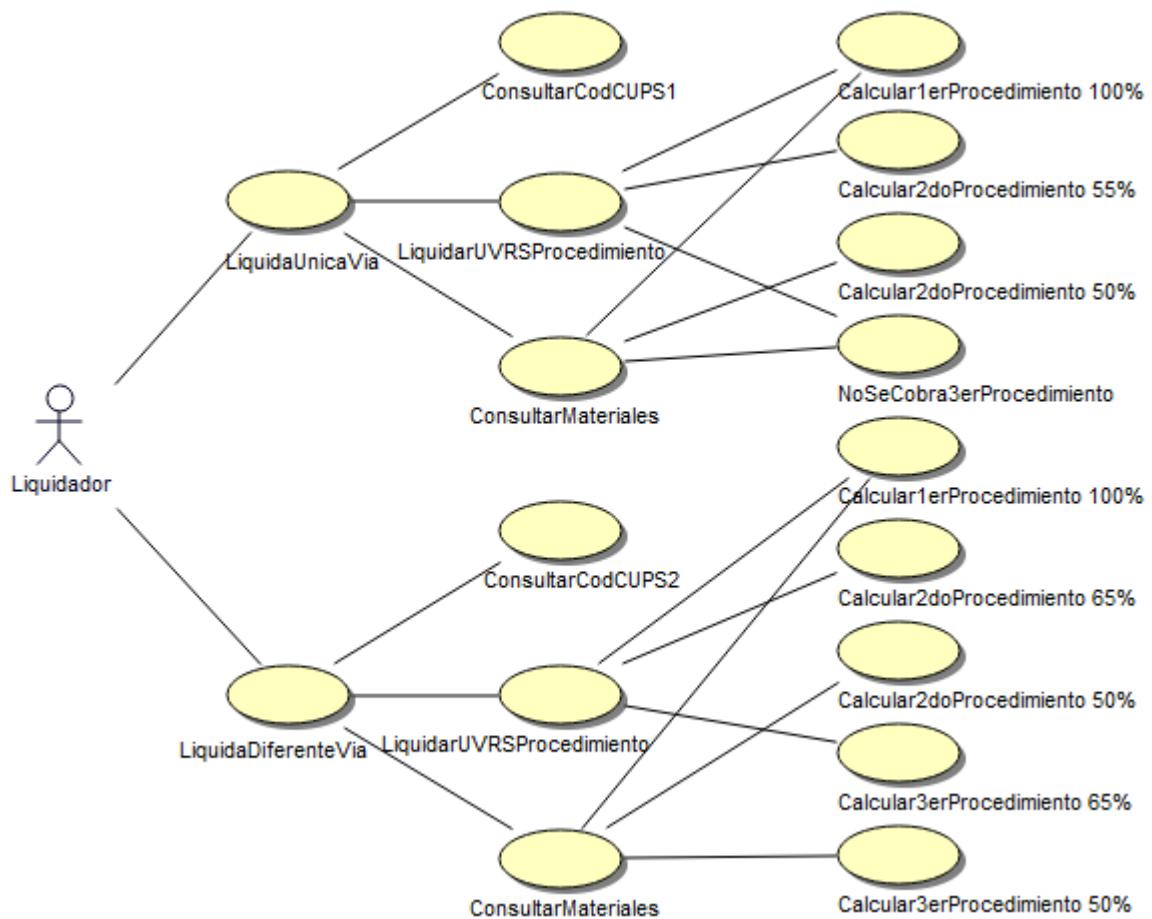


Figura 14. Caso de uso para liquidar procedimientos con la misma y diferente vía de acceso en manual tarifario ISS 2004 (Elaboración propia con herramienta CASE Bouml)

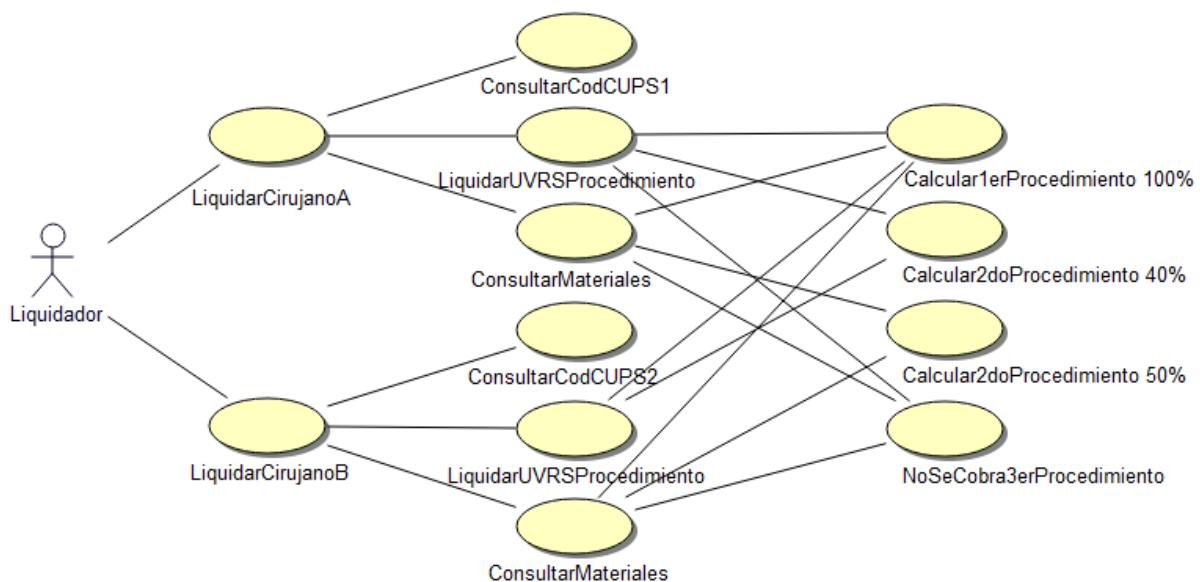


Figura 15. Caso de uso para liquidar procedimientos con diferente especialista en manual tarifario ISS 2004 (Elaboración propia con herramienta CASE Bouml)

4.1.2.3. Casos de uso manual tarifario SOAT

Diagramas UML para Liquidar Procedimientos Quirúrgicos.

El diagrama de la figura 16, representa la forma en que se debe liquidar un procedimiento en el manual tarifario SOAT, iniciando la selección de año, luego la consulta y posteriormente se realiza la liquidación en SLMDV del cirujano, el anestesiólogo, el ayudante, los derechos de sala y los materiales.

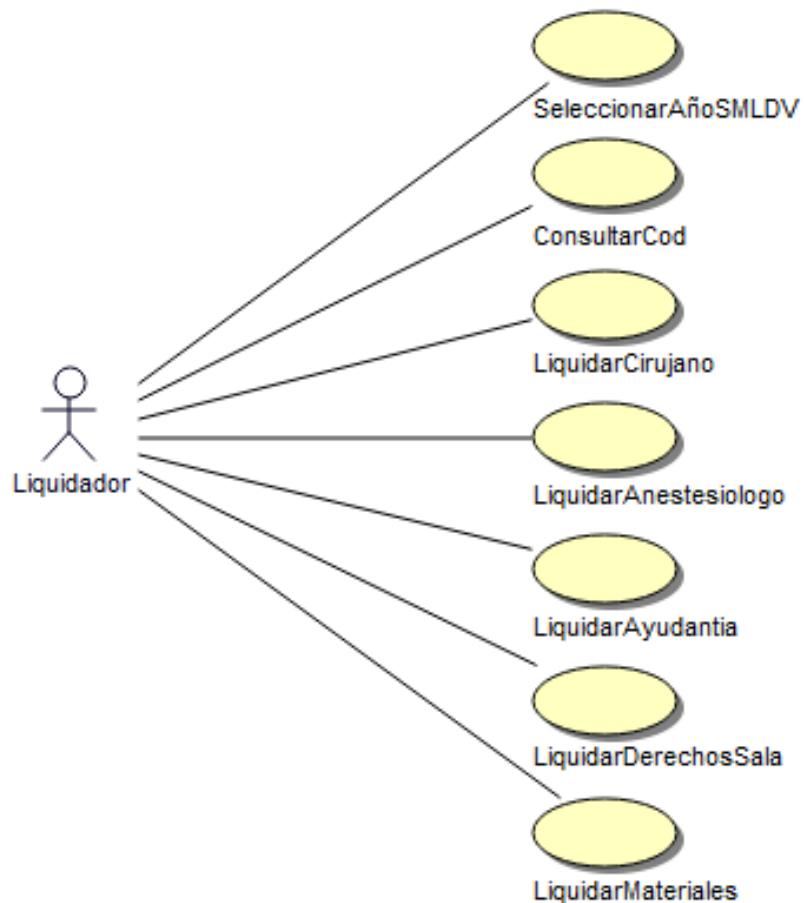


Figura 16. Caso de uso para liquidar un procedimiento en manual tarifario SOAT (Elaboración propia con herramienta CASE Bouml)

El diagrama de la figura 17, representa la forma en que se debe liquidar un procedimiento bilateral en el manual tarifario SOAT, iniciando por la selección del año, la consulta de cada procedimiento y posteriormente se realiza la liquidación en SMLDV del cirujano, el anestesiólogo, el ayudante, los derechos de sala y los materiales al 100%. Para el segundo procedimiento los porcentajes de liquidación se manejan al 75%, a excepción de los derechos de sala que se liquidan al 50%

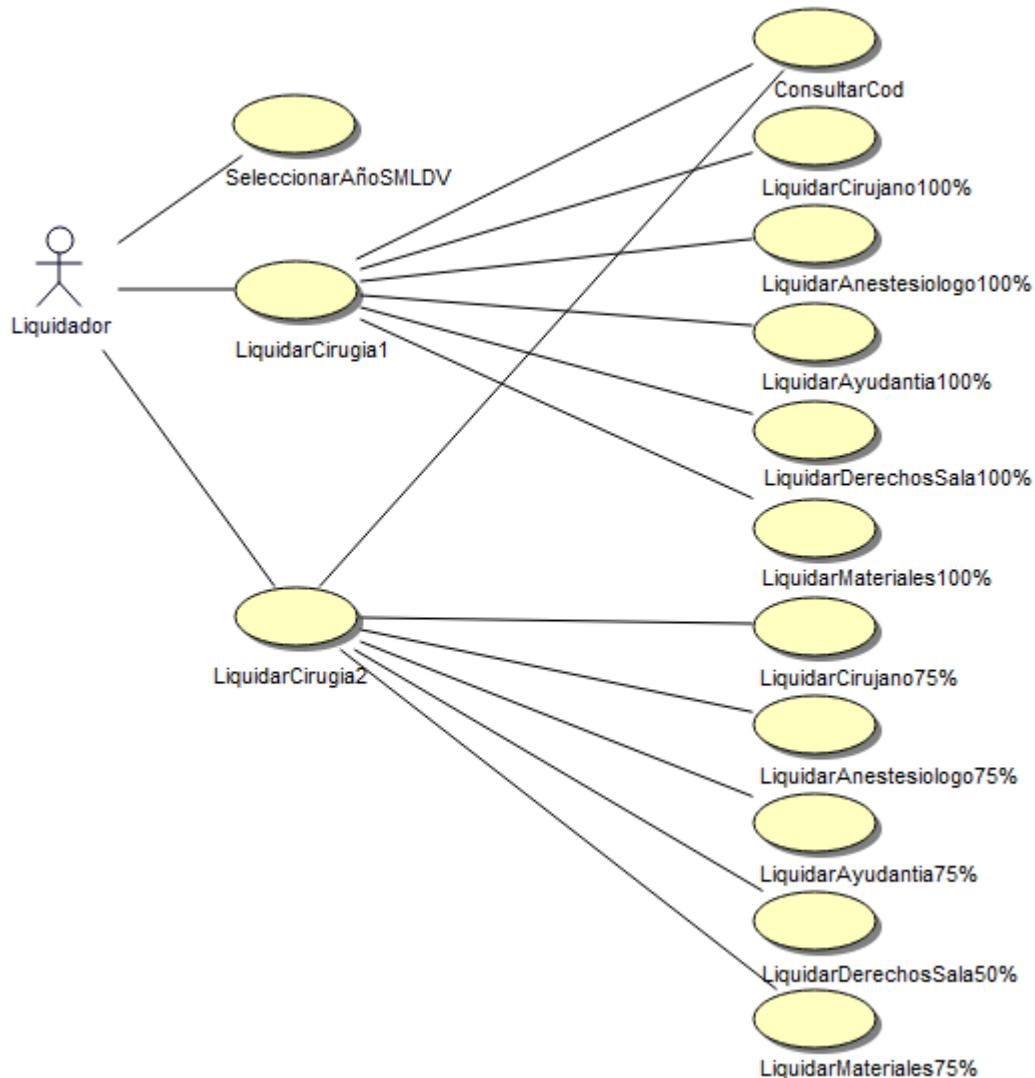


Figura 17. Caso de uso para liquidar un procedimiento bilateral en manual tarifario SOAT (Elaboración propia con herramienta CASE Bouml)

Los diagramas de las figuras 18 y 19, representan la forma en que se debe liquidar los procedimientos con la misma y diferente vía de acceso y con la misma vía de acceso con diferente especialista en el manual tarifario SOAT, iniciando por selección del año, la consulta de cada procedimiento y posteriormente se realiza la liquidación en SMLDV como se observa a continuación:

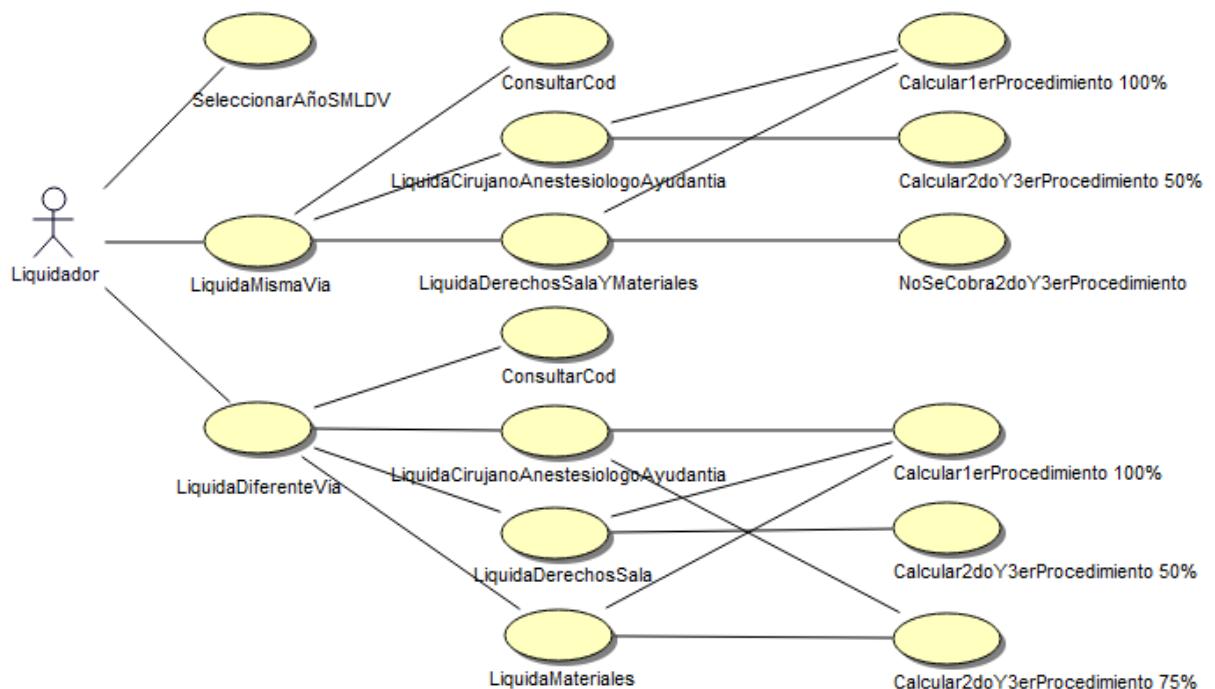


Figura 18. Caso de uso para liquidar procedimientos con la misma y diferente vía de acceso en manual tarifario SOAT (Elaboración propia con herramienta CASE Bouml)

