

SISTEMA DE INFORMACION PARA EL CONTROL DE LAS REMISIONES
DE LA EMPRESA GRUPO AMAREY



ANGEL ANDRES PULIDO SUAREZ

FUNDACION UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERIA DE SISTEMAS
INGENIERIA DE SISTEMAS
BOGOTA D.C.
2015

SISTEMA DE INFORMACION PARA EL CONTROL DE LAS REMISIONES
DE LA EMPRESA GRUPO AMAREY

ANGEL ANDRES PULIDO SUAREZ

PROYECTO PRESENTADO PARA OPTAR AL TÍTULO DE INGENIERO DE
SISTEMAS

ASESOR
MIGUEL ARMANDO HERNÁNDEZ BEJARANO
INGENIERO DE SISTEMAS

FUNDACION UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERIA DE SISTEMAS
INGENIERIA DE SISTEMAS
BOGOTA D.C.
2015

Notas de aceptación



Firma presidente del jurado

Firma del jurado

Firma del jurado

Bogotá 26 de Noviembre de 2015



Las directivas de la fundación
Universitaria los libertadores, los
Jurados calificadores y el cuerpo
Docente no son responsables por los
Criterios e ideas expuestas en el
Presente documento. Estos
Corresponden únicamente a los autores.

TABLA DE CONTENIDO

INTRODUCCION	9
1. PLANTEAMIENTO DEL PROBLEMA.....	10
2. JUSTIFICACIÓN.....	11
3. OBJETIVO GENERAL.....	12
4. OBJETIVOS ESPECÍFICOS.....	12
5. DELIMITACIÓN Y ALCANCE.....	13
6. MARCO REFERENCIAL.....	14
6.1.1 Las remisiones	16
6.1.2 La Nota de Remisión.....	16
6.2 MARCO CONCEPTUAL.....	18
6.3 MARCO TEÓRICO.....	23
6.3.1 Las Metodologías Ágiles.....	23
6.3.2 Las Aplicaciones Web.....	23
7. MARCO METODOLOGICO.....	27
7.1.1 Selección Motor de Bases de datos.....	30
Imagen 6. Motor base de datos.....	31
7.1.2 Selección herramienta de desarrollo para aplicación web.....	31
Imagen 7. mvc.....	32
7.1.3 Selección de la Metodología Ágil.....	33
7.2 Proceso de Desarrollo	36
7.2.1 Los Roles	36
7.2.2 Planificación	36
7.2.2.1 Historias de usuarios.....	36
• Requerimientos funcionales	39
7.2.2.2 Pruebas.....	46
7.2.2.3 Versiones e Iteraciones.....	48
8. CRONOGRAMA	63
9. PRESUPUESTO	64

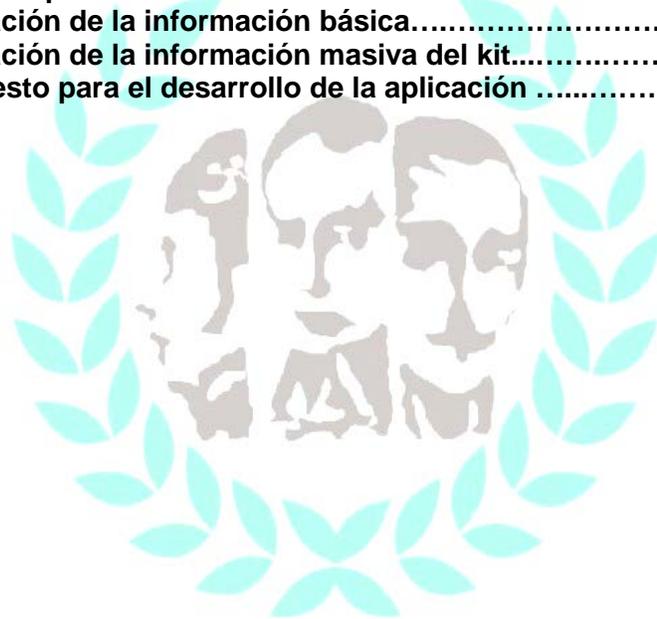
10. DISEÑO	65
10.1 Codificación	65
10.1.1 Cliente siempre disponible	65
10.1.2 Estándares de codificación	65
10.1.3 Pruebas	65
10.1.4 Instalación	66
10.1.5 Mantenimiento	66
11. INTERFACES	67
11.1 Interfaces parte pública	67
11.1.1 Inicio	67
11.1.2 Agregar Nuevo Kit	68
11.1.3 Modificar Kits	69
11.1.4 Cargar Archivo	71
11.1.6 Historial Remisión	75
11.1.7 Cambio de Contraseña	76
11.1.8 Nuevo Usuario	78
12. CONCLUSIONES	81
13. REFERENCIAS BIBLIOGRAFICAS	82
GLOSARIO	84
ANEXOS	90
ANEXO A ESTÁNDARES	90
ANEXO B PRUEBAS	92
ANEXO C CASOS DE USO	96
ANEXO D DIAGRAMA DE CLASES	104
• Tarjetas CRC	105
ANEXO E INSTALACIÓN	109
ANEXO F CONEXIÓN SQL SERVER – WAMP SERVER	111
ANEXO G MERGE	113
ANEXO H DICCIONARIO DE DATOS	116
ANEXO I CARTA DE CESION DE DERECHOS	124

TABLA DE IMAGENES

Imagen. Formulario twg.....	14
Imagen 1. Logística inversa.....	18
Imagen 2. Integración logística.....	19
Imagen 3. Empresa Amarey Nova Medical s.a.....	20
Imagen 4. Caracterización.....	21
Imagen 5. Caracterización.....	22
Imagen 6. Motor base de datos.....	31
Imagen 7. Mvc.....	32
Imagen Caso uso autenticar usuarios ante la aplicación.....	49
Imagen Caso uso creación usuario.....	50
Imagen Caos uso administrar kits y elementos.....	50
Imagen Caso uso administrar kit.....	51
Imagen Caso uso creación de las remisiones y datos Adicionales de la remisión.....	51
Imagen Caso uso consultar reporte de historial de remisiones.....	52
Imagen Diagrama de clases	53
Imagen Diagrama de componentes.....	54
Imagen Modelo entidad relación.....	55
Imagen 8. Página maestra.....	60
Imagen Cronograma.....	63
Imagen 9. Iniciar sesión.....	67
Imagen 10. Inicio.....	67
Imagen 11. Nuevo kit.....	68
Imagen 12. Modificar kit.....	69
Imagen 12.1, Editar.....	70
Imagen 12.2. Eliminar.....	70
Imagen 13. Cargar archivo.....	71
Imagen 14. Generar remisión.....	72
Imagen 14.1. Datos adicionales.....	73
Imagen 14.2. Validación datos adicionales.....	73
Imagen 14.3. Generar pdf.....	74
Imagen 14.3. Pdf.....	74
Imagen 15. Historial remisiones.....	75
Imagen 15.1. Consulta remisión.....	76
Imagen 16. Cambio contraseña.....	76
Imagen 16.1. Validación contraseña.....	77
Imagen 16.2. Confirmación cambio contraseña.....	77
Imagen 17. Nuevo usuario.....	78
Imagen 17.1. Usuario creado.....	79
Imagen 17.2. Lista usuarios.....	79
Imagen 17.3. Formulario edición usuarios.....	79
Imagen 17.4. Confirmación edición usuarios.....	80

INDECE DE TABLAS

Tabla 1. Comparación Metodologías Ágiles y Tradicionales.....	34
Tabla 2. Requerimientos no funcionales.....	37
Tabla 3. Plantilla de Historia de usuarios.....	39
Tabla 4. Requerimiento autenticación de usuarios.....	40
Tabla 5. Requerimiento asignación de roles.....	41
Tabla 6. Requerimiento Administración de Kits y Materiales.....	43
Tabla 7. Administración de Kits.....	43
Tabla 8. Creación de las remisiones.....	44
Tabla 9. Datos Adicionales de la remisión.....	44
Tabla 10. Consultar reporte de historial de remisiones.....	45
Tabla 11. Actualización de la información básica.....	45
Tabla 12. Actualización de la información masiva del kit.....	46
Tabla 13. Presupuesto para el desarrollo de la aplicación	64



INTRODUCCION

Dentro del ámbito del desarrollo de los Sistemas de Información, el actual entorno de evolución de las nuevas tecnologías supone un reto relevante para las facultades de Ingeniería de sistemas que se suman al proceso.

El tiempo actual, tiene como uno de sus rasgos característicos la revolución que ha supuesto la ingeniería de sistemas. En esa perspectiva, una gestión a la altura de los procesos, tiene que acompañar y promover en beneficio de los ciudadanos en el uso de la ingeniería de sistemas. Al servicio del ciudadano.

Los departamentos o áreas de informática tienen un número de recursos finito. Con estos recursos deben de generar y desarrollar las nuevas necesidades de productos que se convierten muchos de ellos en el desarrollo de proyectos de software con tecnología y enfoque.

Sin embargo, hay proyectos que no acaban en presupuesto o tiempo o que no logran los resultados esperados. Una solución para asegurar una adecuada gestión de los mismos y que se cumpla con los plazos establecidos, las calidades acordadas dentro de los presupuestos estimados y la satisfacción del cliente, La adopción de una metodología de desarrollo y de calidad, las técnicas de éstas características considera como relevantes los tres vértices que siempre se han de considerar en cualquier proyecto bajo una metodología de calidad.

- Personas: Son la pieza clave del éxito. Deben estar formadas y concienciadas. El eje de Personas requiere un modelo bien definido de roles y responsabilidades.
- Procedimiento: Compendio escrito y gráfico del proceso, identificando los pasos y estados a seguir. Es importante que estos procedimientos estén basados en una metodología de desarrollo software, para abordar las líneas de trabajo que tienen relación directa con la gestión de proyectos.
- Herramientas: Simples o complejas, deben soportar el proceso.

1. PLANTEAMIENTO DEL PROBLEMA

Todos los días el departamento Logística Inversa de la empresa Grupo Amarey Nova Medical genera remisiones para atender los requisitos de las cirugías de los de las diferentes entidades de salud que se tiene como clientes, Estas remisiones deben seguir una serie de procesos para poder ser cobradas por el departamento de facturación de la empresa. El hecho de que actualmente las remisiones sean generadas de forma manual, lo cual conlleva mucho trabajo para la empresa que tienen que dedicar un gran número de recursos en el proceso de clasificación, búsqueda de documentos relacionados para poder hacer el cobro efectivo de las remisiones.

La trayectoria de una Remisiones es larga y tediosa: Desde su creación tiene que pasar por varios procesos: primero debe ser clasificada y pre-registrada manualmente en Excel, después se comprueba si coincide con el pedido, luego la validación de los valores de los materiales para el cliente y los materiales utilizados, luego la remisión es enviada al cliente, entonces comienza el trámite de las remisiones, se recibe el visto bueno del cliente, después de la cirugía se recogen las partes que no se utilizaron en el proceso quirúrgico y los adicionales como motores e instrumentos enviados, en base a lo enviado y a la novedad del material utilizado se realiza en cobro. En el departamento de facturación debe introducir los datos de las materiales usados y del cliente en el sistema contable, si todo está en orden se da paso a la creación y cobro de la factura a la entidad de salud. El último paso en este proceso, es el archivo de la Remisiones, almacenándose hasta el día en que se necesite realizar una consulta accediendo al documento físico para realizar una verificación o control.

2. JUSTIFICACIÓN

El Área de logística inversa de la empresa Grupo Amarey Novamedical, debe controlar sin duda el procedimiento que se sigue para la generación de remisiones para sus clientes. El no tener definido y con claridad lo que se remisiona, puede generar la pérdida y el recobro de facturas ocasionando molestias a los clientes y posibles restricciones o cancelaciones de la relación comercial.

Es fundamental tener una aplicación y procedimientos que certifique por escrito y describa las funciones que debe hacer cada uno de los involucrados en el proceso, pero más importante es difundirlo para que se aplique totalmente.

El Software de generación de remisiones, apoyará y manejará todo el proceso de logística inversa, iniciando desde la solicitud del material hasta la emisión y cobro de las facturas de la empresa Grupo Amarey Novamedical. El Programa permite, controlar y hacer seguimiento del proceso interno iniciando por, la recepción de la solicitud del pedido del material para cirugía, generando números de radicado, realizando la contabilización y generando el cobro de la factura, apoyando el proceso de control interno de la compañía.