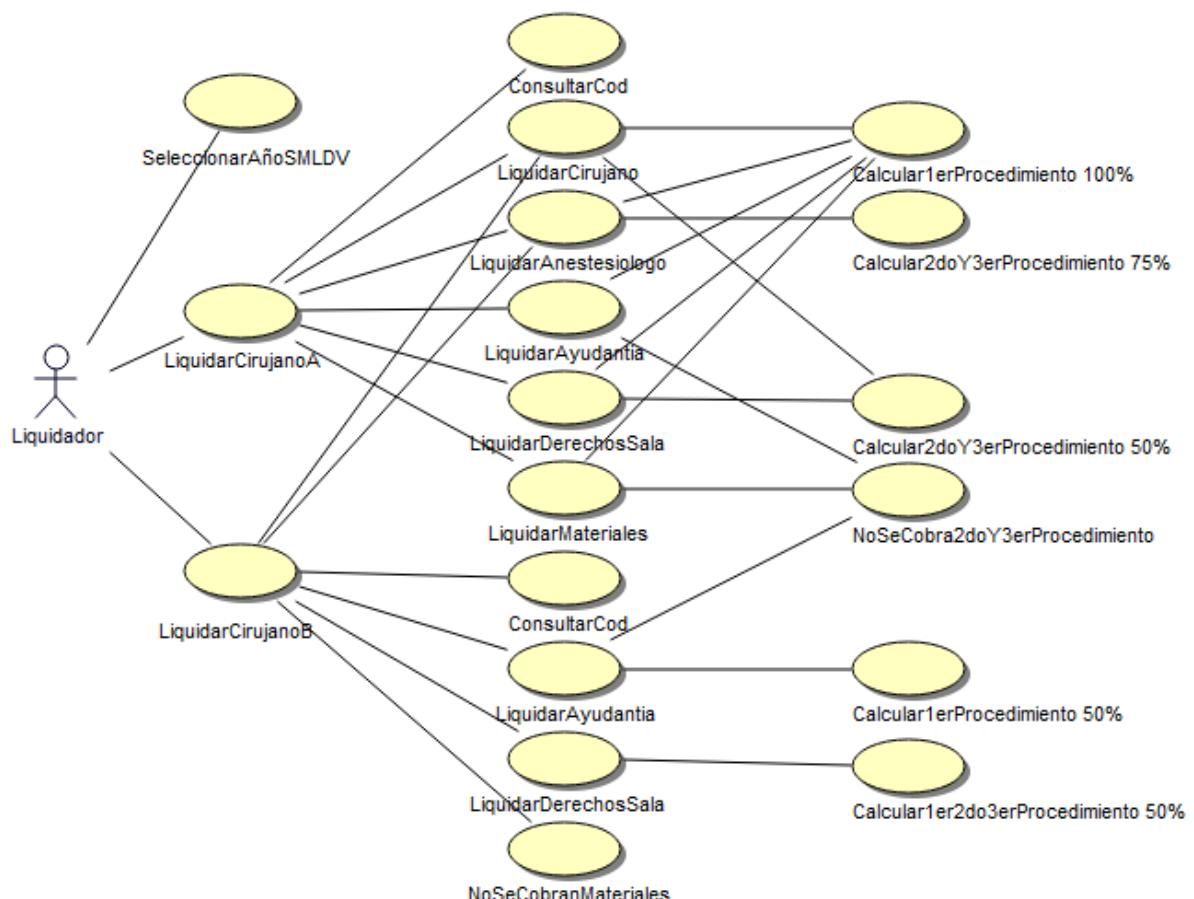


Figura 19. Caso de uso para liquidar procedimientos con la misma vía de acceso y diferente especialista en manual tarifario SOAT (Elaboración propia con herramienta CASE Bouml)

Finalmente, El diagrama de la figura 20, representa la forma en que se debe liquidar los procedimientos con diferente vía de acceso y diferente especialista en el manual tarifario SOAT, iniciando la selección del año, la consulta de cada procedimiento y posteriormente se realiza la liquidación en SMLDV del cirujano, el anestesiólogo, el ayudante, los derechos de sala y los materiales al 100%. Para el segundo y tercer procedimiento los porcentajes de liquidación se manejan como se indica en la figura:

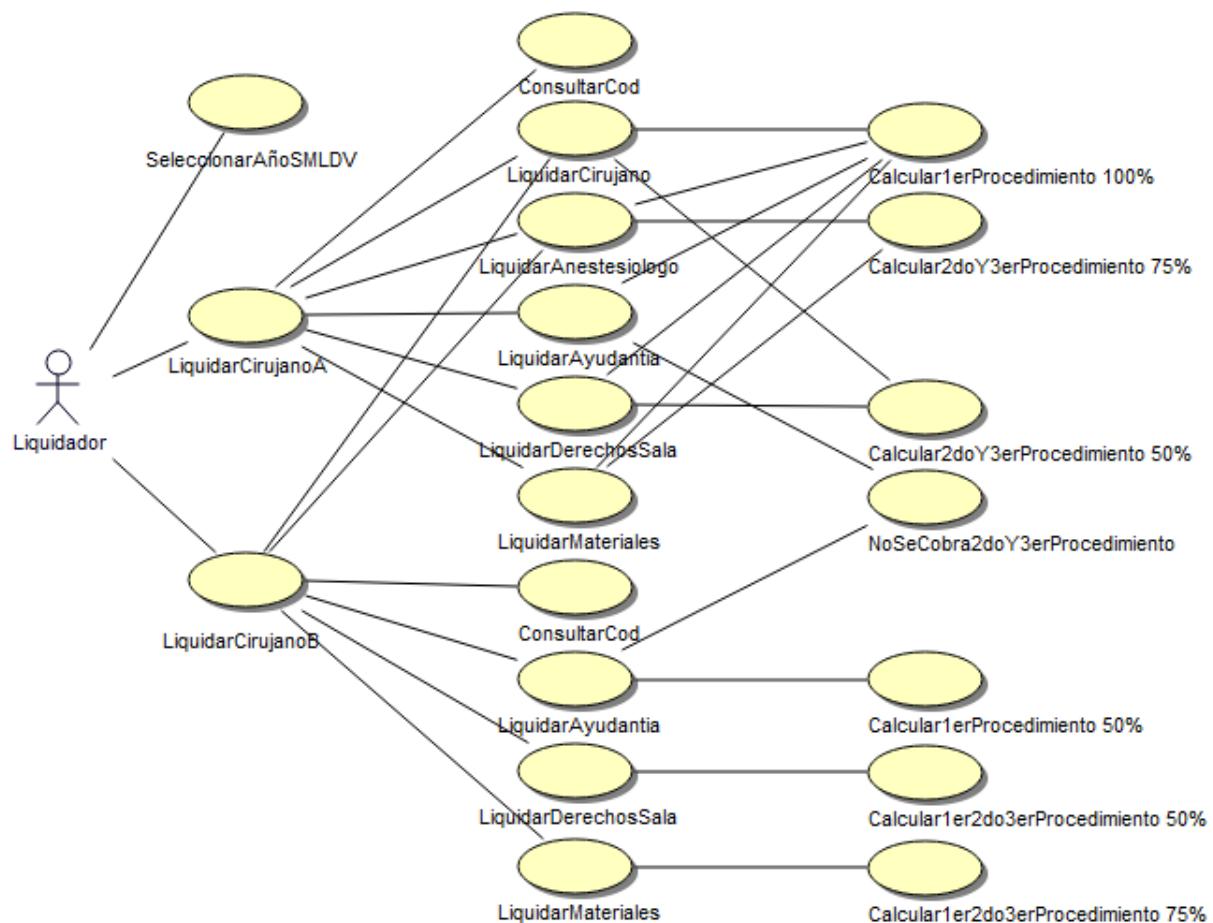


Figura 20. Caso de uso para liquidar procedimientos con diferente vía de acceso y diferente especialista en manual tarifario SOAT (Elaboración propia con herramienta CASE Bouml)

En los apartados siguientes, se justifican las tecnologías utilizadas para llevar a cabo el desarrollo del software.

4.1.2.4. Entorno de Desarrollo Integrado

Según Gámez (2017) “Un IDE es una aplicación de software que proporciona servicios integrales para facilitarle al programador de computadora el desarrollo de software. Generalmente integra un editor de código fuente, algunas herramientas automáticas y un depurador.” (p.7)

La selección del entorno de desarrollo depende del lenguaje y del tipo de aplicación que se va a implementar. Para este caso específico, al ser una aplicación web, se presentan las siguientes opciones que fueron las analizadas para su uso, por ser multiplataforma y de código abierto:

Eclipse: Creado originalmente por IBM, es una plataforma de desarrollo ampliamente utilizada, permite ser personalizada con la instalación de complementos o plug-in para adaptarse a las necesidades del programador. Más información y descargas en <https://eclipse.org/>

NetBeans: Sun MicroSystems fundó el proyecto y es un entorno de desarrollo con un gran número de módulos que permiten extender sus funcionalidades. Más información y descargas en https://netbeans.org/index_es.html

Sublime Text 3: Esta herramienta CASE, se encuentra en la clasificación de editor de texto, tiene una versión de prueba que se puede utilizar de forma indefinida, es multiplataforma y personalizable y cuenta con una amplia comunidad de desarrolladores y complementos. Más información y descargas en <https://www.sublimetext.com/>

Atom: Desarrollado por GitHub, es otro editor de texto con el control de versiones Git por defecto, multiplataforma, personalizable y la mayor parte de sus complementos son de software libre. Más información y descargas en <https://atom.io/>

Brackets: Este editor de texto, desarrollado por Adobe, es creado para el diseño y desarrollo web, está enfocado en HTML, CSS y JavaScript, con complementos para otros lenguajes. Presenta una funcionalidad que permite ver los cambios en el navegador sin necesidad de recargar la página. Más información y descargas en <http://brackets.io/>

De las herramientas disponibles para codificar la solución, se opta por el editor de texto Brackets, por ser un editor orientado al desarrollo web, liviano, personalizable y las opciones que tiene son suficientes para cumplir con los requerimientos del proyecto.

4.1.2.4. Servidor web y de base de datos

Un servidor web es un componente importante de la arquitectura de Internet, son los que permiten compartir información entre todos los usuarios en Internet o en una Intranet.

Un servidor web es un programa instalado en una máquina, que se especializa en escuchar peticiones de un cliente típicamente realizadas a través de un navegador web, para acceder a cierta información. Se encarga de responder al cliente con páginas web, mediante el protocolo HTTP (Hypertext Transfer Protocol). (Morente, 2017, p.1)

Para el servidor web se implementa un entorno de desarrollo local y se contratan los servicios de un proveedor de hosting para el entorno de producción. Debe cumplir con los requerimientos de un servidor Apache, soporte al lenguaje PHP versión 5 o superior, un servidor de base de datos MySQL y servidor web. Es posible instalar y configurar cada una de estas opciones por separado, pero actualmente existen proyectos que ya reúnen todas estas características en un solo programa con una interfaz de fácil administración. Se presentan las siguientes opciones:

XAMPP: Es un entorno de desarrollo para PHP, Apache Friends es el proyecto del servidor web Apache que contiene MariaDB, PHP y Perl. Cuenta con una comunidad con más de 10 años que dan soporte al proyecto. Más información y descargas en <https://www.apachefriends.org/>

WampServer: Es un entorno de desarrollo web de Windows. Permite crear aplicaciones web con Apache2, PHP y una base de datos MySQL. Más información y descargas en <http://www.wampserver.com/>

EasyPHP: Devserver es un entorno de desarrollo portátil, modular, totalmente configurable y fácil de actualizar y ampliar. Soporta PHP 7.x / 5.5.x / 5.4.x / 5.3.x, Apache, Nginx, MySQL, PHPMyAdmin y Xdebug. Más información y descargas en <http://www.easypht.org/>

Las tres opciones cumplen con las características necesarias para el desarrollo de la aplicación web para liquidar los procedimientos médicos, se opta por el servidor web local EasyPHP Devserver, por tener la opción de ser portable y puede desarrollarse en cualquier lugar sin necesidad de exportar las bases de datos, adicionalmente permite configurar un servidor local con las mismas características del servidor de producción.

La versión DevServer de EasyPHP no está diseñada para ser utilizada como servidor de producción, sino como un servidor de desarrollo para probar las aplicaciones web sin

necesidad de conexión a internet y posteriormente ser publicados en un servidor de alojamiento web.

Para su instalación se descarga el archivo ejecutable de la página web del proyecto <http://www.easyphp.org/>. Para este caso se selecciona Devserver V17.0. Se instala y al finalizar se inicia los servidores HTTP y de base de datos como se muestra en la siguiente imagen:

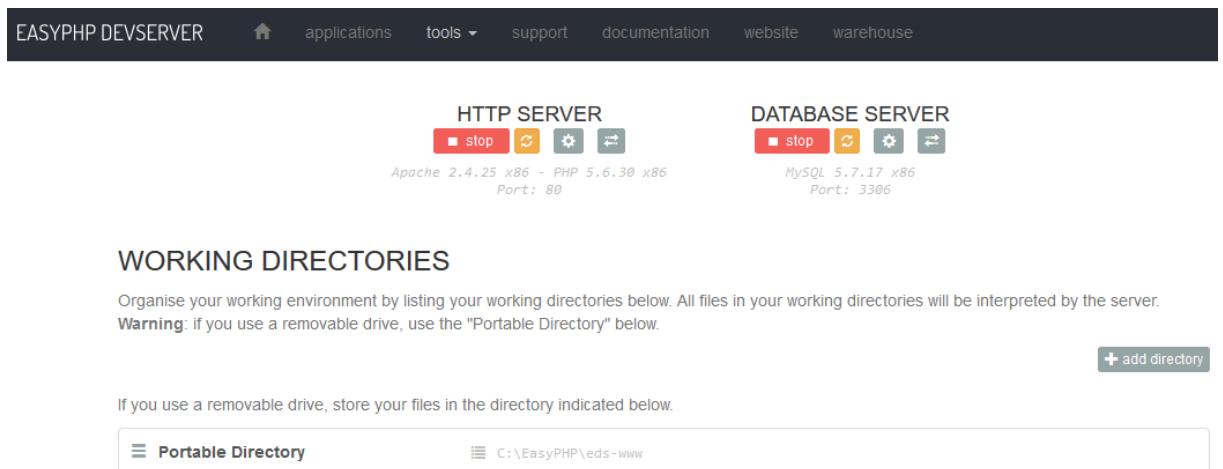


Figura 21. Panel de control EasyPHP Devserver (Elaboración propia)

Para publicar la solución en internet, se contratan los servicios de hosting del proveedor hostinger, <https://www.hostinger.co/>, por cumplir con las características mínimas necesarias para el correcto funcionamiento y una buena relación coste-beneficio.

4.1.2.5. Implementando la base de datos

Según Silberschatz, Korth y Sudarshan (2006) “Un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos” (p.1)

Los procedimientos contemplados en los tres manuales tarifarios tienen características comunes, que pueden ser almacenados en una base de datos para su posterior recuperación. Reunir en una única tabla todos los procedimientos permite desarrollar métodos comunes de consulta en lugar de métodos individuales. De las tablas 1, 11 y 18, de los manuales tarifarios ISS 2001, ISS 2004 Y SOAT respectivamente, se extrae la estructura de la base de datos que almacena la información.

Un modelo de datos ampliamente utilizado actualmente es el modelo relacional, que utiliza tablas para almacenar los datos y representar las relaciones entre ellos.

Una base de datos relacional consiste en un conjunto de tablas, a cada una de las cuales se le asigna un nombre exclusivo. Cada fila de la tabla representa una relación entre un conjunto de valores. De manera informal, cada tabla es un conjunto de entidades, y cada fila es una entidad [...] Dado que cada tabla es un conjunto de tales relaciones, hay una fuerte correspondencia entre el concepto de tabla y el concepto matemático de relación, del que toma su nombre el modelo de datos relacional. (Silberschatz, 2006, p. 29)

En la siguiente figura, se presenta la tabla *cuenta* donde las cabeceras de las tres columnas son sus atributos, que tienen un dominio y son el conjunto de valores que pueden tomar. De esta forma, el saldo solo puede ser de tipo número y deben consistir en una tupla junto con el número de cuenta y el nombre de la sucursal.

<i>número_cuenta</i>	<i>nombre_sucursal</i>	<i>saldo</i>
C-101	Centro	500
C-102	Navacerrada	400
C-201	Galapagar	900
C-215	Becerril	700
C-217	Galapagar	750
C-222	Moralzarzal	700
C-305	Collado Mediano	350

Figura 22. La relación *cuenta* (Silberschatz, 2006, p. 30)

Nótese que se utilizan los términos relación y tupla en lugar de tabla y fila, es decir la relación *cuenta* está conformada por siete tuplas.

Para la implementación de la base de datos, se utiliza el sistema de gestión de bases de datos relacional MySQL, desarrollada por MySQL AB y comprada en 2010 por Oracle Corporation:

MySQL es la base de datos número 1 para las aplicaciones basadas en la web, utilizada por Facebook, Twitter, LinkedIn, Yahoo!, Amazon Web Services y todas las propiedades web más importantes y los inicios exitosos de manera virtual. Con grandes volúmenes de datos, controlados por las redes sociales, la conexión de banda ancha móvil de alta velocidad, los dispositivos inteligentes y las nuevas interacciones de máquina a máquina (Oracle, 2010, p.3)

Se selecciona MySQL por ser de código abierto, “se diseñó y se optimizó para aplicaciones web. [...] MySQL se convirtió en la plataforma elegida por los desarrolladores web y la base de datos predeterminada para las aplicaciones basadas en la web” (Oracle, 2010, p.4) y soportar grandes volúmenes de datos, garantizando la integridad y disponibilidad de la información.

En cuanto al volumen de datos, los tres manuales tarifarios cuentan con 14.786 procedimientos que se extraen directamente de los decretos y acuerdos donde fueron publicados. Aunque en comparación con los billones de datos que está en capacidad de manejar MySQL no resulta ser una cifra significativa, ya es un volumen considerable de información para realizar búsquedas manuales por parte de un usuario.

Otro aspecto que se debe evitar es la duplicidad en la información, es decir que dos tuplas de una misma relación tengan los mismos valores en todos sus atributos. Por ejemplo, un procedimiento puede estar contemplado en los tres manuales tarifarios, pero tienen valores diferentes para ser liquidados, como se ve en la siguiente tabla:

Tabla 27 Comparativa tuplas manuales tarifarios

TARIFARIO	CÓDIGO	DESCRIPCIÓN	UNID DE LIQ.	UNIDADES FIJAS	AÑO
SOAT	7730	Apendicectomía	SMLDV	NO	TODOS
ISS	C40403	APENDICECTOMÍA (392)	UVR	8504	2004
ISS	C40403	APENDICECTOMÍA (392)	PESOS	787560	2001

Fuente: Elaboración propia, a partir de los manuales tarifarios

Para evitar la duplicidad entre tuplas en cada relación se asigna una clave primaria. “Se usará el término clave primaria para denotar una clave candidata que ha elegido el diseñador de la base de datos como medio principal para la identificación de las tuplas de una relación” (Silberschatz, 2006, p. 34)

De acuerdo con el modelo relacional, se implementa la base de datos con tres relaciones, a cada una se le asigna un nombre exclusivo y para cada cabecera o atributo, hay un conjunto de valores permitidos. Las tres relaciones son pacientes, parametros_tarifarios y liquidador_tarifario.