Debido al arduo proceso de Remisiones, se propone la realización de un sistema de información con el cual se mejorará, optimizará y podrá llevar trazabilidad del proceso, para dar una pronta respuesta al cliente interno y externo.

En este proyecto se propone diseñar e implementar un sistema de información de remisiones que beneficie al área de logística inversa en su tarea de planificar y controlar el despacho de instrumental quirúrgico mediante la sistematización de la información. El sistema de Información de Remisiones se usará como herramienta para consultar, gestionar y verificar los datos que están referidos a un instrumental en particular.

3. OBJETIVO GENERAL

- Desarrollar un sistema de información para obtener el control de las Remisiones de la empresa Grupo Amarey

4. OBJETIVOS ESPECÍFICOS

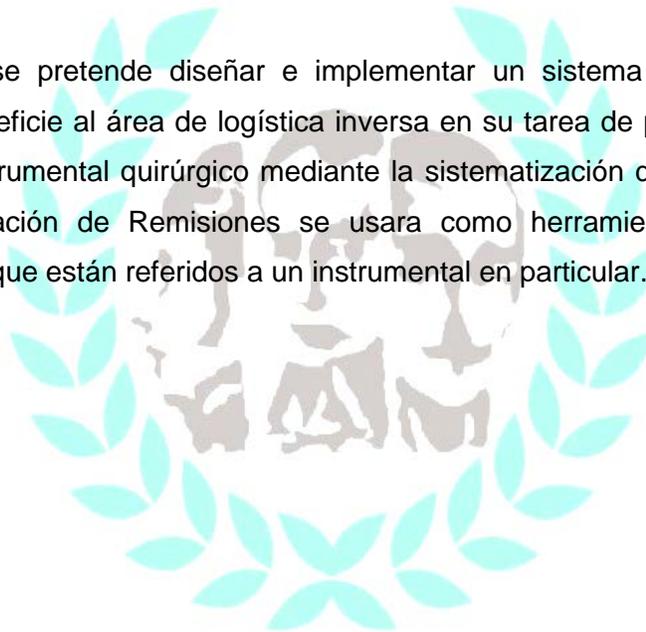
- Implementar un sistema para registrar y controlar la trazabilidad en la salida de las Remisiones a la empresa.
- Controlar el estado en el que se encuentra la Remisiones dentro del proceso
- Permitir al área de Facturación, realizar el cobro de la Remisiones de una forma segura y confiable.
- Dar una mayor facilidad a los usuarios para la consulta de las Remisiones



5. DELIMITACIÓN Y ALCANCE

Este proyecto se concreta en la definición e implantación de una plataforma de software que se ajuste a las características y necesidades del área de logística inversa de la empresa Amarey Nova medical, es decir a los programas relacionados con los sistemas operativos y componentes a utilizar, para la administración técnica, relacionadas con las tareas de control del servicio de remisiones. De esta forma se aprovecharán eficientemente los recursos.

En este proyecto se pretende diseñar e implementar un sistema de información de remisiones que beneficie al área de logística inversa en su tarea de planificar y controlar el despacho de instrumental quirúrgico mediante la sistematización de la información. El sistema de Información de Remisiones se usará como herramienta para consultar, gestionar los datos que están referidos a un instrumental en particular.



6. MARCO REFERENCIAL

6.1 MARCO DE ANTECEDENTES

Sistemas TWG desarrollo un Software Administración de Empresas, Inventarios, Negocios, Facturas, Remisiones. Remisiones con efecto automático sobre inventario y sobre módulo de remisiones por cobrar y cartera vencida.

Hacer remisiones solo necesita digitación solamente en las cantidades de los productos. Toda la información es tomada de la base datos y alimenta el formato de la remisión para permitir un documento oficial hasta de 25 líneas en segundos. Ver imagen formulario.

Factura bloqueada para proteger integridad del inventario. **NUESTRA EMPRESA**

Remisión 005

Buscar:

Cliente: TERRITORIO VISUAL Tel: 2316615

Nº: 900069368-7 CRA 47 # 53-10 Celular: 00 Fax: 000

Persona: Sra Zerada de Alvarez MEDELLIN Email: a@a.com \$212,900.00

F. expedición: 16-Feb-10 Forma de pago: 90 DIAS Fecha venc: 17-May-10 Pedido: Consultar comercial:

	Prod	Referencia	Descripción	Valor	Cantidad	Totales
1	<input type="text"/>	P2009	PAZOTTY	32500.00	1	\$32,500.00
2	<input type="text"/>	EN003	ENZDCHIALY	12500.00	1	\$12,500.00
3	<input type="text"/>	DBK 989	DB KIDS	35000.00	1	\$35,000.00
4	<input type="text"/>	DBK 985	DB KIDS	35000.00	1	\$35,000.00
5	<input type="text"/>	040-171	TESARI	7500.00	1	\$7,500.00
6	<input type="text"/>	P2009	PAZOTTY	32500.00	1	\$32,500.00
7	<input type="text"/>	P2032	PAZOTTY	57900.00	1	\$57,900.00
8	<input type="text"/>					
9	<input type="text"/>					
10	<input type="text"/>					

Imagen formulario twg

Cuando se selecciona el cliente con el cuadro combinado, toda la información del cliente alimenta la factura en la parte superior, cuando se digita la fecha y se selecciona el método de pago, la remisión muestra la fecha de vencimiento.

Cuando se selecciona un producto de la base de datos, la línea se llena automáticamente con la referencia, descripción y el precio. Tan solo se requiere digitar la cantidad en remisión para calcular el total.

En el momento de digitar la cantidad del producto, el Inventario es afectado automáticamente. Antes de hacer una remisión, podemos consultar por medio de un botón al lado derecho del cuadro combinado que nos da la información del producto incluyendo la cantidad existente en Inventario. También podemos entrar a Productos para crear productos que entramos en remisión por primera vez y no están registrados.

Si desea usar alguna línea para información o para cobrar en remisión, por ejemplo mano de obra, entonces se hace preparando la línea con 00PRODUCTO, para entrar la información y el costo fuera de Inventario.

Después de llenar las líneas se está listo para producir el reporte de la Remisión oficial de numeración consecutiva para el cliente.

El software produce tres tipos diferentes de remisiones, una pequeña de tamaño medio pliego para facturación de 12 líneas, una completa de 25 líneas con logo pre programado desde el módulo de la empresa y una remisión sin logo en caso de querer usar papel membreteado.

La gran ventaja del sistema de remisión es el hecho de que una vez terminada la remisión, esta queda bloqueada para evitar futura digitación, función que se puede sobrepasar simplemente haciendo clic en el botón para desbloquear.

En la remisión tienen dos botones de información que son útiles para conocer el funcionamiento total del sistema. Una función importante es el poder digitar en las líneas artículos que no son facturados, por ejemplo mano de obra.

Aparte de la actualización automática de Inventario no solo al hacer remisiones sino al borrar cualquier línea en remisión, el sistema nos muestra en el módulo del cliente las remisiones por cobrar, las remisiones vencidas y las remisiones canceladas si es que las cancelamos.

Esto permite llevar un Sistema de Cartera que muestra cuando una remisión se vence como Cartera Vencida de determinado cliente. También nos produce un reporte de remisiones vencidas de todos los clientes.

Una vez pagada la remisión simplemente ingresa la fecha de pago y la remisión no aparece más como remisión por pagar.

6.1.1 Las remisiones son un documento importante para el área de logística inversa de la empresa Amarey Novamedical, ya que esta permite llevar un control de los elementos enviados para las cirugías así como el despacho oportuno de las mismas, así como saber cuál fue el consumo total de los elementos enviados para su posterior facturación en base a esta información, por lo que se hace indispensable una aplicación que permite llevar este control para el beneficio del área, de la empresa y de los clientes.

En el mercado se encuentran muchas empresas como (Castelec, New Genesis Casa de Software, ABP soluciones s.a.s y sicocolombia) entre otras que ofrecen soluciones para la generación de remisiones, pero para el caso del área logística inversa en un caso espacial ya que se genera en pro de las cirugías que se programen y los materiales se rigen según el cliente y el kit que se envía para el procedimiento médico, por otro lado dependiendo del material estos tiene una cantidad estándar y pueden tener un precio o no, adicional el cobro se efectúa por el total consumido en la cirugía y no por lo enviado.

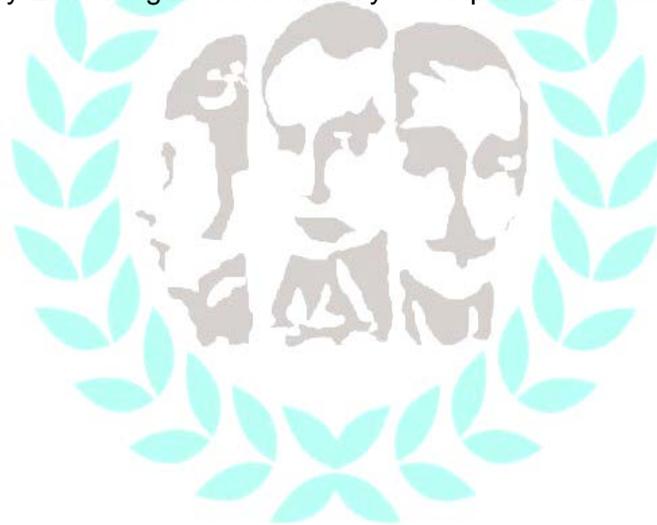
6.1.2 La Nota de Remisión “es aquel documento que se utiliza casi exclusivamente a instancias de un contexto mercantil para acreditar o dejar constancia de la entrega de un pedido. El mismo para que tenga efecto y validez deberá ser sí o sí firmado por el receptor de la mercancía hecho que dará constancia que la mercadería ha sido recibida correctamente y de acuerdo a las condiciones convenidas.

En tanto, dentro de lo que se conoce como compra venta, que puede responder a cualquier índole, libros, muebles, etc., la nota de remisión servirá como prueba documental que la entrega de bienes se hizo conforme. De lo mencionado se desprende entonces que a la Nota de Remisión se le encuentran dos objetivos diferentes, al comprador le servirá para comparar con el pedido que haya realizado, a efectos de constatar si se trata de la mercadería solicitada y para controlar posteriormente si la misma corresponde a lo que se factura.

Y para el vendedor, porque al recibir el duplicado de la nota de remisión, oportunamente firmada por el comprador, tendrá una constancia no solamente de haber entregado la mercadería y así poder facturarla, sino que además que la misma fue aceptada conforme.

La nota de remisión deberá expedirse como mínimo por duplicado y deberá contener la lista de mercadería o de elementos suministrados. Lo más frecuente es que la misma se extienda por triplicado, el original para el comprador, la copia para el vendedor y la tercera para quien transporta la mercadería a efectos que le sirva como constancia de la operación que se llevó a cabo en caso de haber reclamos de parte de algunas de las dos partes, comprador o vendedor. Las notas de remisión no son incluidas en los libros de contabilidad dado que las mismas generalmente se emiten sin valores, hecho por el cual no tienen un valor tributario, principal diferencia que guardan con respecto a las facturas y que nos permitirá diferenciar entre una y otra. Es más bien una especie de documento interno de las empresas que cumple una función de verificación.

Si bien la denominación nota de remisión se encuentra muy extendida en la mayoría de los países hispanoparlantes como ser México, Colombia, Paraguay, Bolivia, también a la misma se la conoce con otras denominaciones en América Latina, por ejemplo en Argentina se la llama remito, en Chile guía de despacho, en Costa Rica y Guatemala nota de entrega, en Perú y Ecuador guía de remisión y en República Dominicana conduce”.¹



¹ Definiciones ABC, Definición de Nota de Remisión, (2007) disponible en: <http://www.definicionabc.com/general/nota-de-remision.php> [citado el 9 de septiembre de 2015]

6.2 MARCO CONCEPTUAL

6.2.1 La logística “determina y coordina en forma óptima el producto correcto, el cliente correcto, el lugar correcto y el tiempo correcto. Si asumimos que el rol del mercadeo es estimular la demanda, el rol de la logística será precisamente satisfacerla. Solamente a través de un detallado análisis de la demanda en términos de nivel, locación y tiempo, es posible determinar el punto de partida para el logro del resultado final de la actividad logística, atender dicha demanda en términos de costos y efectividad. La logística no es por lo tanto una actividad funcional sino un modelo, un marco referencial; no es una función operacional, sino un mecanismo de planificación; es una manera de pensar que permitirá incluso reducir la incertidumbre en un futuro desconocido”.²