Para realizar la administración de bases de datos desde la web, e implementar estas tres relaciones en el servidor de base de datos MySQL, se hace uso de la herramienta PHPMyAdmin. “La aplicación PHPMyAdmin no es más que un conjunto de páginas escritas

en PHP y que son copiadas directamente en el directorio que aloja las páginas web del servidor." (Cobo, 2005, p. 390)

Con esta aplicación se realiza la administración desde un entorno gráfico, de una forma intuitiva y fácil de usar, además permite:

Crear y eliminar bases de datos. Crear, copiar, eliminar, renombrar y alternar tablas. Eliminar, editar y agregar campos, ejecutar sentencias SQL, administrar campos claves, cargar archivos de textos a las tablas, exportar los datos a diferentes formatos, administrar usuarios MySQL y privilegios, administrar múltiples servidores, crear archivos PDF del diseño de la base de datos, búsquedas globales. (Ferrer, 2014, p. 59)

En la siguiente figura, se presenta la primera relación, pacientes, donde se asigna la clave primaria id y otros atributos como nombres y apellidos, identificación, entre otros. El motivo de incluir esta relación es para asociar un paciente a la liquidación de un procedimiento quirúrgico. A continuación, se muestra el modelo relacional con su respectiva implementación en PHPMyAdmin:

#	Nombre	Tipo	Cotejamiento
1	id	double	
2	primer_nombre	varchar(100)	utf8_spanish_ci
3	segundo_nombre	varchar(100)	utf8_spanish_ci
4	primer_apellido	varchar(100)	utf8_spanish_ci
5	segundo_apellido	varchar(100)	utf8_spanish_ci
6	identificacion	varchar(30)	utf8_spanish_ci
7	direccion	varchar(200)	utf8_spanish_ci
8	telefono	varchar(30)	utf8_spanish_ci
9	localizacion_telefono	varchar(255)	utf8_spanish_ci
10	ciudad	int(11)	
11	nacimiento	date	

Figura 23. La relación pacientes (Elaboración propia)

La segunda relación, parámetros_tarifarios, sirve para almacenar el valor del SMLV y el SMDLV de los últimos años. Esta relación debe ser actualizada con una nueva tupla cada año

con el decreto donde se publica el valor del salario mínimo en Colombia. El ultimo valor registrado corresponde al año 2017. En la siguiente figura se observa modelo relacional con su respectiva implementación en PHPMyAdmin:

#	Nombre	Tipo	Cotejamiento
1	id	int(11)	
2	anual	varchar(45)	utf8_unicode_ci
3	valor_uvr	varchar(45)	utf8_unicode_ci
4	valor_smdlv	varchar(45)	utf8_unicode_ci
5	valor_smmlv	varchar(45)	utf8_unicode_ci

Figura 24. La relación *parametros_tarifarios* (Elaboración propia)

En la tercera relación, *liquidador_tarifario*, se almacenan los 14786 procedimientos que componen los tres manuales tarifarios. Como clave primaria se establece *idliquidador_tarifario* y contiene todos los atributos estudiados en los requerimientos de cada tarifario. En la siguiente figura se observa modelo relacional con su respectiva implementación en PHPMyAdmin:

#	Nombre	Tipo	Cotejamiento
1	idliquidador_tarifario	int(11)	
2	tarifario	varchar(45)	utf8_spanish_ci
3	referencia	varchar(45)	utf8_spanish_ci
4	codigo	varchar(45)	utf8_spanish_ci
5	tipo	longtext	utf8_spanish_ci
6	descripcion	longtext	utf8_spanish_ci
7	grupo_quirurgico	varchar(45)	utf8_spanish_ci
8	unidad_liquidacion	varchar(45)	utf8_spanish_ci
9	unidad_fija	varchar(45)	utf8_spanish_ci
10	especialista	varchar(30)	utf8_spanish_ci
11	anestesiologo	varchar(30)	utf8_spanish_ci
12	ayudante	varchar(30)	utf8_spanish_ci
13	med_general_odontologo	varchar(30)	utf8_spanish_ci
14	anio	varchar(30)	utf8_spanish_ci

Figura 25. La relación *liquidador_tarifario* (Elaboración propia)

4.1.2.6. Lenguajes de programación

El lenguaje de programación para codificar la solución del lado del servidor es PHP. Este lenguaje es utilizado para sitios y aplicaciones web, script de línea de comandos y aplicaciones de escritorio. Cuando se utiliza PHP para la web se necesita de un servidor y un navegador, es multiplataforma (Linux MacOS y Windows) o también se puede utilizar un servicio de hosting para subir los scripts y ver los resultados en un navegador accediendo a través de internet.

PHP para scripts de línea de comandos no utiliza de servidor web ni navegador, necesita un ejecutable de línea de comandos para realizar tareas automatizadas con los scripts. En el caso de PHP para aplicaciones de escritorio (GUI) es necesaria la extensión PHP-GTK y no genera ningún código HTML.

Otra ventaja que tiene este lenguaje de programación es que permite utilizar la técnica de programación orientada a objetos (POO). La POO “se basa en la idea natural de la existencia de un mundo lleno de objetos, con características que los diferencian (atributos), y con un conjunto de acciones propias que pueden realizarse sobre ellos (operaciones)”. (Sicilia, 2017, p.3)

```
<?php
    class ClaseEjemplo{
        //Aquí van las propiedades
        public $propiedad1 = "Primera propiedad";
        public $propiedad2 = "Segunda propiedad";

        //Aquí van los métodos
        public function getPropiedad1(){
            return $this-> propiedad1;
        }
    }
?>
```

Figura 26. Clases y métodos en PHP (Padilla, 2017, p.16)

PHP permite crear los elementos de la POO como son las clases y métodos. “Una clase es un modelo que se utiliza para crear objetos que comparten un mismo comportamiento, estado e identidad.” (Bahit, 2011, p.12). La forma de definir una clase en PHP es utilizando la “palabra reservada class, seguida del nombre que vamos a dar a dicha clase y un conjunto de

variables, constantes y funciones, que estarán contenidas entre un par de llaves. A las variables y constantes las llamaremos propiedades; a las funciones, métodos". (Padilla, 2017, p.16) Por su parte, Bahit (2011) define un método como "el algoritmo asociado a un objeto que indica la capacidad de lo que éste puede hacer." (p.12), como se ve en la figura 26.

Otro elemento es un Objeto, que es "una entidad provista de métodos o mensajes a los cuales responde (comportamiento); atributos con valores concretos (estado); y propiedades (identidad)". (Bahit, 2011, p.12) En PHP" para crear una instancia (objeto) de una clase, utilizaremos la palabra reservada new" (Padilla, 2017, p.17), como se aprecia en la siguiente imagen:

```
<?php
    $objetoEjemplo = new ClaseEjemplo();
?>
```

Figura 27. Clases en PHP (Padilla, 2017, p.17)

Los eventos y mensajes se pueden definir según Bahit (2011) así: "Un evento es un suceso en el sistema mientras que un mensaje es la comunicación del suceso dirigida al objeto." y las propiedades y atributos como: "Las propiedades y atributos, son variables que contienen datos asociados a un objeto. \$nombre = 'Juan'; \$edad = '25 años'; \$altura = '1,75 mts';" (p.12)

En el lado del cliente, el acceso a la aplicación se realiza a través de internet y un navegador web, haciendo uso del protocolo HTTP y la información se muestra en código HTML al usuario. "HTTP es un protocolo de pregunta/respuesta basada en un modelo cliente-servidor donde, [...] el navegador web envía un mensaje de petición al servidor web y el servidor que contiene dicho recurso envía el mensaje de respuesta." (Padilla, 2017, p.9)

Para presentar la información de forma elegante y estética se utilizan hojas de estilo, donde a las etiquetas HTML se le asignan características como el color de fondo o el tamaño de la letra:

La principal función que tiene el uso de las hojas de estilo es separar el contenido de nuestra web de la forma en que dicho contenido se presenta. Esto implica, por ejemplo, que el estilo puede quedar definido en un fichero independiente del HTML del contenido, consiguiendo reutilización del mismo y permitiendo modificar la forma de visualización de nuestro contenido, únicamente cambiando el fichero de estilo que tiene asignado. Por otra parte, al igual que los ficheros HTML, los ficheros de estilo son ficheros de texto plano, con extensión .css, concretamente. (Padilla, 2017, p.22)

Un ejemplo de un estilo de color de fondo rojo aplicado a la etiqueta `<h1>` se muestra en la siguiente figura:

```

<head>
    <title> Página con estilo incrustado </title>
    <style type="text/css">
        h1{
            background-color: red;
        }
    </style>
</head>

```

Figura 28. Ejemplo hoja de estilos en cascada CSS (Padilla, 2017, p.25)

Para programar la interactividad de la aplicación y realizar las validaciones de la información que alimenta el usuario al sistema, se utiliza JavaScript. Este lenguaje usa scripts que “son pequeños trozos de código que se pueden ejecutar en los navegadores y que nos van a permitir incluir el contenido dinámico en nuestra web.” (Padilla, 2017, p.2)

Cuando un creador incorpora JavaScript a su página, proporciona al usuario cierta capacidad de interacción con la página web, es decir, cierto dinamismo y por lo tanto se incrementan las prestaciones de la misma al añadir procesos en respuesta a las acciones del usuario. Es importante señalar que estos procesos se ejecutan en la máquina del cliente (en el navegador) y por tanto no implican intercambio de datos con el servidor. Con JavaScript se accede al mundo de las web dinámicas. (Mohedano, 2012, p.9)

```

<html>
<head>

    <title> Ejemplo prompt </title>

</head>
<body>

    <script type="text/javascript"> //OJO: nota al pie
        tuNombre=prompt("Un saludo. ¿Cómo te llamas?", "");
        alert("Hola " + tuNombre + ", encantado de conocerte");
    </script>

</body>
</html>

```

Figura 29. Ejemplo de un Script en JavaScript (Padilla, 2017, p.10)

Los scripts se ejecutan del lado del servidor y del lado del cliente. En el lado del servidor se ha manejado PHP y del lado del cliente se emplea JavaScript, que “es un lenguaje de

programación que se utiliza principalmente para crear páginas web capaces de interactuar con el usuario. (Mohedano, 2012, p.9).

En la figura 29, se presenta un script donde se le solicita al usuario que digite el nombre por medio de la función prompt, se guarda el resultado en la variable tuNombre y posteriormente se muestra el resultado en un alert:

JavaScript es un lenguaje interpretado en el navegador web, sus segmentos de código se escriben en un editor de texto plano y son embebidos en ficheros HTML, como se muestra en la siguiente figura:

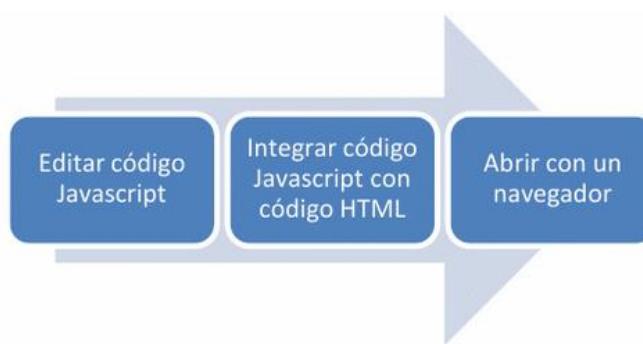


Figura 30. Proceso de edición y ejecución de un programa JavaScript (Mohedano, 2012, p.11)

Existe dos formas de integrar el código JavaScript con el código HTML. La primera forma es directamente en la cabecera del documento HTML y la segunda es a través de un archivo externo con la extensión .js. Los dos ejemplos se muestran en las siguientes figuras, en el segundo caso, el nombre del archivo externo es mensaje.js:

```

<html>
<head>
    <title>Ejemplo de código JavaScript en el propio documento</title>
    <script type="text/javascript">
        alert("Esto es mensaje de prueba");
    </script>
</head>
<body>
    <p>Y esto un párrafo de texto.</p>
</body>
</html>
  
```

Figura 31. Ejemplo de código en el propio documento HTML (Ribest, 2011, p.16)

```
<html>
  <head>
    <title>Ejemplo de código JavaScript en otro documento</title>
    <script type="text/javascript" src="mensaje.js"></script>
  </head>
  <body>
    <p>Y esto un párrafo de texto.</p>
  </body>
</html>
```

Archivo mensaje.js

```
alert("Esto es un mensaje de prueba");
```

Figura 32. Ejemplo de código en documento aparte (Ribest, 2011, p.17)

Para agilizar la escritura del código JavaScript, se utiliza la librería jQuery, que además de manejar selectores y sentencias más cortas, presenta ventajas como manipulación sencilla de los elementos de DOM, compatibilidad entre los diferentes navegadores, manejo de comunicación asíncrona con métodos AJAX de fácil configuración, incorporación de complementos para autocompletado de campos de búsqueda y manipulación de texto plano con formato JSON. Como lo menciona Vara (2014) jQuery es:

Una librería JavaScript del año 2006 con licencia MIT y GPL, por lo que podemos utilizarla tanto en entornos libres como de software comercial (Bibeault, 2010). A partir de la librería podemos acceder al árbol DOM de una manera más amigable, así como el manejo del contenido y eventos de la página, crear efectos visuales o modificar el CSS. De igual forma, puede trabajar con JSON y contiene componentes visuales como cuadros de diálogo, paneles colapsables o calendarios. (p.193)

4.1.2.7. Implementando el patrón arquitectónico Modelo Vista Controlador MVC en PHP

Para implementar el patrón de diseño arquitectónico MVC se va a realizar la codificación de un framework básico. Un marco o framework es “un conjunto de clases que constituye una aplicación incompleta y genérica. Si el marco se complementa de manera adecuada (si se ‘especializa’), se obtienen aplicaciones especializadas de un cierto tipo.” (Campderrick, 2003, p.193).

Según Campderrich (2003), entre la ventajas se encuentran:

- a) Reducen el trabajo de programación y mantenimiento de aplicaciones. Los marcos reducen la codificación y la puesta a punto, ya que proporcionan subsistemas que sabemos que funcionan. En definitiva, suministran código que ya no se deberá volver a escribir ni a mantener.
- b) Proporcionan una arquitectura para el software.
- c) Llevan a desarrollar pequeñas aplicaciones que encajan dentro de los marcos, en lugar de aplicaciones monolíticas.
- d) Son una buena base para la industria de componentes de software. Los marcos bien diseñados permiten que terceras compañías puedan suministrar componentes o partes de componentes que los desarrolladores podrán añadir. (p.193)

En el mercado se encuentran una gran cantidad de frameworks en PHP listos para utilizar, como Laravel, Yii, CodeIgniter, Symfony, Phalcon, Zend Framework 2, entre otros, pero teniendo en cuenta la dimensión de la aplicación a desarrollar y con el fin de poner en práctica los conocimientos adquiridos, se opta por codificar un framework básico con el patrón MVC.

Se inicia con la configuración del servidor Apache para que realice determinados comportamientos mediante el archivo .htaccess (Hypertext Access) para que permita el acceso y llamado a controladores y métodos, al igual que el envío de parámetros a través de la URL:

Archivo .htaccess

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-
RewriteRule ^(.+)\$ index.php?url=\$1 [QSA,L]
```

Para verificar su funcionamiento en el archivo index.php se digita:

```
<?php
print_r( $_GET['url'] );
?>
```

Y en el navegador web se obtiene el siguiente resultado al invocar un controlador, un método y enviar dos parámetros vía get:

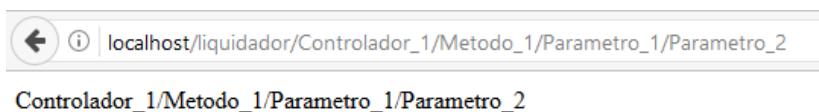


Figura 33. Ejemplo funcionamiento archivo .htaccess (Elaboración propia)

Para continuar con la implementación del framework básico MVC se propone la siguiente estructura de archivos donde se puede apreciar el archivo .htaccess e index.php en el folder raíz de la aplicación:

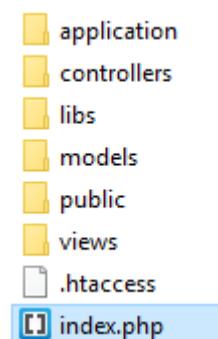


Figura 34. Estructura de archivos patrón MVC (Elaboración propia)

En el folder application es donde va el código propio de la aplicación y se encuentra el archivo de configuración, los archivos con las clases de conexión a la base de datos y los archivos con las clases de donde van a heredar los modelos, vistas y controladores que se creen para realizar la codificación de la solución final, al igual que el archivo que maneja y redirige las peticiones realizadas por los usuarios.

Los folders controllers, models y views, contienen los controladores para procesar las peticiones, los modelos para acceder a los datos y las vistas con las plantillas o templates HTML para interactuar con el usuario.

El folder libs se utiliza para guardar librerías desarrolladas por terceros, como por ejemplo una librería que permita exportar datos en formato de Excel o pdf.

Finalmente, el folder public se utiliza para los archivos de estilos, imágenes, scripts y demás archivos necesarios para el funcionamiento del aplicativo web.

Iniciando con el archivo de configuración, Config.php, se definen las características principales como la ruta base de la URL, el controlador y la vista por defecto, datos de la aplicación y acceso a la base de datos, como el usuario, contraseña, que son esenciales en el momento de migrar la aplicación a otro servidor web. En la siguiente figura se muestra el contenido del archivo Config.php:

```

12 define('BASE_URL', 'http://localhost/liquidador/');
13 define('DEFAULT_CONTROLLER', 'index');
14 define('DEFAULT_LAYOUT', 'default');
15
16 define('APP_NAME', 'Liquidador de Cirugias');
17 define('APP_SLOGAN', '');
18 define('APP_COMPANY', 'http://tarificador.hol.es/');
19
20 define('DB_HOST', 'localhost');
21 define('DB_USER', 'root');
22 define('DB_PASS', '');
23 define('DB_NAME', 'tarifario');
24 define('DB_CHAR', 'utf8');
--
```

Figura 35. Archivo Config.php (Elaboración propia con herramienta CASE Brackets)

En la siguiente figura, se puede apreciar la codificación del archivo index.php donde se define la ruta del sistema y se realiza la llamada a todos los archivos que conforman la aplicación.

```

1 <?php
2
3 define('DS', DIRECTORY_SEPARATOR); //Separador del S.O. ej: win= \ linux= /
4 define('ROOT', realpath(dirname(__FILE__)) . DS); //Ruta donde esta guardado el index de la aplicación
5 define('APP_PATH', ROOT . 'application' . DS); //Directorio de las aplicaciones.
6
7 require_once APP_PATH . 'Config.php'; //Variables de configuración. Se puede hacer un solo archivo o crear uno para las rutas, otro para
//las bases de datos y otro para las configuraciones de la aplicación.
8 require_once APP_PATH . 'Request.php'; //Recibe las peticiones por la URL y las pasa al Bootstrap.php
9 require_once APP_PATH . 'Bootstrap.php'; //Llama al controlador que se encuentre en controllers/ y que es pedido en el Request.php.
10 require_once APP_PATH . 'Controller.php'; //Es el controlador principal, de donde se van a extender todos los controladores. Contiene los
//métodos que va a ser utilizados por todos controladores.
11
12 require_once APP_PATH . 'Model.php'; //controllers/ -> El controlador que reciba la petición para su gestión de modelo y vista.
13 require_once APP_PATH . 'View.php'; //Contiene todos los métodos que van a ser comunes a todos los modelos.
14 require_once APP_PATH . 'Database.php'; //Contiene todos los métodos que van a ser comunes a todas las vistas. Las vistas no son instanciadas,
//la clase view maneja el trabajo con las vistas.
15 //Connexion a la base de datos.
16 ?>
17 try{
18     Bootstrap::run(new Request); //Llama el Bootstrap y request para que procesar la peticion.
19 } catch(Exception $e){
20     echo $e->getMessage(); //Si no encuentra un controlador, muestra mensaje de error devuelto por el bootstrap: throw new
Exception('no encontrado');
21 }
22 ?>
```

Figura 36. Archivo index.php (Elaboración propia con herramienta CASE Brackets)

A continuación, se codifica el archivo Request.php que es el encargado de recibir las peticiones por la URL y las pasa al archivo Bootstrap.php. El archivo Request.php Contiene la clase Request, que en su constructor utiliza el método explode('/', \$url) para crear un arreglo separando la URL del dominio para extraer el controlador, método, argumentos y almacenarlos en las variables privadas \$_controlador, \$_metodo y \$_argumentos. Sus métodos públicos getControlador(), getMetodo() y getArgs() son utilizados por las clases hijas para recuperar los valores de dichas variables.

El archivo Bootstrap.php verifica si existe el archivo del controlador que se encuentre en controllers/ y que es pedido en el archivo Request.php. También guarda el método y los argumentos, en caso de no existir estos parámetros en la URL asigna por defecto el controlador index.

Por su parte el archivo Controller.php, es el controlador principal, de donde se heredan todos los controladores, contiene la clase abstracta Controller donde se obtiene el objeto View que se encuentra disponible en la clase Request, luego se crean los métodos para importar los modelos, librerías y redireccionar en caso de ser necesario.

El archivo View.php contiene la clase View, a diferencia de los controladores, la vista no es necesario que hereden de esta clase, pero si se debe tener un objeto que maneje el trabajo con las vistas. El método renderizar es el encargado de realizar esta tarea, muestra el menú de opciones de la página y recibe como parámetro el nombre de la vista solicitada a través de la URL.

Como parte del template por defecto, se crean dos archivos que son header.php y footer.php, que contienen la estructura HTML que es común a todas las vistas de la aplicación, como el llamado a los archivos que contienen las hojas de estilos en cascada, los scripts de JavaScript y el menú.

Para el acceso a la base de datos se crea el archivo Database.php, que contiene la clase Database que hereda de PDO, por sus siglas en inglés PHP Data Objects y “proporciona una capa de abstracción de acceso a datos, lo que significa que, independientemente de la base de datos que se esté utilizando, se emplean las mismas funciones para realizar consultas y obtener datos.” (PHP Group, 2001-2017). En la siguiente imagen se muestra el código para realizar la conexión con la base de datos:

```

11 ▼ class Database extends PDO {
12
13 ▼   public function __construct() {
14     parent::__construct(
15       'mysql:host=' . DB_HOST .
16       ';dbname=' . DB_NAME,
17       DB_USER,
18       DB_PASS, array(
19         PDO::MYSQL_ATTR_INIT_COMMAND => 'SET NAMES ' . DB_CHAR
20       ));
21   }
22 }
```

Figura 37. Archivo Database.php (Elaboración propia con herramienta CASE Brackets)

El archivo Model.php, contiene la clase Model que instancia un objeto de la clase Database para que pueda ser usada para las consultas a la base de datos que realicen los modelos de la aplicación. En la siguiente figura se observa el código implementado:

```

11 ▼ class Model {
12
13     protected $_db;
14
15 ▼     public function __construct() {
16         $this->_db = new Database();
17     }
18 }
```

Figura 38. Archivo Model.php (Elaboración propia con herramienta CASE Brackets)

4.1.2.8. Codificando la solución

Una vez se tiene listo el framework con la estructura MVC, que se encarga de manejar las peticiones y la conexión a la base de datos, se procede a codificar la solución.

En el folder controllers se encuentran los archivos con los controladores de la aplicación, como regla general se crean con el nombre del controlador seguido de la palabra Controller, por ejemplo, para el controlador index se crea el archivo indexController.php que hereda de la clase Controller. En este caso no se invoca un modelo, solamente llama la vista y pasa los parámetros título y el nombre del archivo que se va a renderizar, como se ve en la siguiente figura:

```

11  class indexController extends Controller
12 ▼ {
13     public function __construct() {
14         parent::__construct();
15     }
16
17     public function index()
18     {
19         $this->_view->título = 'Portada';
20         $this->_view->renderizar('index', 'inicio');
21     }
22 }
```

Figura 39. Archivo indexController.php (Elaboración propia con herramienta CASE Brackets)

Como se expone en el capítulo 3.2, el modelo son los datos y las clases que se crean para consultar la base de datos. Para este fin, en el folder Models se encuentra el archivo liquidadorModel.php que hereda de la clase Model y contiene los métodos para autocompletar la consulta en la tabla de pacientes, la tabla de procedimientos médicos según el tarifario seleccionado y la tabla de salarios mínimos para el caso del tarifario SOAT.

```

3 ▼ class liquidadorModel extends Model {
4
5 ▼   public function __construct() {
6     parent::__construct();
7   }
8
9 ►   public function getPaciente($valor) { [...]}
25
26 ►   public function getProcedimiento($valor, $idtarifario, $anio) { [...]}
73
74 ▼   public function getAnual() {
75     $result = $this->_db->query("SELECT anual
76                               FROM parametros_tarifarios
77                               WHERE cancel = 0
78                               ORDER BY anual DESC");
79
80     return $result->fetchall();
81   }
82 ▼   public function getQuirofano($cod_quirofano) {
83     $result = $this->_db->query('SELECT
84                               idliquidador_tarifario,
85                               codigo,
86                               descripcion,
87                               grupo_quirurgico,
88                               suma,
89                               tarifario,
90                               anio,
91                               tipo,
92                               unidad_liquidacion,
93                               med_general_odontologo
94                               FROM
95                               liquidador_tarifario
96                               WHERE

```

Figura 40. Segmento archivo liquidadorModel.php (Elaboración propia con herramienta CASE Brackets)

En el folder views, se crean las vistas que componen la aplicación y se encuentra un folder por cada vista que se necesite. Para la vista index se crea el archivo index.phtml que contiene el código que se muestra en la imagen siguiente y es un segmento de etiquetas HTML para ser embebidas en la plantilla del framework.

```

1 ▼ <dl>
2   
3   <h1>Aplicación web para liquidar procedimientos médicos, quirúrgicos y hospitalarios en Colombia</h1>
4   <h2>Máster universitario en Ingeniería de Software y Sistemas Informáticos</h2>
5   |tarifarios:
6 ▼   <dd>
7     <ul>
8       <li type="circle">SOAT</li>
9       <li type="circle">ISS 256 - 2001</li>
10      <li type="circle">ISS 312 - 2004</li>
11    </ul>
12  </dd>
13 </dl>

```

Figura 41. Archivo index.phtml (Elaboración propia con herramienta CASE Brackets)

Luego de ser invocada por el archivo indexController.php, la vista index.phtml es renderizada y embebida en el resto del template de la aplicación y se presenta al usuario como se ve en la siguiente figura:

Máster universitario en Ingeniería de Software y Sistemas Informáticos

Tarifarios:

- SOAT
- ISS 256 - 2001
- ISS 312 - 2004

Figura 42. Renderizado vista index.phtml (Elaboración propia)

La aplicación se va a dividir en dos secciones, la primera es el inicio donde se proporciona la información general y la segunda es el liquidador de cirugías. Para presentar los elementos en el navegador, se crea una hoja de estilos en cascada CSS contenida en el archivo estilos.css, donde se definen colores de la plantilla, tipos y tamaños de las fuentes, forma y colores de los botones, cuadros de texto, tablas y en general todos los detalles que permiten crear una interfaz agradable a la vista y amigable con el usuario. En la siguiente figura se aprecia un segmento de estos estilos aplicados a elementos como el menú, los logos, entre otros:

```

53 ▼ #main{
54     position: relative;
55     margin: 0 auto 0 auto;
56     width: 900px;
57     padding-top: 20px;
58 }
59
60 ▼ #header{
61     width: 900px;
62     height: 100px;
63 }
64
65 ▼ #logo{
66     width: 270px;
67     float: left;
68     margin: 0;
69     padding: 10px 10px 10px 20px;
70     height: 80px;
71 }

73 ▼ #logo h1{
74     color: #FFFFFF;
75     line-height: 40px;
76     text-shadow: 0 0 0;
77     margin: 0;
78 }
79
80 ▼ #logo-unir{
81     float: left;
82     margin-right: 0.5cm
83 }
84 ▼ #menu{
85     width: 580px;
86     height: 80px;
87     float: left;
88     margin: 0;
89     padding: 10px;
90 }
91
92 ▼ #content{background: #999999;
93     color: #6E6E6E;

```

Figura 43. Segmento archivo estilos.css (Elaboración propia con herramienta CASE Brackets)

Para codificar el contenido dinámico de la aplicación web se utilizan scripts. Según Ribes (2011) “(script se traduce como guion, literalmente) son archivos de órdenes, programas por lo general simples.” (p.15) “Se trata de programas sencillos. Una de las peculiaridades que tienen es que se guardan en archivos de tipo texto.” (p.25).

En concordancia con los requerimientos, el sistema debe ser alimentado por el usuario según el tipo de tarifario a utilizar, la vía de acceso, el especialista y la cantidad de procedimientos, también debe realizar una búsqueda del procedimiento a liquidar. Incluir JavaScript en esta aplicación web, implica poder realizar estas tareas validando la información proporcionada por el cliente.

En primera instancia se solicitan los datos del paciente, su búsqueda se puede realizar por nombre, apellido o número de documento, luego se selecciona el tarifario ISS 2001, ISS 2004 o SOAT. Para el caso del tarifario SOAT se debe escoger el año de liquidación. A continuación, se realiza la búsqueda del procedimiento por código o nombre y finalmente se seleccionan las opciones con que se liquida ese procedimiento.

La captura de pantalla muestra un formulario con los siguientes campos:

- Paciente:** Un cuadro de texto que contiene el valor "1::CHARLES::RICHARD::TORRES::MORENO::79874321".
- Tarifario:** Una lista desplegable que muestra "SOAT".
- Año:** Una lista desplegable que muestra "2017".
- Procedimiento:** Un cuadro de texto que contiene el valor "7730::Apéndicectomía::SOAT:::566".
- Via Acceso:** Una lista desplegable que muestra "Selección..".
- Especialista:** Una lista desplegable que muestra "Selección..".
- Buscar...**: Un botón para iniciar la búsqueda.

Figura 44. Interfaz web Liquidador de Cirugías (Elaboración propia)

Para los campos Paciente y Procedimiento de la interfaz web del Liquidador de Cirugías, que se muestra en la figura anterior, se utiliza la función de autocompletado, de jQuery:

La captura de pantalla muestra un formulario con un campo de texto para "Procedimiento" que contiene "apendi". Debajo de este campo, una lista desplegable muestra los siguientes resultados:

- 3110::Resección apéndice pre auricular::SOAT::251
- 507003:: Apendicectomía apéndice no perforada ::SOAT::3231
- 7730::Apéndicectomía::SOAT:::566** (Este resultado está resaltado con un efecto de fondo azul).
- 9663::Resección de apéndice testicular::SOAT::759

Figura 45. Búsqueda de procedimiento con autocompletado (Elaboración propia)

El proceso se inicia agregando el evento keypress de JavaScript a la caja de texto del procedimiento. Con este evento se invoca el script cada vez que el usuario presione una tecla para recuperar el valor de la cadena de texto a medida que se va escribiendo. El tarifario seleccionado es importante para realizar la búsqueda en la base de datos, este valor se almacena en la variable idtarifario. Ver la siguiente figura.

```

1  /*global $, $, console, alert*/
2  /*jshint eqeq: true*/
3
4  $(function () {
5      'use strict';
6      var BASE_URL, id, dato, c_via_acceso;
7
8      BASE_URL = $('#base_url').val();
9
10     // Autocompletado Pacientes
11    $("#c_paciente").autocomplete({ ... });
12
13    //Autocompletar Procedimiento
14    $('#c_procedimiento').keypress(function () {
15        var idtarifario = $('#c_tarifario').val();
16        $('#c_procedimiento').autocomplete({
17            source: BASE_URL + 'liquidador/busca_procedimiento/?idtarifario=' + idtarifario,
18            select: function (event, ui) {
19                id = (ui.item.value);
20                dato = id.split("::");
21            }
22        });
23    });

```

Figura 46. Script autocompletar procedimiento archivo *liquidarCirugia.js* (Elaboración propia con herramienta CASE Brackets)

Posteriormente, se invoca el controlador (liquidador) encargado de realizar la búsqueda del procedimiento, se pasa en la URL el método (busca_procedimiento) y como parámetro el idtarifario: liquidador/busca_procedimiento/?idtarifario=' + idtarifario.