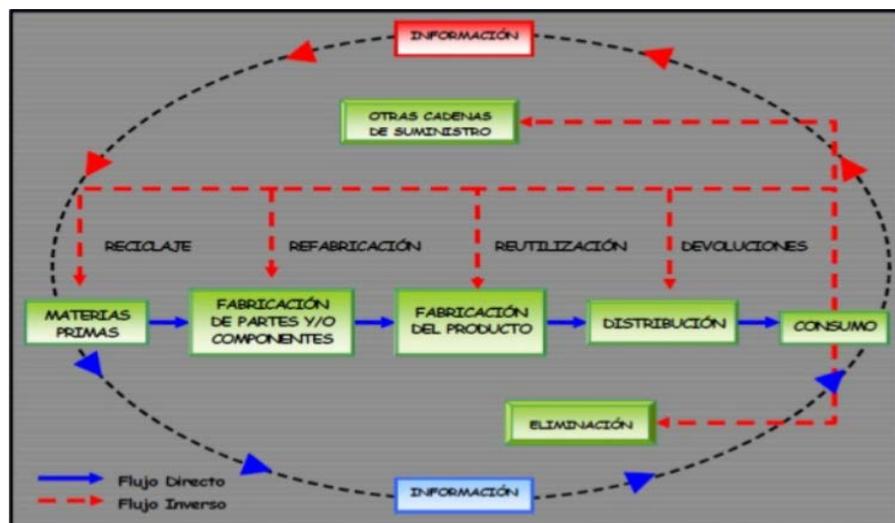
Las actividades claves son las siguientes:

- Servicio al cliente.
- Transporte
- Gestión de Inventarios.
- Procesamiento de pedidos.

6.2.2 La logística inversa es el proceso de planificación, implantación y control eficiente del flujo de costos y almacenamiento de materia prima, productos en proceso y productos terminados o inventarios, así como de la información concerniente, desde el punto de consumo al punto de origen, con el fin de recuperar valor o asegurar su correcta eliminación. Se puede definir también como un macro proceso en el cual se planifica, administra y controla el flujo de los productos y materiales desde el lugar de consumo hasta el lugar de origen del fabricante o proveedor con el propósito de crear valor económico, ecológico, legal o de imagen y generar satisfacción a los clientes”.³

Imagen 1. Logística inversa

Fuente: [slideshare.net/scourge/logistica-inversa-17191095](https://www.slideshare.net/scourge/logistica-inversa-17191095)



Integración de la red de logística tradicional y logística inversa



Imagen 2. Integración logística

Fuente: <http://www.revistadelogistica.com/La-logistica-reversa-o-inversa.asp>

² Monografias.com, La logística inversa, (2012) disponible en:
<http://www.monografias.com/trabajos15/logistica/logistica.shtml> [citado el 9 de septiembre de 2015]

³ Gestiopolis, Cadena de suministros y la logística inversa, (2013) disponible en:
<http://www.gestiopolis.com/cadena-de-suministros-y-la-logistica-inversa/> [citado el 9 de septiembre de 2015]

6.2.3 La empresa Amarey Nova Medical S.A. cuenta con 30 años de experiencia en el mercado farmacéutico colombiano. Especializada en distribución, comercialización y venta de productos elaborados por reconocidas casas farmacéuticas.

- **Misión:** Generamos valor sirviendo a la salud, apoyados en nuestra gente y orientados por nuestros valores.
- **Visión:** Ser el grupo empresarial que en constante innovación, brinda las mejores soluciones integrales para la salud

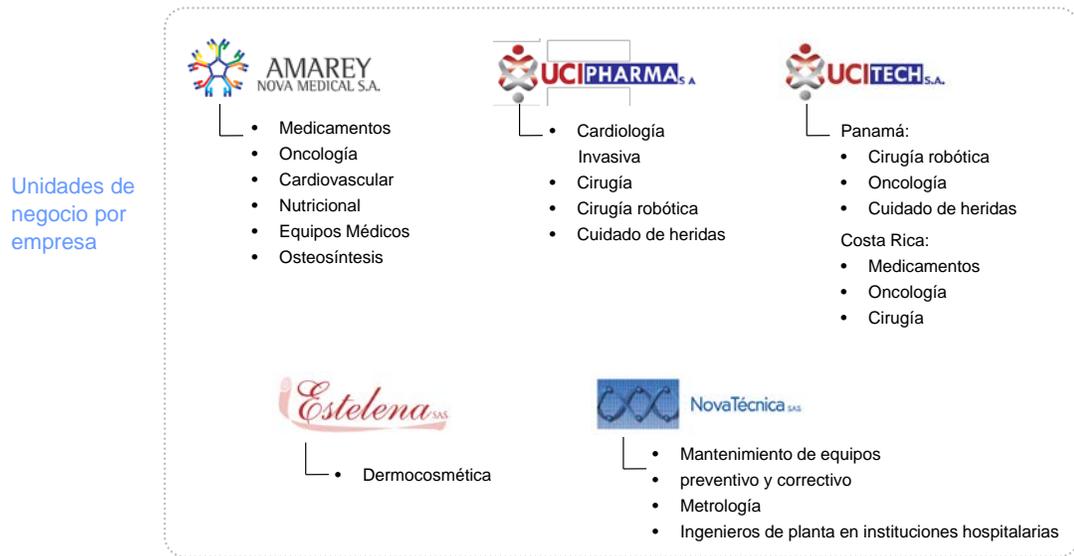


Imagen 3. Empresa Amarey Nova Medical s.a

En el área de logística inversa de la empresa Amarey Nova medical, se cuenta con la siguiente caracterización, ver (Imagen 4 y 5):

	CARACTERIZACIÓN LOGÍSTICA INVERSA	CA-DE-LI-G-001-V3
		01-AGO-14
		PAGINA 1 DE 6
Objetivo: Atender adecuada y oportunamente los requerimientos de cirugía de las líneas de trauma, columna, artroscopia, reemplazos articulares, cardiovascular y CMF (Craneomaxilofacial) cumpliendo con los tiempos de entrega y con los estándares de lavado, almacenamiento, preparación y entrega de instrumental y dispositivos médicos de acuerdo al modelo de logística inversa establecido por el Grupo Amarey.		Responsables: Jefes de logística inversa (trauma, CMF y otras líneas) Coordinador de facturación e inventarios.