```

23  public function busca_procedimiento() {
24      $resultados = array();
25      $valor = $_REQUEST['term'];
26      $idtarifario = $_REQUEST['idtarifario'];
27      if ($idtarifario == -1) {           //si NO escogieron manual tarifario
28          $err[] = array(
29              'label' => '...: ERROR - SELECCIONE UN TARIFARIO ::..',
30              'value' => '...: ERROR - SELECCIONE UN TARIFARIO ::..'
31          );
32          $idtarifario = "";
33          $anio = "";
34      }
35      if ($idtarifario == 1) {           //si escogieron manual tarifario SOAT
36          $idtarifario = "SOAT";
37          $anio = 'TODOS';
38      }
39      if ($idtarifario == 2) {           //si escogieron manual tarifario ISS 2001
40          $idtarifario = "ISS";
41          $anio = '2001';
42      }
43      if ($idtarifario == 3) {           //si escogieron manual tarifario ISS 2004
44          $idtarifario = "ISS";
45          $anio = '2004';
46      }
47      if (isset($err)) {
48          echo json_encode($err);       //Retona error o procedimiento encontrado
49      } else {
50          $resultados = $this->liquidador->getProcedimiento($valor, $idtarifario, $anio);
51          echo json_encode($resultados);
52      }
53  }

```

Figura 47. Método *busca_procedimeinto* de *liquidadorController* (Elaboración propia con herramienta CASE Brackets)

El método busca_procedimiento, figura 47, verifica el manual tarifario seleccionado, enviar los datos al modelo getProcedimiento y devuelve a la vista el resultado de la consulta. En caso de no seleccionar un tarifario, no se realiza el llamado al modelo y genera un mensaje de error al usuario. Los datos se retornan a la vista en formato JSON.

JSON “es el acrónimo de JavaScript Object Notation y podemos verlo desde dos puntos de vista. Por un lado, es un formato ligero para el intercambio de datos y, por otro lado, es una manera de almacenar información.” (vara, 2014, p.185). Su sintaxis se basa en el elemento object y “está conformado por un conjunto desordenado de pares nombre/ valor. Un objeto comienza con una llave de apertura y finaliza con una llave de cierre. Cada nombre es seguido por dos puntos, estando los pares nombre/valor separados por una coma” (vara, 2014, p.186) como se muestra en el siguiente ejemplo:

```
{
    "nombreFruta": "Manzana",
    "cantidad": 10,
    "valor": 500
}
```

```
public function getProcedimiento($valor, $idtarifario, $anio) {
    $resultados = array();
    if ($idtarifario == "ISS" && $anio == '2001') {
        $sql2001 = $this->db->query("SELECT idliquidador_tarifario FROM `liquidador_tarifario` WHERE `tarifario` = 'iss' AND `anio` = '2001' AND descripcion LIKE '%$valor%' AND `tipo` IN ('Servicios Intrahospitalarios', 'Proteccion Deteccion y Atencion de Enfermedades' AND `unidad_liquidacion`='uvr')");
        $Select_Consulta1 = $sql2001->fetch();
        $if_iss_2001 = "
            AND idliquidador_tarifario>" . $Select_Consulta1['idliquidador_tarifario'] . "
        ";
    } else {
        $if_iss_2001 = "";
    }
    $res = $this->db->query("select idliquidador_tarifario, codigo, descripcion, tarifario, anio, unidad_liquidacion
        from liquidador_tarifario
        where (codigo like '%$valor%' or descripcion like '%$valor%' or referencia like '%$valor%')
        and (tarifario='$idtarifario' and anio='$anio')
        AND suma <> 0
        $if_iss_2001
        order by codigo
        limit 0, 10");

    if ($idtarifario == "SOAT") {
        $anio = "";
    } else {
        $anio = " " . $anio;
    }
    while ($reg = $res->fetch(PDO::FETCH_ASSOC)) {
        $resultados[] = array(
            'label' => $reg['codigo'] . ":" . $reg['descripcion'] . ":" . $reg['tarifario'] . $anio
            'value' => $reg['codigo'] . ":" . $reg['descripcion'] . ":" . $reg['tarifario'] . $anio
        );
    }
    return $resultados;
}
```

Figura 48. Método getProcedimiento de liquidadorModel (Elaboración propia con herramienta CASE Brackets)

El modelo liquidadorModel, figura 48, recibe tres parámetros en su método getProcedimiento(\$valor, \$idtarifario, \$anio) necesarios para realizar la búsqueda en la base de datos y recuperar el código, la descripción y la unidad de liquidación.

En la codificación de la solución, el uso de AJAX se aplica en procesos como en la de realizar la búsqueda de un procedimiento. La petición por parte del cliente se inicia al pulsar el botón Buscar que invoca la función BuscarServicios() de JavaScript.

```

65 ▼ function BuscarServicios(){
66     jQuery.ajax({
67         url: 'liquidador/tabla_procedimiento/',
68         type: 'GET',
69         data: ({
70             Dato:idprocedimiento[1],
71             Tarifario:tarifario,
72             anio:anio
73         }),
74         beforeSend: function(){
75             //Muestra mensaje mientras se ejecuta el proceso.
76             jQuery("#resultado_datos").html("Cargando...");
77         },
78         success: function(data){
79             jQuery('#resultado_datos').html("");
80             jQuery('#Div_Servicios').html(data);
81
82         }/*data*/
83     });
}

```

Figura 49. Uso de AJAX en la búsqueda de procedimientos (Elaboración propia con herramienta CASE Brackets)

La función AJAX de jQuery, invoca el controlador liquidador y el método tabla_procedimiento. Como parámetros se envían el id del procedimiento, el tarifario y el año.

```

23 ▼     public function tabla_procedimiento() {
24
25         $this->_view->dato_servicio = $this->_liquidador->getServicio();
26
27         $this->_view->renderizar_no_menu('servicio');
28     }

```

Figura 50. Método tabla_procedimiento del controlador liquidadorController (Elaboración propia con herramienta CASE Brackets)

Este método, se encarga de invocar el método getServicio del modelo liquidadorModel y enviar la información a la vista servicio.phtml

Para el caso de adicionar un procedimiento. Se produce un proceso similar. En primer instancia se llama la función Add_Service() de JavaScript y vía AJAX se invoca el método adicionar_procedimiento de liquidadorController, que finalmente llama el método addservicio para pasar los datos a la vista.

Finalmente, se aplica la función para liquidar los procedimientos agregados. En la función Liquidar() de JavaScript se encuentran todas las condiciones necesarias para realizar las liquidaciones según las condiciones especificadas en los tres manuales tarifarios.

4.1.3. Evaluación

Para la evaluación de la aplicación se tienen en cuenta dos aspectos. El primero desde el punto de vista de la usabilidad y el segundo analizando la aplicabilidad de la herramienta para resolver el problema propuesto.

Cuando se habla de usabilidad es importante tener en cuenta la accesibilidad. Una aplicación web accesible es un buen inicio para lograr que sea usable. De acuerdo con Pintos (2014):

La palabra usabilidad no existe en el diccionario, sino que es un anglicismo que en castellano se puede traducir literalmente como “facilidad de uso”. Se refiere a la facilidad con la que se puede utilizar una herramienta, objeto o interfaz web. Aunque ya existía y se aplicaba desde mucho antes para el diseño de herramientas e interfaces en otros ámbitos, se empezó a hablar del término usabilidad con la proliferación de la interacción entre personas y ordenadores. (p. 103)

Por su parte la accesibilidad según Córcoles y Montero (2014) la definen como:

Hablar de accesibilidad web es hablar de un acceso universal a la web, es hablar de que todo el mundo pueda acceder a la web y desde cualquier contexto, independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los usuarios. (p.170)

Para la evaluación inicial de accesibilidad se maneja la herramienta en línea WAVE (web accessibility evaluation tool) disponible en la URL <http://wave.webaim.org/>, donde el principal error encontrado fue el bajo contraste que se manejaba entre el fondo y el color del texto en

los estilos de las etiquetas para el formulario. En la siguiente figura se aprecia el error encontrado y se corrige en la hoja de estilos CSS utilizando un color de texto más oscuro.



Figura 51. Análisis de contraste (Fuente <http://wave.webaim.org/>)

Según Gómez y Alcayde, (2014) la usabilidad de un sitio web debe estar condicionada por los siguientes factores:

- **Facilidad de aprendizaje.** Una página ha de ofrecer a un usuario que no la ha visitado antes, la manera más rápida y sencilla de navegar por la web y mostrarle su información.
- **Eficiencia de uso.** Una vez realizado el aprendizaje, la página debe ofrecer al usuario la forma más rápida de trabajar o llegar a los contenidos.
- **Robustez.** Se debe intentar minimizar los errores que puedan producirse, ganando así la fidelidad del usuario.
- **Flexibilidad.** Debe ofrecer la posibilidad al usuario de la multiplicidad de vías para realizar el mismo cometido. (p. 74)

Atendiendo a estos factores, se establece la siguiente estructura del sitio web:

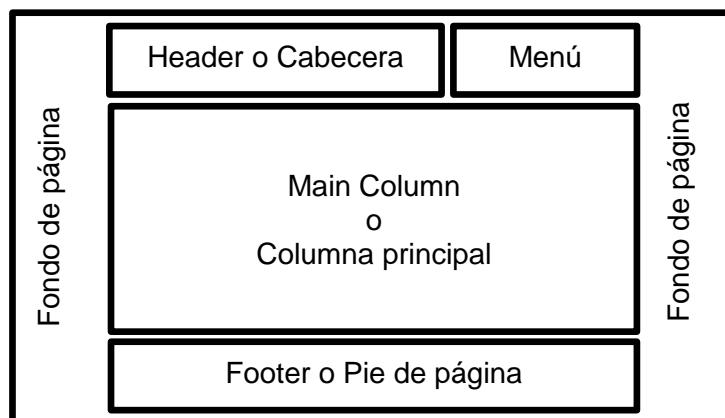


Figura 52. Estructura de la aplicación web (Elaboración propia)

Como menú, por solo tener dos opciones, se opta por uno de tipo horizontal, ubicado en la parte superior derecha de la página y se resalta la opción que el cliente ha seleccionado.

El contenido se divide en dos secciones. La página principal, o de inicio, donde se expone la información general de la aplicación y la página liquidar cirugías que contiene un formulario para que el cliente alímente la información necesaria para realizar el proceso.

El formulario tiene etiquetas que identifica cada elemento de entrada de datos. En las cajas de texto, con el uso del parámetro placeholder, se ha indicado el tipo de dato que se debe digitar y las demás opciones se establecieron por medio del elemento Select, para restringir la cantidad de entradas disponibles. Adicionalmente, el formulario es validado antes de enviarlo, muestra mensajes de error y con la tecla TAB permite desplazarse de un control a otro.

Para evaluar la aplicabilidad, se agregó un sistema de comentarios a la página web y se solicitó la valoración del funcionamiento y sugerencias de mejoras del software a personal experto.

The screenshot shows two stacked web pages. The top page is titled 'Servicios a Liquidar' and contains search fields for 'Paciente', 'Tarifario', 'Procedimiento', 'Vía Acceso', and 'Especialista', along with a 'Buscar...' button. The bottom page has a dark blue header with the text 'Ayúdanos a mejorar. Déjanos tus comentarios:' and a form for leaving comments. The comment form includes fields for 'Titular', 'Nombre' (with a note '(No sera publicado)'), 'Correo', and 'Comentario'. A 'Comenta' button is at the bottom of the comment area. The copyright notice 'Copyright © 2017 http://tarificador.hol.es/' is visible at the bottom of the top page.

Figura 53. Sistema de comentarios de la aplicación web (Elaboración propia)

Se destaca la sugerencia de mejora realizada por uno de los expertos, donde según el artículo 48 del decreto 2423 de 1996, las unidades en SMLDV para los servicios profesionales del anestesiólogo del grupo 7 en el manual tarifario SOAT es 5,3 y no 7,13.

No.	Codigo	Descripcion	Grupo Quirurg.	Especialista	Anestesiologo	Ayudante	Suma	Agregar. <input type="checkbox"/>
1	7730	Apéndicectomía	7	9	7,13	2,37	18,50	<input type="checkbox"/>
	39209	Derechos de sala Grupo 07	7	0	0	0	16,88	
	39303	Materiales de Sutura GrupoS 07 - 08 - 09	7	0	0	0	9,92	

No.	Codigo	Descripcion	Grupo Quirurg.	Especialista	Anestesiologo	Ayudante	Suma	Agregar. <input type="checkbox"/>
1	7730	Apéndicectomía	7	9	5,3	2,37	18,50	<input type="checkbox"/>
	39209	Derechos de sala Grupo 07	7	0	0	0	16,88	
	39303	Materiales de Sutura GrupoS 07 - 08 - 09	7	0	0	0	9,92	

Figura 54. Corrección unidades de liquidación Anestesiólogo tarifario SOAT (Elaboración propia)

Para solucionar el error detectado en la base de datos, se ejecuta la sentencia UPDATE, en la relación liquidador_tarifario desde PHPMyAdmin:

```
UPDATE liquidador_tarifario SET anestesiologo = 5,3 WHERE grupo_quirurgico = 7 AND anestesiologo = 7,13
```

Otro comentario para destacar es referente a la usabilidad, donde se recomienda comunicar al usuario la descripción del control y complementarla con un catálogo de descripción de los procedimientos.

Para finalizar este capítulo, en general se recibieron comentarios positivos, que indican que la aplicación cumple con su funcionalidad de liquidar procedimientos médicos, quirúrgicos y hospitalarios a partir de los tres manuales tarifarios en los que está basado.

5. Conclusiones y trabajo futuro

5.1. Conclusiones

El problema principal que se plantea en el presente TFM, es la ausencia de un aplicativo web de libre acceso para realizar liquidaciones de procedimientos médicos, quirúrgicos y hospitalarios, para fines de consulta y/o capacitaciones en el uso de estas herramientas.

Gracias al desarrollo y publicación en <http://tarificador.hol.es/> de la solución informática **Liquidador de Cirugías**, propuesta en ese proyecto, es posible encontrar una herramienta de libre acceso para liquidar procedimientos médicos, quirúrgicos y hospitalarios en Colombia.

Las principales contribuciones tras la realización de este proyecto son:

- Una base de datos unificada, con los procedimientos médicos quirúrgicos y hospitalarios que fueron extraídos directamente de los tres tarifarios ISS 2001, ISS 2004 y SOAT.
- Una sección para liquidar cirugías, de fácil manejo y aprendizaje, con un formulario alimentado por datos básicos para liquidar procedimientos con única o diferente vía de acceso, mismo o diferente especialista y tipo de cirugías bilaterales o multiples.
- Campo de autocompletado de pacientes por nombre, apellido o número de documento de identidad.
- Campo de autocompletado de procedimientos por nombre, descripción o código.
- Un aplicativo web de libre acceso para realizar liquidaciones.

Teniendo en cuenta las contribuciones mencionadas, se va a analizar el alcance de los objetivos propuestos:

Objetivo 1: Realizar el estudio de los tres tarifarios, su estructura, sus artículos y condiciones de liquidación.

Se cumple este objetivo el capítulo 4, al realizar un análisis de cada manual tarifario, que es una tarea indispensable para la identificación de los requisitos. Adicionalmente, aunque comparten características similares, cada manual tiene diferencias en su forma de liquidar los procedimientos, que sin identificación resultaría imposible tener un producto de software funcional.

Objetivo 2: Identificar los requerimientos para el desarrollo de la aplicación.

A nivel de desarrollo de software, los requerimientos establecidos por el cliente, o en este caso por los tres documentos legales que componen los manuales tarifarios, son la base fundamental para el inicio de un proyecto como el contemplado en este TFM. Por tal motivo, este objetivo se cumple al plasmar los requerimientos extraídos a partir de la documentación.

Objetivo 3: Establecer el lenguaje de codificación del lado del cliente y del servidor, el servidor de base de datos y el servidor web.

Luego de un análisis de las diferentes tecnologías presentes, se cumple este objetivo al implementar la solución en un entorno web, del lado del cliente haciendo uso del lenguaje de marcado HTML5, hojas de estilos CSS3 y JavaScript como lenguaje para programar las interactividades del aplicativo. Del lado del servidor, se implementó la solución en el lenguaje de programación PHP, un servidor web Apache y como servidor de base de datos MySQL.

Objetivo 4: Implementar la solución, publicar en el servidor web y realizar pruebas de funcionamiento.

Como resultado del cumplimiento de este objetivo, se presenta un aplicativo web, funcional y que cumple con los objetivos planteados, que sirve para realizar liquidaciones de procedimientos médicos, publicado en un servicio hosting público y accesible desde el dominio <http://tarificador.hol.es/>. A este aplicativo se le realizaron pruebas de accesibilidad, de usabilidad y pruebas con usuarios para verificar su correcto funcionamiento.

Por consiguiente, se puede concluir que, atendiendo al objetivo principal de diseñar y codificar una aplicación en entorno web para liquidar procedimientos médicos, quirúrgicos y hospitalarios de los tarifarios ISS 2001, ISS 2004 y SOAT en Colombia, ha sido alcanzado porque se entrega un aplicativo publicado en internet, que tiene un módulo para liquidar los procedimientos contemplados en los tres manuales tarifarios mediante opciones de selección y autocompletado, teniendo en cuenta las reglas de liquidación y porcentajes que se exponen en cada manual.

5.2. Líneas de trabajo futuro

Este proyecto, se pudo ampliar mediante el desarrollo de un módulo que gestione pacientes y otro para el manejo de la facturación.

Para una entidad proveedora de este tipo de servicios en salud, es indispensable tener sus pacientes registrados en una base de datos, con su información básica y otro tipo como la eps a la que están afiliados, el tipo de convenio que manejan, etc. También la historia clínica del paciente, donde se registren cada uno de los procedimientos que se le han practicado, consultas, diagnósticos y medicamentos formulados.

Adicionalmente, liquidar un procedimiento resulta útil cuando se puede realizar un cobro mediante una factura. En este sentido, adicionar a ese sistema un módulo de facturación donde se asocie a cada paciente el procedimiento y su correspondiente factura es lo más adecuado para tener un sistema de información que resulte útil a una empresa del sector.

6. Bibliografía

Acuerdo 256/2001, de 19 de diciembre, por el cual se aprueba el “Manual de Tarifas” de la Entidad Promotora de Salud del Seguro Social “EPS-ISS”. Diario Oficial El Consejo Directivo del Instituto de Seguros Sociales, 44662, de 30 de diciembre de 2001.

Acuerdo 312/2004, de 24 de febrero, por el cual se aprueba el “Manual de Tarifas” de la Entidad Promotora de Salud del Seguro Social “EPS-ISS”. Diario Oficial El Consejo Directivo del Instituto de Seguros Sociales, 45478, de 2 de marzo de 2004.

Afasalud, (s.f.). Es ilegal contratar prestación de servicios de salud a tarifa ISS. Recuperado el 10 de julio de 2017 de <https://www.afasalud.com.co/ILEGALISS.pdf>

Bahit, E. (2011). POO y MVC en PHP. Recuperado de <http://eugeniahahit.blogspot.com/>

Banco de la República, (s.f.). Indicadores de inflación básica y su variación anual. Recuperado el 10 de agosto de 2017 de <http://www.banrep.gov.co/es/inflacion-basica>

Campderrick, F. B. (2003). Ingeniería del software. Barcelona, ES: Editorial UOC. Recuperado de <http://bv.unir.net:2068>

Cobo, Á., Gómez, P., & Pérez, D. (2005). PHP y MySQL: tecnologías para el desarrollo de aplicaciones web. Madrid, ES: Ediciones Díaz de Santos. Recuperado de <http://bv.unir.net:2068>

Córcoles, T. J. E., & Montero, S. F. (2014). Diseño de interfaces web. Madrid, ES: RA-MA Editorial. Recuperado de <http://bv.unir.net:2068>

Decreto 2423/1996, de 31 de diciembre, por el cual se determina la nomenclatura y clasificación de los procedimientos médicos, quirúrgicos y hospitalarios del manual tarifario y se dictan otras disposiciones. Diario Oficial El Presidente de la Republica de Colombia, 42961, de 20 de enero de 1997.

Decreto 887/2001, de 11 de mayo, Por el cual se modifica el artículo 1o. del Decreto 2423 de 1996. Diario Oficial El Presidente de la Republica de Colombia, 44425, de 17 de abril de 2001.

Decreto 2013/2012, de 28 de septiembre, por el cual se suprime el Instituto de Seguros Sociales, ISS, se ordena su liquidación, y se dictan otras disposiciones. Diario Oficial El Presidente de la Republica de Colombia, 48567, de 28 de septiembre de 2012.

Ferrer, M. J. (2014). Implantación de aplicaciones web. Madrid, ES: RA-MA Editorial. Recuperado de <http://bv.unir.net:2068>

Gómez, L. J., & Alcayde, G. A. (2014). Construcción de páginas web. Madrid, ES: RA-MA Editorial. Recuperado de <http://bv.unir.net:2068>

Granollers, I. S. T., Lorés, V. J., & Cañas, D. J. J. (2005). Diseño de sistemas interactivos centrados en el usuario. Barcelona, ES: Editorial UOC. Retrieved from <http://bv.unir.net:2068>

Jacobson, I, Booch, G. y Rumbaugh, J. (2000). El Proceso Unificado de Desarrollo de Software. Addison Wesley.

Leuro, M. M., & Oviedo, S. I. T. (2016). Facturación & auditoría de cuentas en salud (5a. ed.). Bogotá, CO: Ecoe Ediciones. Retrieved from <http://www.ebrary.com.bdigital.sena.edu.co>

Ley 100/1993, de 23 de diciembre, por la cual se crea el sistema de seguridad social integral y se dictan otras disposiciones. Diario Oficial El Congreso de la Republica de Colombia, 41148, de 23 de diciembre de 2003

Martínez, D. R, Valderas Aranda, P. J. y Pastor López, O, (2010). Aplicaciones web. México: Alfaomega

Mohedano, J., Saiz, J. M., & Salazar, R. P. (2012). Iniciación a JavaScript. Madrid, ES: Ministerio de Educación de España. Recuperado de <http://bv.unir.net:2068>

Oracle (2014). 10 razones para elegir MySQL para las aplicaciones web de la próxima generación, Recuperado el 23 de agosto de 2017) de <https://www.mysql.com/why-mysql/white-papers/10-razones-para-elegir-mysql-para-las-aplicaciones-web-de-la-proxima-generacion/>

PHP Group, (2001-2017), Introducción a PDO, Recuperado el 27 de agosto de 2017 de <http://php.net/manual/es/intro pdo.php>

Pintos, F. J. (2014). Aplicación de técnicas de usabilidad y accesibilidad en el entorno cliente: desarrollo de aplicaciones con tecnologías web (UF1843). Madrid, ESPAÑA: IC Editorial. Recuperado de <http://bv.unir.net:2068>

Pressman, R. (2010). *Ingeniería del Software* (7^a ed.). Editorial Mcgraw-Hill.

Resolución 5261/1994, de 5 de agosto, Por la cual se establece el Manual de Actividades, Intervenciones y Procedimientos del Plan Obligatorio de Salud en el Sistema General de Seguridad Social en Salud. Diario Oficial, de 5 de agosto de 1994.

Ribes, A. P. (2011). Manual de JavaScript. Madrid, ES: Editorial CEP, S.L. Recuperado de <http://bv.unir.net:2068>

Silberschatz, A., Korth, H. F., y Sudarshan, S. (2006). Fundamentos de bases de datos (5a. ed.). Madrid, ES: McGraw-Hill España. Recuperado de <http://bv.unir.net:2068>

Sommerville, I. (2005). *Ingeniería del Software* (7^a ed.). Editorial Pearson Education.

Universidad Internacional de la Rioja. (2017). Clase Magistral de Juan Antonio Morente Molinera: Administración de Servidores web - Introducción a servidores web. Material no publicado. Recuperado el 1 de agosto de 2017, de http://campusescueladeingenieria.unir.net/cursos/lecciones/lecc_ingsw03PER3_5/documents/tema1/ideasclave.html

Universidad Internacional de la Rioja. (2017). Clase Magistral de Juan Antonio Sicilia Montalvo: Metodologías, Desarrollo y Calidad de la Ingeniería de Software - Desarrollo de software orientado a objetos. Material no publicado. Recuperado el 21 de agosto de 2017, de http://campusescueladeingenieria.unir.net/cursos/lecciones/lecc_ingsw01PER3_5/documents/tema3/ideasclave.html

Universidad Internacional de la Rioja. (2017). Clase Magistral de Nadia Gámez Gómez: Plataformas de Desarrollo de Software - Entornos de desarrollo integrado. Material no publicado. Recuperado el 3 de agosto de 2017, de http://campusescueladeingenieria.unir.net/cursos/lecciones/lecc_ingsw06PER3_5/documents/tema1/ideasclave.html

Universidad Internacional de la Rioja. (2017). Clase Magistral de Natalia Padilla Zea: Computación en el Cliente web - Protocolos de Internet: HTTP. Material no publicado. Recuperado el 25 de agosto de 2017, de http://campusescueladeingenieria.unir.net/cursos/lecciones/lecc_ingsw04PER3_5/documents/tema1/ideasclave.html

Universidad Internacional de la Rioja. (2017). Clase Magistral de Natalia Padilla Zea: Computación en el Cliente web - Desarrollo de sistemas web con HTML y CSS. Material no publicado. Recuperado el 26 de agosto de 2017, de http://campusescueladeingenieria.unir.net/cursos/lecciones/lecc_ingsw04PER3_5/documents/tema2/ideasclave.html

Universidad Internacional de la Rioja. (2017). Clase Magistral de Natalia Padilla Zea: Computación en el Cliente web - Programación con JavaScript. Material no publicado. Recuperado el 26 de agosto de 2017, de http://campusescueladeingenieria.unir.net/cursos/lecciones/lecc_ingsw04PER3_5/documents/tema3/ideasclave.html?virtualpage=0

Universidad Internacional de la Rioja. (2017). Clase Magistral de Natalia Padilla Zea: Computación en el Servidor web - PHP Avanzado. Material no publicado. Recuperado el 20 de agosto de 2017, de http://campusescueladeingenieria.unir.net/cursos/lecciones/lecc_ingsw05PER3_5/documents/tema2/ideasclave.html

Vara, M. J. M., López, S. M., & Granada, D. (2014). Desarrollo web en entorno cliente. Madrid, ES: RA-MA Editorial. Recuperado de <http://bv.unir.net:2068>

Anexos

Artículo

Manual de Usuario

Aplicación web para liquidar procedimientos médicos, quirúrgicos y hospitalarios en Colombia

Charles R. Torres Moreno
Área de Ingeniería y Tecnología
Universidad Internacional de la Rioja
Girardot, Colombia
Email: charlesrct@gmail.com

Abstract— In this work of end of Master, TFM, is the design and implementation of software developed in a web environment, to make liquidations of the medical, surgical and hospital procedures in Colombia, contemplated in decree 2423 of 1996, the agreement No. 256 of 2001 and decree 312 of 2004, known as Tariff Manuals SOAT, ISS 2001 and ISS 2004 respectively.

Keywords— Tariff Manuals, SOAT, ISS 2001, ISS 2004, Health Tariff.

I. INTRODUCCIÓN

En este trabajo se expone el diseño, desarrollo y resultados de una aplicación web para realizar liquidaciones de procedimientos médicos y quirúrgicos en Colombia, tomando como base los decretos y acuerdos, expedidos por las autoridades colombianas, donde se clasifican y aprueban las tarifas que se aplican a los usuarios de las entidades promotoras de salud.

II. TRABAJO RELACIONADO

En el área de facturación de las instituciones de salud públicas, al ser instituciones con autonomía administrativa, técnica y financiera otorgadas por la ley 100, se maneja información sensible como datos personales de los pacientes, historias clínicas, gestión de cartera y son procesos auditables. Adicionalmente, en un párrafo del artículo 185 la ley 100 se indica que “toda Institución Prestadora de Servicios de salud contará con un sistema contable que permita registrar los costos de los servicios ofrecidos”. [1]

La estructura que se presenta en el decreto 2423 de 1996 y los acuerdos 256 y 312 de 2001 y 2004 respectivamente, permite desarrollar un software que agilice las consultas de los más de 14.000 procedimientos que contemplan y que los liquide de forma automatizada.

En este sentido, se obtienen los requisitos a partir del análisis de los tres tarifarios, se crea la base de datos con los procedimientos que describe cada manual y para la codificación de la solución se tienen en cuenta las reglas de liquidación, dependiendo del tipo de procedimiento o los porcentajes según la cantidad de actos quirúrgicos.

III. CONTEXTO

En Colombia, con la ley 100 de 1993, se crea el sistema de seguridad social con el fin de unificar la normatividad y coordinar las entidades prestadoras de servicios de salud. Con esta ley, surge la necesidad de estandarizar los procedimientos médicos, quirúrgicos y hospitalarios, al igual que sus tarifas, nomenclatura y clasificación de los mismos.

El primero de los tarifarios que publica es el decreto 2423 de 1996 “por el cual se determina la nomenclatura y clasificación de los procedimientos médicos, quirúrgicos y hospitalarios del Manual Tarifario y se dictan otras disposiciones” [2] y es la base

de la liquidación utilizada con pacientes víctimas en escenarios como accidentes de tránsito, desastres naturales y atentados terroristas. Posteriormente, el 19 de diciembre de 2001 se firma el acuerdo No. 256 “por el cual se aprueba el “Manual de Tarifas” de la Entidad Promotora de Salud del Seguro Social EPS-ISS” [3] con el que se liquidan todos los procedimientos de los servicios prestados por esta entidad, amparados por la ley 100 en su artículo 275 donde “al Consejo Directivo del Instituto le compete determinar las tarifas que el Instituto aplicará en la venta de servicios de salud.” [1]

Tres años después, el 24 de febrero de 2004 se publica el acuerdo 312 “por el cual se aprueba el “Manual de Tarifas” de la Entidad Promotora de Salud del Seguro Social EPS-ISS” [4] donde se actualiza la codificación a la Clasificación Única de Procedimientos en Salud (CUPS) y el contenido de los procedimientos del manual ISS 2001. El 28 de septiembre de 2012, se publica el decreto 2013 “Por el cual se suprime el Instituto de Seguros Sociales ISS, se ordena su liquidación, y se dictan otras disposiciones” [4]. Esta medida involucra los tarifarios ISS 2001 y 2004 que eran los manuales utilizados por esa entidad pública del estado colombiano.

Aunque por razones de rentabilidad económica, no es recomendable contratar teniendo como base de liquidación los tarifarios del ISS, se incluyen en el presente desarrollo por ser herramientas que actualmente se incluyen en capacitaciones sobre facturación de servicios en salud y también por ser una muy buena oportunidad de aplicar los conocimientos y técnicas adquiridas durante el desarrollo de los contenidos de Master Universitario en Ingeniería del Software y Sistemas Informáticos.

Teniendo en cuenta lo expuesto, en cuanto a la legislación que rige la tarificación de procedimientos médicos y los complejos sistemas de información que actualmente se encuentran en el mercado, se determina orientar este trabajo a la creación de un módulo de liquidación de procedimientos, basado en las reglas establecidas por las normas que los rigen y aplicando las técnicas de desarrollo de software que fueron objeto de estudio en las diferentes asignaturas cursadas.

IV. OBJETIVOS

GENERAL

Diseñar y codificar una aplicación en entorno web para liquidar procedimientos médicos, quirúrgicos y hospitalarios de los tarifarios ISS 2001, ISS 2004 y SOAT en Colombia.

ESPECÍFICOS

- Realizar el estudio de los tres tarifarios, su estructura, sus artículos y condiciones de liquidación.
- Identificar los requerimientos para el desarrollo de la aplicación.
- Establecer el lenguaje de codificación del lado del cliente y del servidor, el servidor de base de datos y el servidor web.
- Implementar la solución, publicar en el servidor web y realizar pruebas de funcionamiento.

V. METODOLOGÍA

Jacobson, Booch y Rumbaugh definen el proceso de desarrollo de software como “el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software” [5] mientras que, Sommerville lo define como “un conjunto de actividades que conducen a la creación de un producto software”. [6] Estas actividades es posible organizarlas de forma incremental e iterativa. De acuerdo con Pressman “el modelo espiral es un modelo evolutivo del proceso del software y se acopla con la naturaleza iterativa de hacer prototipos con los aspectos controlados y sistémicos del modelo de cascada”. [7] Como se observa en la siguiente figura, el modelo en espiral permite organizar el desarrollo de una aplicación. Como primer resultado se obtienen las especificaciones del producto a partir de los manuales tarifarios, luego el modelado y codificación para generar un primer prototipo y posteriormente, versiones mejoradas del software de acuerdo con las pruebas y las retroalimentaciones:

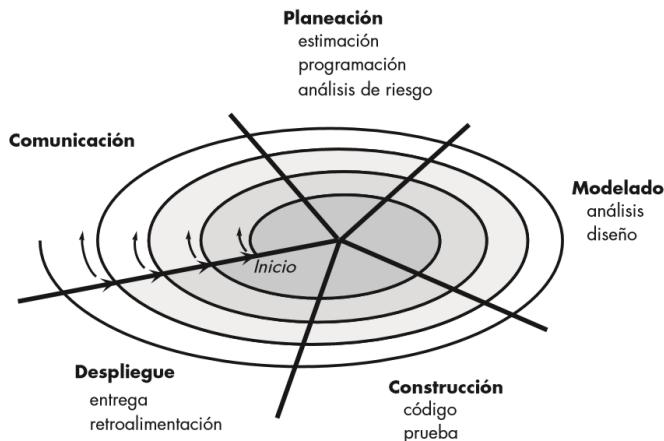


Figura 1. Modelo de espiral común para el desarrollo de software. [8]

VI. IDENTIFICACIÓN DE REQUISITOS

En el acuerdo No. 256 del 2001, para la liquidación de un procedimiento, se debe tener en cuenta las Unidades de Valor Relativo. En la tabla 1 se presenta el valor de la UVR para Especialista, Anestesiólogo, Ayudante y Odontólogo general, según el Artículo 59.

Tabla 1 Valor UVR Manual Tarifario ISS 2001 Artículo 59

Ref.	Código	Descripción	Valor
39101	PB S41101	ESPECIALISTAS DE CLÍNICAS QUIRÚRGICAS O GINECOOBSTÉTRICAS	1.270
39102	PB S41201	ESPECIALISTAS EN ANESTESIOLOGÍA	960
39103	PB S41301	MÉDICO AYUDANTE QUIRÚRGICO	360
39145	PB S41401	MÉDICO U ODONTÓLOGO GENERAL	810

Para realizar el proceso de liquidación, se debe tener en cuenta las UVR de las tablas anteriores, es decir, para un procedimiento quirúrgico son necesarios servicios profesionales, compuestos de un Cirujano, un anestesiólogo y un ayudante, más los derechos de sala y materiales.

En el acuerdo No. 312 del 2004 se establece una forma de liquidación a partir de un solo valor para las Unidades de Valor Relativo en Salud, UVR-S = \$100, como ejemplo, para obtener la tarifa en pesos de una cranealización de seno frontal, las 9.626 UVR-S por el factor de multiplicación \$100, la tarifa será de \$962.600.

Este valor del procedimiento clínico incluye: los servicios profesionales, el personal técnico y auxiliar, uso de los equipos, derechos de sala, consumo de cualquier material o elemento (reactivos, medios de contraste, película o papel fotográfico,

material de sutura).

En el Decreto 2423 de 1996, para la liquidación de un procedimiento, se utiliza como base el Salario Mínimo Legal Diario Vigente (SMLV), lo que garantiza que los precios de los procedimientos se actualicen automáticamente cada año. Por ejemplo, para el año 2017, en Colombia, el Salario Mínimo Legal Diario Vigente (SMLDV) es de \$24.590,56. Los conceptos para tener en cuenta en el proceso de liquidación son el cirujano, el anestesiólogo, el ayudante quirúrgico, los derechos de sala y los materiales. Los valores en SMLDV según el grupo quirúrgico se muestran en las siguientes tablas del artículo 48:

Tabla 2 Servicios profesionales del cirujano o ginecobstetra

CÓDIGO	GRUPO QUIRÚRGICO	SMLDV
39000	Grupo 02	2.93
39001	Grupo 03	3.57
39002	Grupo 04	4.31
39003	Grupo 05	5.86
39004	Grupo 06	7.68
39005	Grupo 07	9.00
39006	Grupo 08	10.44
39007	Grupo 09	12.76
39008	Grupo 10	15.71
39009	Grupo 11	17.62
39010	Grupo 12	19.63
39011	Grupo 13	21.48
39012	Grupo especial 20	25.43
39013	Grupo especial 21	33.10
39014	Grupo especial 22	38.67
39015	Grupo especial 23	60.73

VII. DESCRIPCIÓN DEL SOFTWARE

Para el servidor web se implementa un entorno de desarrollo local y se contratan los servicios de un proveedor de hosting para el entorno de producción. Debe cumplir con los requerimientos de un servidor apache, soporte al lenguaje php versión 5 o superior, un servidor de base de datos MySql y servidor web. Es posible instalar y configurar cada una de estas opciones por separado, pero actualmente existen proyectos que ya reúnen todas estas características en un solo programa con una interfaz de fácil administración.

Implementando la base de datos: Según Silberschatz, Korth y Sudarshan “Un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos” [9].

Los procedimientos contemplados en los tres manuales tarifarios tienen características comunes, que pueden ser almacenados en una base de datos para su posterior recuperación. Reunir en una única tabla todos los procedimientos permite desarrollar métodos comunes de consulta en lugar de métodos individuales. De los manuales tarifarios ISS 2001, ISS 2004 Y SOAT, se extrae la estructura de la base de datos que almacena la información.

Para la implementación de la base de datos, se utiliza el sistema de gestión de bases de datos relacional MySQL, desarrollada por MySQL AB y comprada en 2010 por Oracle Corporation.

En cuanto al volumen de datos, los tres manuales tarifarios cuentan con 14.786 procedimientos que se extraen directamente de los decretos y acuerdos donde fueron publicados. Aunque en comparación con los billones de datos que está en capacidad de manejar MySQL no resulta ser una cifra significativa, ya es un volumen considerable de información para realizar búsquedas manuales por parte de un usuario. Otro aspecto que se debe evitar es la duplicidad en la información, es decir que dos tuplas de una misma relación tengan los mismos valores en todos sus atributos. Por ejemplo, un procedimiento puede estar contemplado en los tres manuales tarifarios, pero tienen valores diferentes para ser liquidados, como se ve en la tabla 3. Para evitar la duplicidad entre tuplas en cada relación se asigna una clave primaria.

Tabla 3 Comparativa Tuplas Manuales Tarifarios

TARIFARIO	CÓDIGO	DESCRIPCIÓN	UNID DE LIQ.	UNIDADES FIJAS	AÑO
SOAT	7730	Apendicetomía	SMIDV	NO	TODOS
ISS	C40403	APENDICECTOMÍA (392)	UVR	8504	2004
ISS	C40403	APENDICECTOMÍA (392)	PESOS	787560	2001

Para realizar la administración de bases de datos desde la web, e implementar estas tres relaciones en el servidor de base de datos MYSQL, se hace uso de la herramienta PHPMyAdmin. “La aplicación phpmyadmin no es más que un conjunto de páginas escritas en PHP y que son copiadas directamente en el directorio que aloja las páginas web del servidor.” [10]

En la relación liquidador_tarifario, se almacenan los 14786 procedimientos que componen los tres manuales tarifarios. Como clave primaria se establece idliquidador_tarifario y contiene todos los atributos estudiados en los requerimientos de cada tarifario. En la siguiente figura se observa modelo relacional con su respectiva implementación en PHPMyAdmin:

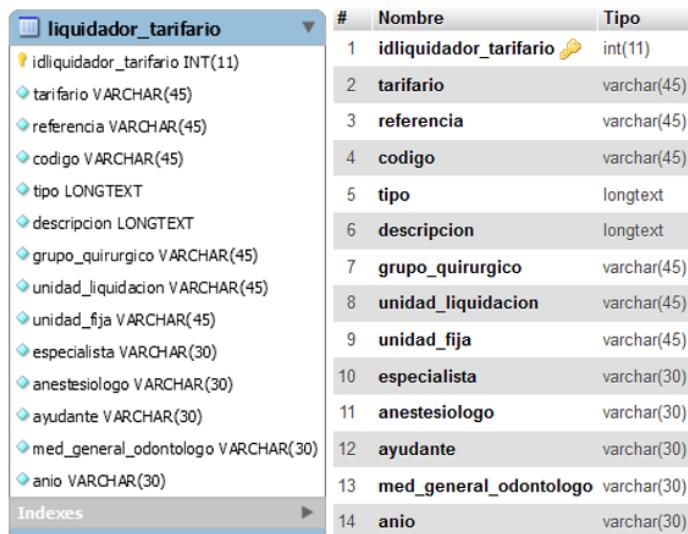


Figura 2. La relación liquidador_tarifario

Para implementar el patrón de diseño arquitectónico MVC se va a realizar la codificación de un framework básico.

Se inicia con la configuración del servidor apache para que realice determinados comportamientos mediante el archivo .htaccess (Hypertext Access) para que permita el acceso y llamado a controladores y métodos, al igual que el envío de parámetros a través de la URL.

Para continuar con la implementación del framework básico MVC se propone la siguiente estructura de archivos donde se puede apreciar el archivo .htaccess e index.php en el folder raíz de la aplicación:

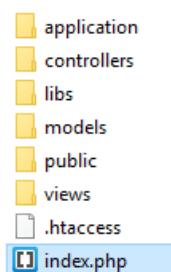


Figura 3. Estructura de archivos patrón MVC

En el folder application es donde va el código propio de la aplicación y se encuentra el archivo de configuración, los archivos con las clases de conexión a la base de datos y los archivos con las clases de donde van a heredar los modelos, vistas y controladores que se crean para realizar la codificación de la solución final, al igual que el archivo que maneja y redirige las peticiones realizadas por los usuarios.

Los folders controllers, models y views, contienen los controladores para procesar las peticiones, los modelos para acceder a los datos y las vistas con las plantillas o templates HTML para interactuar con el usuario. El folder libs se utiliza para guardar librerías desarrolladas por terceros, como por ejemplo una librería que permita exportar datos en formato de Excel o pdf. Finalmente, el folder public se utiliza para los archivos de estilos, imágenes, scripts y demás archivos necesarios para el funcionamiento del aplicativo web.

En el folder controllers se encuentran los archivos con los controladores de la aplicación, como regla general se crean con el nombre del controlador seguido de la palabra Controller, por ejemplo, para el controlador index se crea el archivo indexController.php que hereda de la clase Controller. En este caso no se invoca un modelo, solamente llama la vista y pasa los parámetros título y el nombre del archivo que se va a renderizar, como se ve en la siguiente figura:

```

11 class indexController extends Controller
12 {
13     public function __construct() {
14         parent::__construct();
15     }
16
17     public function index()
18     {
19         $this->view->titulo = 'Portada';
20         $this->view->renderizar('index', 'inicio');
21     }
22 }
  
```

Figura 4. Archivo indexController.php

En el folder Models se encuentra el archivo liquidadorModel.php que hereda de la clase Model y contiene los métodos para autocompletar la consulta en la tabla de pacientes, la tabla de procedimientos médicos según el tarifario seleccionado y la tabla de salarios mínimos para el caso del tarifario SOAT.

```

3 class liquidadorModel extends Model {
4
5     public function __construct() {
6         parent::__construct();
7     }
8
9     public function getPaciente($valor) { ... }
10
11    public function getProcedimiento($valor, $idtarifario, $anio) { ... }
12
13    public function getAnual() {
14        $result = $this->db->query("SELECT anual
15                                    FROM parametros_tarifarios
16                                    WHERE cancel = 0
17                                    ORDER BY anual DESC");
18
19        return $result->fetchall();
20    }
21
22    public function getQuirofano($cod_quirofano) {
23        $result = $this->db->query('SELECT
24            idliquidador_tarifario,
25            codigo,
26            descripcion,
27            grupo_quirurgico,
28            suma,
29            tarifario,
30            anio,
31            tipo,
32            unidad_liquidacion,
33            med_general_odontologo
34            FROM
35            liquidador_tarifario
36            WHERE
37            ');
38    }
  
```

Figura 5. Segmento archivo liquidadorModel.php

En el folder views, se crean las vistas que componen la aplicación y se encuentra un folder por cada vista que se necesite. Para la vista index se crea el archivo index.phtml que contiene el código que se muestra en la imagen siguiente y es un segmento de etiquetas HTML para ser embebidas en la plantilla del framework.

```

1 <dl>
2   
3   <h1>Aplicación web para liquidar procedimientos médicos, quirúrgicos y hospitalarios</h1>
4   <h2>Máster universitario en Ingeniería de Software y Sistemas Informáticos</h2>
5   <ul>
6     <li>Tarifarios:</li>
7     <dd>
8       <ul>
9         <li type="circle">SOAT</li>
10        <li type="circle">ISS 256 - 2001</li>
11        <li type="circle">ISS 312 - 2004</li>
12     </ul>
13   </dd>
  
```

Figura 6. Archivo index.phtml

Luego de ser invocada por el archivo indexController.php, la vista index.phtml es renderizada y embebida en el resto del template de la aplicación y se presenta al usuario como se ve en la siguiente figura:

Tarifarios:

- SOAT
- ISS 256 - 2001
- ISS 312 - 2004

Figura 7. Renderizado vista index.phtml

La aplicación se va a dividir en dos secciones, la primera es el inicio donde se proporciona la información general y la segunda es el liquidador de cirugías. Para presentar los elementos en el navegador, se crea una hoja de estilos en cascada CSS contenida en el archivo estilos.css, donde se definen colores de la plantilla, tipos y tamaños de las fuentes, forma y colores de los botones, cuadros de texto, tablas y en general todos los detalles que permiten crear una interfaz agradable a la vista y amigable con el usuario. En la siguiente figura se aprecia un segmento de estos estilos aplicados a elementos como el menú, los logos, entre otros:

```

53 #main{
54     position: relative;
55     margin: 0 auto 0 auto;
56     width: 900px;
57     padding-top: 20px;
58 }
59
60 #header{
61     width: 900px;
62     height: 100px;
63 }
64
65 #logo{
66     width: 270px;
67     float: left;
68     margin: 0;
69     padding: 10px 10px 10px 20px;
70     height: 80px;
71 }
73 #logo h1{
74     color: #FFFFFF;
75     line-height: 40px;
76     text-shadow: 0 0 0;
77     margin: 0;
78 }
79
80 #logo-unir{
81     float: left;
82     margin-right: 0.5cm;
83 }
84 #menu{
85     width: 580px;
86     height: 80px;
87     float: left;
88     margin: 0;
89     padding: 10px;
90 }
92 #content{background: #999999;
93     color: #E6E6E6;

```

Figura 8. Segmento archivo estilos.css

En concordancia con los requerimientos, el sistema debe ser alimentado por el usuario según el tipo de tarifario a utilizar, la vía de acceso, el especialista y la cantidad de procedimientos, también debe realizar una búsqueda del procedimiento a liquidar. Incluir JavaScript en esta aplicación web, implica poder realizar estas tareas validando la información proporcionada por el cliente.

En primera instancia se solicitan los datos del paciente, su búsqueda se puede realizar por nombre, apellido o número de documento, luego se selecciona el tarifario ISS 2001, ISS 2004 o SOAT. Para el caso del tarifario SOAT se debe escoger el año de liquidación. A continuación, se realiza la búsqueda del procedimiento por código o nombre y finalmente se seleccionan las opciones con que se liquida ese procedimiento.

Figura 9. Interfaz WEB Liquidador de Cirugías

Para los campos Paciente y Procedimiento de la interfaz Web del Liquidador de Cirugías, que se muestra en la figura anterior, se utiliza la función de autocompletado, de jQuery:

Figura 10. Búsqueda de procedimiento con autocompletado

En la codificación de la solución, el uso de Ajax se aplica en

procesos como en la de realizar la búsqueda de un procedimiento. La petición por parte del cliente se inicia al pulsar el botón Buscar que invoca la función BuscarServicios() de JavaScript.

```

65 function BuscarServicios(){
66     jQuery.ajax({
67         url: 'liquidador/tabla_procedimiento/',
68         type: 'GET',
69         data: {
70             Dato:idprocedimiento[1],
71             Tarifario:tarifario,
72             anio:anio
73         },
74         beforeSend: function(){
75             //Muestra mensaje mientras se ejecuta el proceso.
76             jQuery("#resultado_datos").html("Cargando...");
77         },
78         success: function(data){
79             jQuery('#resultado_datos').html("");
80             jQuery('#Div_Servicios').html(data);
81         }
82     }/*data*/
83 );

```

Figura 11. Uso de Ajax en la búsqueda de procedimientos

La función Ajax de jQuery, invoca el controlador liquidador y el método tabla_procedimiento. Como parámetros se envían el id del procedimiento, el tarifario y el año. Este método, se encarga de invocar el método getServicio del modelo liquidadorModel y enviar la información a la vista servicio.phtml

Para el caso de adicionar un procedimiento. Se produce un proceso similar. En primer instancia se llama la función Add_Service() de JavaScript y vía Ajax se invoca el método adicionar_procedimiento de liquidadorController, que finalmente llama el método addservicio para pasar los datos a la vista. Finalmente, se aplica la función para liquidar los procedimientos agregados. En la función Liquidar() de JavaScript se encuentran todas las condiciones necesarias para realizar las liquidaciones según las condiciones especificadas en los tres manuales tarifarios.

VIII. EVALUACIÓN

Para la evaluación de la aplicación se tienen en cuenta dos aspectos. El primero desde el punto de vista de la usabilidad y el segundo analizando la aplicabilidad de la herramienta para resolver el problema propuesto. Para la evaluación inicial de accesibilidad se maneja la herramienta en línea WAVE (web accessibility evaluation tool) disponible en la URL <http://wave.webaim.org/>, donde el principal error encontrado fue el bajo contraste que se manejaba entre el fondo y el color del texto en los estilos de las etiquetas para el formulario. En la siguiente figura se aprecia el error encontrado y se corrige en la hoja de estilos CSS utilizando un color de texto más oscuro.



Figura 12. Análisis de contraste

Según Gómez y Alcayde, la usabilidad de un sitio web debe estar condicionada por los siguientes factores:

- Facilidad de aprendizaje. Una página ha de ofrecer a un usuario que no la ha visitado antes, la manera más rápida y sencilla de navegar por la web y mostrarle su información.
- Eficiencia de uso. Una vez realizado el aprendizaje, la página debe ofrecer al usuario la forma más rápida de trabajar o llegar a los contenidos.
- Robustez. Se debe intentar minimizar los errores que puedan producirse, ganando así la fidelidad del usuario.
- Flexibilidad. Debe ofrecer la posibilidad al usuario de la multiplicidad de vías para realizar el mismo cometido. [11]

Atendiendo a estos factores, se establece la siguiente estructura del

sitio web:

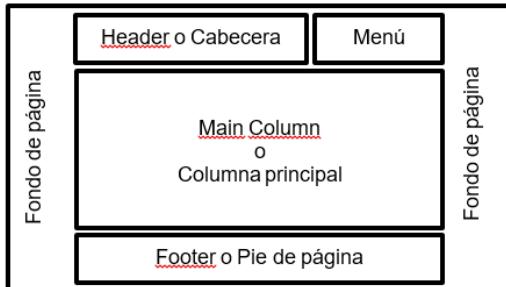


Figura 13. Estructura de la aplicación web

Como menú, por solo tener dos opciones, se opta por uno de tipo horizontal, ubicado en la parte superior derecha de la página y se resalta la opción que el cliente ha seleccionado.

El contenido se divide en dos secciones. La página principal, o de inicio, donde se expone la información general de la aplicación y la página liquidar cirugías que contiene un formulario para que el cliente alimente la información necesaria para realizar el proceso.

El formulario tiene etiquetas que identifica cada elemento de entrada de datos. En las cajas de texto, con el uso del parámetro placeholder, se ha indicado el tipo de dato que se debe digitar y las demás opciones se establecieron por medio del elemento Select, para restringir la cantidad de entradas disponibles. Adicionalmente, el formulario es validado antes de enviarlo, muestra mensajes de error y con la tecla TAB permite desplazarse de un control a otro.

IX. CONCLUSIONES

El problema principal que se plantea en el presente TFM, es la ausencia de un aplicativo web de libre acceso para realizar liquidaciones de procedimientos médicos, quirúrgicos y hospitalarios, para fines de consulta y/o capacitaciones en el uso de estas herramientas.

Gracias al desarrollo y publicación en <http://tarificador.hol.es/> de la solución informática Liquidador de Cirugías, propuesta en ese proyecto, es posible encontrar una herramienta de libre acceso para liquidar procedimientos médicos, quirúrgicos y hospitalarios en Colombia.

Las principales contribuciones tras la realización de este proyecto son:

- Una base de datos unificada, con los procedimientos médicos quirúrgicos y hospitalarios que fueron extraídos de los tres tarifarios ISS 2001, ISS 2004 y SOAT.
- Una sección para liquidar cirugías, de fácil manejo y aprendizaje, con un formulario alimentado por datos básicos para liquidar procedimientos con única o diferente vía de acceso, mismo o diferente especialista y tipo de cirugías bilaterales o múltiples.
- Campo de autocompletado de pacientes por nombre, apellido o número de documento de identidad.
- Campo de autocompletado de procedimientos por nombre, descripción o código.

Teniendo en cuenta las contribuciones mencionadas, se va a analizar el alcance de los objetivos propuestos:

Objetivo 1: Realizar el estudio de los tres tarifarios, su estructura, sus artículos y condiciones de liquidación.

Se cumple este objetivo al realizar un análisis de cada manual tarifario, que es una tarea indispensable para la identificación de los requisitos. Adicionalmente, aunque comparten características similares, cada manual tiene diferencias en su forma de liquidar los procedimientos, que sin identificación resultaría imposible tener un producto de software funcional.

Objetivo 2: Identificar los requerimientos para el desarrollo de la

aplicación.

A nivel de desarrollo de software, los requerimientos establecidos por el cliente, o en este caso por los tres documentos legales que componen los manuales tarifarios, son la base fundamental para el inicio de un proyecto como el contemplado en este TFM. Por tal motivo, este objetivo se cumple al plasmar los requerimientos extraídos a partir de la documentación.

Objetivo 3: Establecer el lenguaje de codificación del lado del cliente y del servidor, el servidor de base de datos y el servidor web.

Luego de un análisis de las diferentes tecnologías presentes, se cumple este objetivo al implementar la solución en un entorno web, del lado del cliente haciendo uso del lenguaje de marcado HTML5, hojas de estilos CSS3 y JavaScript como lenguaje para programar las interactividades del aplicativo. Del lado del servidor, se implementó la solución en el lenguaje de programación PHP, un servidor web apache y como servidor de base de datos MySQL.

Objetivo 4: Implementar la solución, publicar en el servidor web y realizar pruebas de funcionamiento.

Como resultado del cumplimiento de este objetivo, se presenta un aplicativo web, funcional y que cumple con los objetivos planteados, que sirve para realizar liquidaciones de procedimientos médicos, publicado en un servicio hosting público y accesible desde el dominio <http://tarificador.hol.es/>. A este aplicativo se le realizaron pruebas se realizaron de accesibilidad, de usabilidad y pruebas con usuarios para verificar su correcto funcionamiento.

Por consiguiente, se puede concluir que, atendiendo al objetivo principal de diseñar y codificar una aplicación en entorno web para liquidar procedimientos médicos, quirúrgicos y hospitalarios de los tarifarios ISS 2001, ISS 2004 y SOAT en Colombia, ha sido alcanzado porque se entrega un aplicativo publicado en internet, que tiene un módulo para liquidar los procedimientos contemplados en los tres manuales tarifarios mediante opciones de selección y autocompletado, teniendo en cuenta las reglas de liquidación y porcentajes que se exponen en cada manual.

REFERENCIAS

- [1] Ley 100/1993, Por la cual se crea el sistema de seguridad social integral y se dictan otras disposiciones. Diario Oficial El Congreso de la Republica de Colombia, 41148, de 23 de diciembre de 2003.
- [2] Decreto 2423/1996 por el cual se determina la nomenclatura y clasificación de los procedimientos médicos, quirúrgicos y hospitalarios del Manual Tarifario y se dictan otras disposiciones. Diario Oficial El Presidente de la Republica de Colombia, 42961, de 20 de enero de 1997.
- [3] Acuerdo 256/2001, por el cual se aprueba el “Manual de Tarifas” de la Entidad Promotora de Salud del Seguro Social “EPS-ISS”. Diario Oficial El Consejo Directivo del Instituto de Seguros Sociales, 44662, de 30 de diciembre de 2001.
- [4] Acuerdo 312/2004, por el cual se aprueba el “Manual de Tarifas” de la Entidad Promotora de Salud del Seguro Social “EPS-ISS”. Diario Oficial El Consejo Directivo del Instituto de Seguros Sociales, 45478, de 2 de marzo de 2004.
- [5] Decreto 2423/1996, de 31 de diciembre, por el cual se determina la nomenclatura y clasificación de los procedimientos médicos, quirúrgicos y hospitalarios del Manual Tarifario y se dictan otras disposiciones. Diario Oficial El Presidente de la Republica de Colombia, 42961, de 20 de enero de 1997.
- [6] Jacobson, I., Booch, G. y Rumbaugh, J. (2000). El Proceso Unificado de Desarrollo de Software. Addison Wesley.
- [7] Sommerville, I. (2005). Ingeniería del Software (7^a ed.). Editorial Pearson Education.
- [8] Pressman, R. (2010). Ingeniería del Software (7^a ed.). Editorial McGraw-Hill.
- [9] Silberschatz, A., Korth, H. F., y Sudarshan, S. Fundamentos de bases de datos (5a. ed.). Madrid, ES: McGraw-Hill España. Recuperado de <http://bv.unir.net:2068>
- [10] Cobo, Á., Gómez, P., & Pérez, D. PHP y MySQL: tecnologías para el desarrollo de aplicaciones web. Madrid, ES: Ediciones Díaz de Santos. Recuperado de <http://bv.unir.net:2068>
- [11] Gómez, L. J., & Alcayde, G. A. Construcción de páginas web. Madrid, ES: RA-MA Editorial. Recuperado de <http://bv.unir.net:2068>

**Universidad Internacional de La Rioja
Máster universitario en Ingeniería de Software y
Sistemas Informáticos**

**Aplicación web para liquidar
procedimientos médicos,
quirúrgicos y hospitalarios en
Colombia.**

Manual de Usuario

Trabajo Fin de Máster

Presentado por: Torres Moreno, Charles Richar

Director/a: Machío Regidor, Francisco

Ciudad: Girardot
Fecha: 08/09/2017

Resumen

En este manual de usuario, se encuentran las instrucciones para realizar liquidaciones de procedimientos médicos en Colombia, contemplados en el decreto 2423 de 1996, el acuerdo No. 256 del 2001 y el decreto 312 del 2004, conocidos como Manuales Tarifarios SOAT, ISS 2001 e ISS 2004 respectivamente, haciendo uso del aplicativo web “Liquidador de cirugías” que se encuentra publicado en la URL <http://tarificador.hol.es/>

Índice de contenidos

1. Interfaz gráfica de la aplicación	4
2. Liquidación de procedimientos	4
2.1 Liquidar procedimiento única vía de acceso.	5
2.2 Liquidar procedimientos múltiples.....	7

Índice de figuras

Figura 1. Portada aplicación web (Elaboración propia).....	4
Figura 2. Interfaz para Liquidar Cirugías (Elaboración propia).....	5
Figura 3. Autocompletado del campo paciente (Elaboración propia)	5
Figura 4. Selección del tarifario (Elaboración propia)	6
Figura 5. Autocompletado del campo Procedimiento (Elaboración propia)	6
Figura 6. Resultado de la Búsqueda de un Procedimiento (Elaboración propia)	6
Figura 7. Costo del procedimiento (Elaboración propia)	7
Figura 8. Opciones para liquidar procedimientos múltiples (Elaboración propia)	7
Figura 9. Ejemplo de procedimientos múltiples (Elaboración propia).....	8
Figura 10. Liquidación de procedimientos múltiples (Elaboración propia).....	8

1. Interfaz gráfica de la aplicación

Para ingresar a la aplicación web liquidador de cirugías, debe contar con acceso a internet y un navegador WEB. En la barra de direcciones digite la URL <http://tarificador.hol.es/>

La aplicación consta de dos secciones accesibles a través del menú, la portada o inicio y liquidar cirugías, como se ve en la figura 1:

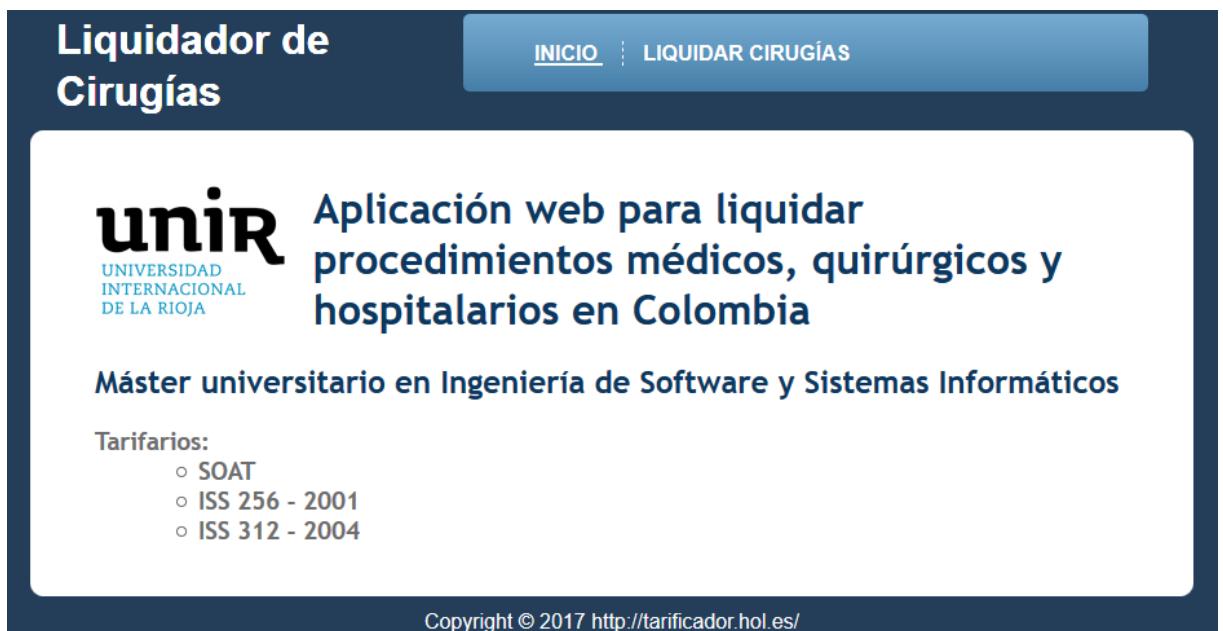


Figura 1. Portada aplicación web (Elaboración propia)

En la página principal o portada se encuentra la información general del aplicativo, como la descripción y los manuales tarifarios que maneja.

La sección LIQUIDAR CIRUGÍAS es la encargada de solicitar los datos para realizar la búsqueda y condiciones para realizar la liquidación. En los siguientes apartados se describen los pasos a seguir.

2. Liquidación de procedimientos

Para iniciar el proceso primero se debe acceder a la página web del aplicativo a través de la dirección <http://tarificador.hol.es/>

Luego, en el menú, dar clic en la opción LIQUIDAR CIRUGÍAS. En la figura 2 se observa el formulario de la aplicación.

Figura 2. Interfaz para Liquidar Cirugías (Elaboración propia)

Los requisitos que se deben tener para realizar una liquidación son el nombre del paciente, el tarifario a utilizar, el procedimiento, la vía de acceso y si es realizada por el mismo o diferente especialistas.

2.1 Liquidar procedimiento única vía de acceso.

En el campo paciente, figura 3, digitar el nombre, apellido o número de documento, esperar que el sistema muestre las opciones de autocompletado y seleccionar el paciente al que se le practicó el procedimiento.

Figura 3. Autocompletado del campo paciente (Elaboración propia)

A continuación, seleccionar el tarifario entre las tres opciones disponibles. SOAT, ISS 256 AÑO 2001 o ISS 312 AÑO 2004. Para el caso del tarifario SOAT, se debe seleccionar también el año. Figura 4.

The screenshot shows a user interface for selecting a tariff. At the top right is a text input field containing "ANDREA::ISABEL::CAICEDO::RODRIGUEZ::1004236106:::10". Below it is a dropdown menu labeled "Tarifario" with "Seleccion.." and "SOAT" selected. To the right is a search bar with placeholder text "Digite el código o nombre del procedimiento.". On the left, there are buttons for "Proced" (disabled), "Via Acceso" (disabled), and "Especialista" (disabled). Below the "Via Acceso" button is a dropdown menu with "ISS 256 AÑO 2001" and "ISS 312 AÑO 2004".

Figura 4. Selección del tarifario (Elaboración propia)

En el campo Procedimiento se debe digitar el código o el nombre. Figura 5.

The screenshot shows the same interface as Figure 4, but the search bar now contains "histerectomia ab". Below the search bar, a list of suggestions is displayed: "11470::Histerectomía abdominal (total o subtotal)::SOAT::840", "11471::Histerectomía abdominal radical::SOAT::841", "11472::Histerectomía abdominal ampliada::SOAT::842", and "511001:: Histerectomía abdominal total con o sin remoción de trompas u ovarios ::SOAT::3245". The first suggestion is highlighted with a blue background.

Figura 5. Autocompletado del campo Procedimiento (Elaboración propia)

Seleccionar la vía de acceso, el especialista y dar clic en el botón buscar.

The screenshot shows the search results for the procedure "Histerectomía abdominal (total o subtotal)". The search bar above the table contains "11470::Histerectomía abdominal (total o subtotal)::SOAT:::840". The table below has columns: No., Código, Descripción, Grupo Quirurg., Especialista, Anestesiólogo, Ayudante, Suma, and Agregar. The first row of the table is selected, indicated by a blue border and a checked checkbox in the "Agregar" column.

Procedimiento:		SERVICIOS: TARIFARIO SOAT						
No.	Código	Descripción	Grupo Quirurg.	Especialista	Anestesiólogo	Ayudante	Suma	Agregar
1	11470	Histerectomía abdominal (total o subtotal)	11	17,62	10,08	4,83	32,53	<input checked="" type="checkbox"/>
	39213	Derechos de sala Grupo 11	11	0	0	0	29,95	
	39304	Materiales de Sutura GrupoS 10 - 11 - 12 - 13	11	0	0	0	15,72	

Figura 6. Resultado de la Búsqueda de un Procedimiento (Elaboración propia)

Verificar el procedimiento, dar clic en la casilla de verificación y luego dar clic en el botón Adicionar.

SERVICIOS A LIQUIDAR. TARIFARIO SOAT								
No.	Codigo	Descripcion	Grupo Quirurg.	Especialista	Anestesiologo	Ayudante	Suma	<input checked="" type="checkbox"/>
1	11470	Histerectomía abdominal (total o subtotal)	11	\$418100	\$245900	\$98400	\$762400	<input type="checkbox"/>
	39213	Derechos de sala Grupo 11	11	0	0	0	\$713200	
	39304	Materiales de Sutura GrupoS 10 - 11 - 12 - 13	11	0	0	0	\$368900	
							SubTotal:	\$1,844,500
							Total:	\$1,844,500

.: LIQUIDAR ..

Figura 7. Costo del procedimiento (Elaboración propia)

Para finalizar, dar clic en el botón LIQUIDAR y verificar el mensaje Procedimiento No. 1 liquidado al 100%.

2.2 Liquidar procedimientos múltiples.

Inicialmente se debe seleccionar el paciente, el tarifario y buscar el procedimiento.

Servicios a Liquidar

Paciente: ANDREA::ISABEL::CAICEDO::RODRIGUEZ::1004236106:::10

Tarifario: SOAT Año: 2017

Procedimiento: 1101::Craneotomía para extracción cuerpo extraño, incluye esquirlectomía::SO

Vía Acceso: Diferente Vía Especialista: Diferente Especialista

Tipo Cirugia: Multiple Cant.Cirugias Multiples: 3

Buscar...

Figura 8. Opciones para liquidar procedimientos múltiples (Elaboración propia)

En vía de acceso seleccionar la opción Misma vía o Diferente vía. En tipo de cirugía escoger Bilateral o Múltiple. Si el tipo de cirugía es Múltiple se deben escoger la cantidad de procedimientos realizados.

Luego se debe buscar y adicionar el procedimiento. Para los procedimientos adicionales, dar clic en el campo para buscar y agregar los procedimientos adicionales.

SERVICIOS A LIQUIDAR. TARIFARIO SOAT

No.	Codigo	Descripcion	Grupo Quirurg.	Especialista	Anestesiologo	Ayudante	Suma	<input checked="" type="checkbox"/> <input type="checkbox"/>
1	15224	Reparación de coloboma, incluye naso oculares oro oculares	10	\$368900	\$221400	\$98400	\$688700	<input type="checkbox"/>
	39212	Derechos de sala Grupo 10	10	0	0	0	\$688600	
	39304	Materiales de Sutura GrupoS 10 - 11 - 12 - 13	10	0	0	0	\$368900	
SubTotal:								\$1,746,200
2	1101	Craneotomía para extracción cuerpo extraño, incluye esquirlectomía	9	\$295100	\$245900	\$73800	\$614800	<input type="checkbox"/>
	39211	Derechos de sala Grupo 09	9	0	0	0	\$516400	
	39303	Materiales de Sutura GrupoS 07 - 08 - 09	9	0	0	0	\$221400	
SubTotal:								\$1,352,600
3	18100	Laringoscopia o antroscopia	4	\$98400	\$73800	\$0	\$172200	<input type="checkbox"/>
	39206	Derechos de sala Grupo 04	4	0	0	0	\$172200	
	39302	Materiales de Sutura GrupoS 04 - 05 - 06	4	0	0	0	\$98400	
SubTotal:								\$442,800
Total:								\$3,541,600

.: LIQUIDAR .:

Figura 9. Ejemplo de procedimientos múltiples (Elaboración propia)

Para finalizar se debe dar clic en el botón liquidar esperar que se apliquen los porcentajes, empezando por el procedimiento de mayor al menor valor.

SERVICIOS A LIQUIDAR. TARIFARIO SOAT

No.	Codigo	Descripcion	Grupo Quirurg.	Especialista	Anestesiologo	Ayudante	Suma	<input checked="" type="checkbox"/> <input type="checkbox"/>
1	15224	Reparación de coloboma, incluye naso oculares oro oculares	10	\$368900	\$221400	\$98400	\$688700	<input type="checkbox"/>
	39212	Derechos de sala Grupo 10	10	0	0	0	\$688600	
	39304	Materiales de Sutura GrupoS 10 - 11 - 12 - 13	10	0	0	0	\$368900	
SubTotal:								\$1,746,200
2	1101	Craneotomía para extracción cuerpo extraño, incluye esquirlectomía	9	\$147600	\$184500	\$0	\$332100	<input type="checkbox"/>
	39211	Derechos de sala Grupo 09	9	0	0	0	\$258200	
	39303	Materiales de Sutura GrupoS 07 - 08 - 09	9	0	0	0	\$166000	
SubTotal:								\$756,300
3	18100	Laringoscopia o antroscopia	4	\$49200	\$55400	\$0	\$104600	<input type="checkbox"/>
	39206	Derechos de sala Grupo 04	4	0	0	0	\$86100	
	39302	Materiales de Sutura GrupoS 04 - 05 - 06	4	0	0	0	\$73800	
SubTotal:								\$264,500
Total:								\$2,767,000

Figura 10. Liquidación de procedimientos múltiples (Elaboración propia)