Imagen 4.

Fuente: CA-DE-LI-G-001-V3 CARACTERIZACIÓN LOGISTICA INVERSA.pdf

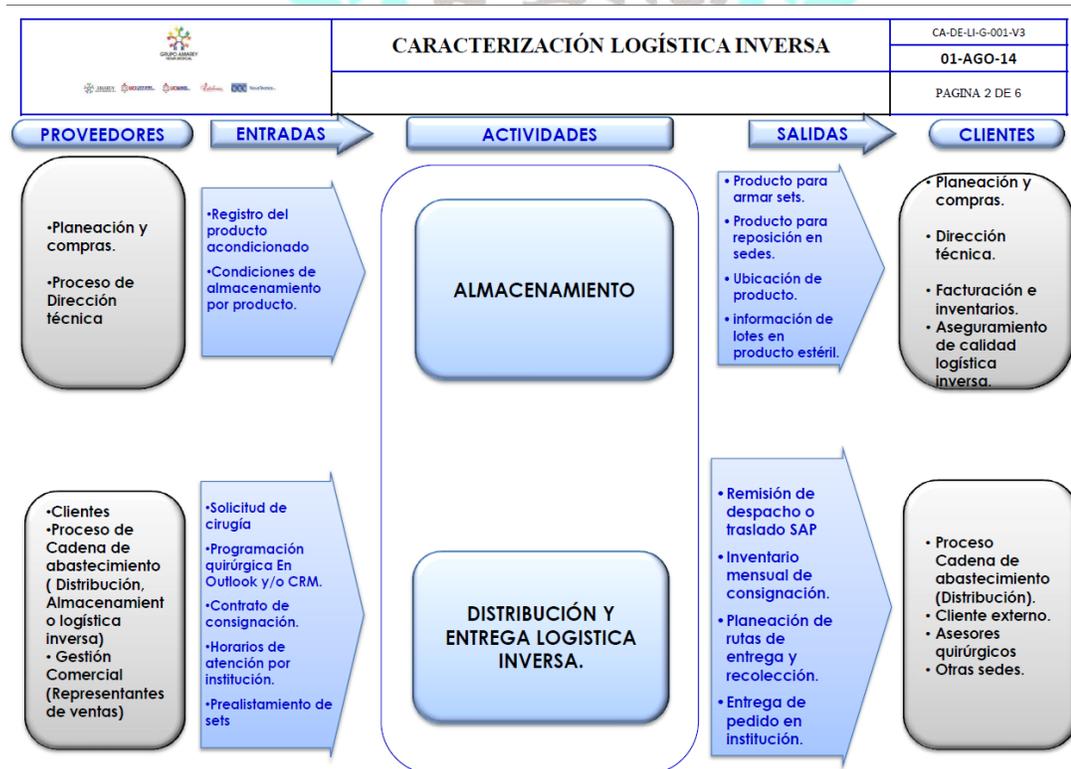




Imagen 5.

Fuente: CA-DE-LI-G-001-V3 CARACTERIZACIÓN LOGISTICA INVERSA.pdf



6.3 MARCO TEÓRICO

6.3.1 Las Metodologías Ágiles

Para el desarrollo del proyecto se adoptara uno de los tipos de metodología ágil (xp), debido a que es modelo rápido y se ajuste a los tiempos de entrega definidos por el cliente y dar respuesta frente a cambios que se pueden presentar en el seguimiento del plan.

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos. Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software, y a la vez generando un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativa para las metodologías tradicionales. En este trabajo se presenta resumidamente el contexto en el que surgen las metodologías ágiles, sus valores, principios y comparaciones con las metodologías tradicionales. Además se describe con mayor detalle Programación Extrema (eXtreme Programming, XP) la metodología ágil más popular en la actualidad. ⁴

6.3.2 Las Aplicaciones Web

El concepto de la aplicación web no es nuevo. De hecho, uno del primer lenguaje de programación para el desarrollo de aplicaciones web es el "Perl". Fue inventado por Larry Wall en 1987 antes de que internet se convirtiera en accesible para el público en general.

Pero fue en 1995 cuando el programador Rasmus Lerdorf puso a disposición el lenguaje PHP con lo que todo el desarrollo de aplicaciones web realmente despegó. Hoy en día, incluso muchas de estas aplicaciones se han desarrollado en PHP, como Google, Facebook y Wikipedia.

Luego vino la famosa plataforma Flash utilizada para añadir contenido interactivo para sitios Web. Flash hizo su aparición en 1997, conocido como Shockwave Flash. Más tarde, después de ser adquirido por Macromedia y Adobe, Flash se convirtió en una plataforma para desarrollar aplicaciones web interactivas.

El año siguiente marcó un punto de inflexión para los medios de comunicación en línea. De hecho, el 17 de enero de 1998, el sitio web The Drudge Report anunció por primera vez un informe de noticias antes de que se difundiera en los medios de televisión y la prensa tradicional. Se informó el escándalo Clinton/Lewinsky. Este evento fue el detonante del periodismo en línea tal como lo conocemos hoy en día. Antes de esa fecha, internet nunca había sido considerado un medio de comunicación más importante.

El concepto de "web como plataforma" fue mencionado por primera vez. Esta innovación allanó el camino para futuras aplicaciones web, es decir, un software que aprovecha las ventajas de la conexión a internet y que se desvía del uso tradicional del escritorio. En segundo lugar, el sitio interactivo de Digg se puso en marcha. Propuso una forma innovadora de crear y encontrar contenido en internet mediante la promoción de noticias y enlaces democráticamente votado por los usuarios. Y, por último, el tercer gran evento, pero no menos importante, fue el lanzamiento de Facebook, que estaba entonces en su infancia, abierto sólo a los estudiantes. Con un millón de suscriptores a finales de 2004, Facebook se ha convertido en el medio de comunicación más utilizado con más de 900 millones de usuarios. Este es el segundo sitio más visitado en el planeta y tiene la mayor cantidad de fotos compartidas por los usuarios con un total de casi 500000 millones de fotos subidas a la plataforma. Facebook ha revolucionado la miríada de aspectos relacionados con la vida social, la comercialización, y la política... en la Web.

En 2005, YouTube fue lanzado oficialmente, permite a los usuarios compartir vídeos en línea. De simple sitio para compartir vídeos en internet a una plataforma madura que se conoce hoy en día, YouTube ahora ofrece cerca de 4000 millones de vídeos al día, además de un servicio de alquiler de películas en línea, y, finalmente, episodios de emisión para las empresas o las películas de MGM, Lions Gate Entertainment y CBS.

Puesto que en los 43 años de la historia de internet, los programadores han tratado de hacer estallar las barreras entre las aplicaciones tradicionales y aplicaciones web. El progreso de los últimos años en tecnología, velocidad de descarga, así como herramientas de desarrollo (incluyendo la plataforma .NET de Microsoft) aborda algunas deficiencias, las aplicaciones Web se están convirtiendo en una funcionalidad más completa, mientras que están siendo fáciles de usar.⁵

6.3.3 Las Bases de Datos

Las bases de datos son recursos que recopilan todo tipo de información, para atender las necesidades de un amplio grupo de usuarios. Su tipología es variada y se caracterizan por una alta estructuración y estandarización de la información.

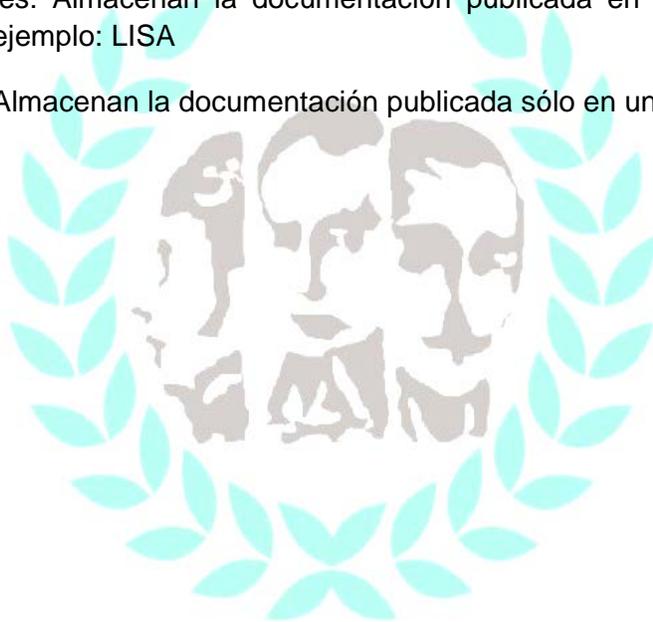
- Componentes básicos de una base de datos
 - Documentos, constituyen la entidad físico/cognitiva compleja que alberga la estructura formal, basada en los datos físicos necesarios para su identificación (título, autor, lugar de publicación, fecha, edición) y la estructura lógico-cognitiva, centrada en el contenido y en las propiedades semánticas.
 - Representación de documentos, tanto de sus propiedades físicas como semánticas se hace mediante palabras clave, frases, etc. que servirán de puntos de acceso cuando interroguemos al sistema.
 - Necesidades de información de los usuarios, manifestadas en la solicitud de información.
 - Representación de las necesidades de información, expresadas también con palabras clave o frases.
 - Comparación de la representación de información con la representación de los documentos. Las bases de datos, basadas en la función semejanza compara, a través de un índice, ambas representaciones para seleccionar los documentos relevantes.

- Clasificación

Según la naturaleza de la información contenida en las bases de datos, se distinguen dos grandes grupos:

- Bases de datos referenciales: La información que contiene es muy estructurada principalmente a través de tablas. Con éstas se pueden establecer relaciones que pueden dar lugar a nuevas tablas o bases de datos. Ejemplo de este tipo de base de datos es el programa Access, Oracle, etc.
- Bases de datos documentales: Los registros que componen la base de datos se relaciona con los documentos almacenados. Su organización se estructura de la siguiente manera: un registro se relacionan con un número de identificación del documento original, y se puede acceder a éste mediante los distintos campos.

- Dependiendo del tipo de cobertura temática pueden ser:
 - Bases de datos multidisciplinarias: la documentación almacenada abarca distintas disciplinas científicas como es el caso de TESEO (tesis de todas las disciplinas)
 - Bases de datos especializadas: la documentación almacenada abarca sólo una disciplina como es el caso de Medline (sólo documentación relacionada con Medicina)
- Según la cobertura geográfica:
 - Internacionales: Almacenan la documentación publicada en cualquier lugar del mundo. Por ejemplo: LISA
 - Nacionales: Almacenan la documentación publicada sólo en un ámbito nacional. ⁶



⁴ cyta.com.ar, Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP), (2006), ISSN 1666-1680, disponible en: http://www.cyta.com.ar/ta0502/b_v5n2a1.htm [citado el 12 de septiembre de 2015]

⁵ um.es, Artículos. Informática. Servicios en internet Historia del desarrollo de aplicaciones Web;,(2012), Rafael Barzanallana. Universidad de Murcia, disponible en: <http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Historia-desarrollo-aplicaciones-web.html> [citado el 15 de septiembre de 2015]

⁶ Maria Pinto Molina, Electronic content management skills: Bases de datos, (2004), disponible en http://www.mariapinto.es/e-coms/bases_datos.htm [citado el 12 de septiembre de 2015]

7. MARCO METODOLOGICO

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

- El cliente define el valor de negocio a implementar.
- El programador estima el esfuerzo necesario para su implementación.
- El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- El programador construye ese valor de negocio.
- Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

- **Fase I: Exploración**

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

- **Fase II: Planificación de la Entrega**

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la "velocidad" de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

- **Fase III: Iteraciones**

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción.

Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores. Wake en [18] proporciona algunas guías útiles para realizar la planificación de la entrega y de cada iteración.

- **Fase IV: Producción**

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de mantenimiento).

- **Fase V: Mantenimiento**

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

- **Fase VI: Muerte del Proyecto**

Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.⁷



⁷ cyta.com.ar, Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP), (2006), ISSN 1666-1680, disponible en: http://www.cyta.com.ar/ta0502/b_v5n2a1.htm [citado el 12 de septiembre de 2015]

7.1 Selección de Herramientas.

7.1.1 Selección Motor de Bases de datos.

Como un objetivo de este proyecto se propuso evaluar diferentes Bases de Datos, para observar cual es la base de datos que más se acomodaba a las necesidades y a las exigencias de la aplicación que se desarrolló. Entre las bases de datos más usadas se encuentran: Postgres, MySQL y SQL Server, escogiendo esta última como el motor de base de datos para la aplicación por las ventajas que se consideraron tenía sobre los otros motores de bases de datos.

El Motor de base de datos de SQL Server es el servicio principal para almacenar, procesar y proteger datos. El Motor de base de datos proporciona acceso controlado y procesamiento de transacciones rápido para cumplir con los requisitos de las aplicaciones consumidoras de datos más exigentes de su empresa.

Se usa el Motor de base de datos para crear bases de datos relacionales para el procesamiento de transacciones en línea o datos de procesamiento analítico en línea. Esto incluye la creación de tablas para almacenar datos y objetos de base de datos (p.ej., índices, vistas y procedimientos almacenados) para ver, administrar y proteger datos. Se puede usar SQL Server Management Studio para administrar los objetos de bases de datos y SQL Server Profiles para capturar eventos de servidor.

Las ventajas son

- Es un Sistema de Gestión de Bases de Datos Relacionales (SGBDR).
- Puede ser útil para manejar y/o obtener datos de la red de redes...
- Nos permite olvidarnos de los ficheros que forman la base de datos.
- SQL permite administrar permisos a todo. También permite que alguien conecte su SQLO al del proyecto pero sin embargo podemos decirle que no puede ver esta base de datos pero otro sí.
- Ofrece una potente forma de unir SQL e Internet.
- Utiliza una extensión al SQL estándar, que se denomina Transact SQL.
- El Transact SQL, soporta la definición, modificación y eliminación de bases de datos, tablas, atributos, índices, etc., es decir, el lenguaje de definición de datos (LDD), así como la consulta, actualización y borrado de tuplas de tablas, es decir, el lenguaje de manipulación de datos (LMD).

- Para empezar SQL Server nos permite olvidarnos un poco de los ficheros que forman la base de datos. En Access tenemos que abrir el fichero .mdb para ver el contenido. Aquí abrimos la consola de SQL Server y tenemos todas las bases de datos de todos nuestros programas. Esto es muy cómodo, ciertamente.
- Si trabajamos en una red local nos permite agregar otros servidores de SQL Server. por ejemplo: yo tengo en mi ordenador una base de datos en mi servidor SQL y mi compañero tiene en su servidor otra base de datos. Yo puedo conectar mi SQL al suyo y así tengo sus bases de datos en mi SQL Server. Esto, también es muy cómodo.

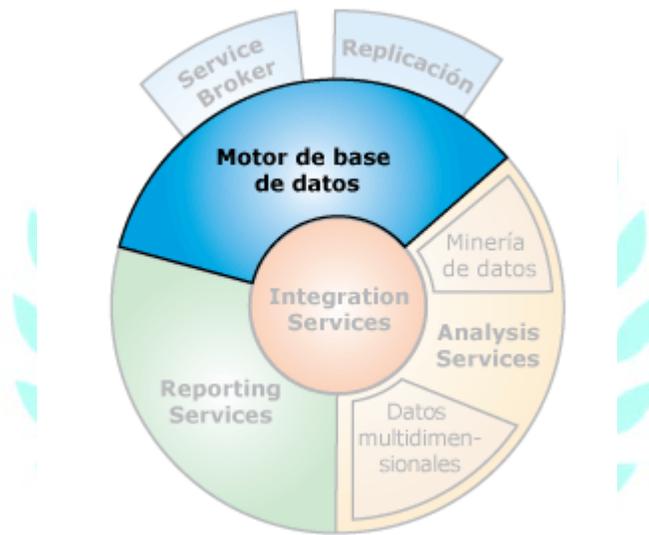


Imagen 6. Motor base de datos

Referencia: <https://i-technet.sec.s-msft.com/dynimg/IC401338.gif>

7.1.2 Selección herramienta de desarrollo para aplicación web

Existen numerosas herramientas para el desarrollo de aplicaciones Web, entre los que destacan: PHP, Java y ASP/ASP.NET, escogiendo PHP con CodeIgniter como herramienta de desarrollo por sus múltiples ventajas, CodeIgniter es un framework para el desarrollo de aplicaciones en php que utiliza el MVC. Permite a los programadores Web mejorar la forma de trabajar y hacerlo a mayor velocidad.

- MVC: El Modelo Vista Controlador es un estilo de programación en el que la aplicación está dividida en 3 capas:

- Modelo: es dónde se procesa y obtienen los datos, la conexión con la bd.
- Vista: presenta los datos en pantalla, es donde va el código HTML.
- Controlador: controla los datos, dicho de forma rápida obtiene datos de un modelo, los procesa, y se los pasa a la vista.

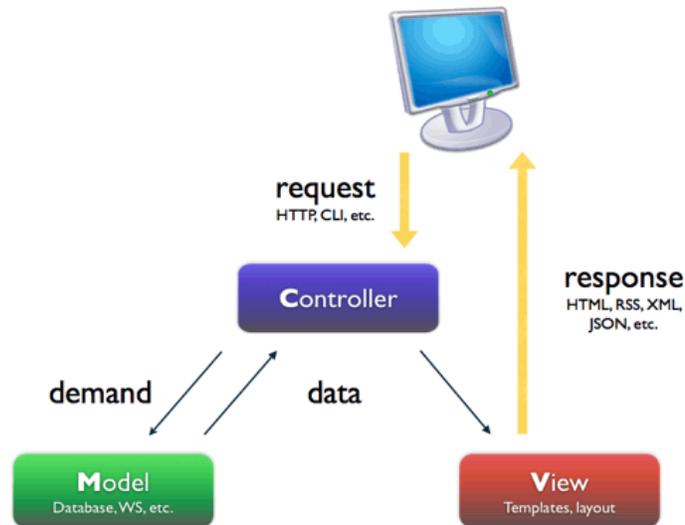


Imagen 7. Mvc

Referencia: <http://1.bp.blogspot.com/-BZWfWVdX34o/UGZC-v61GGI/AAAAAAAAABgg/RfSp81Fp6m0/s1600/MVC.png>

Ventajas de utilizar un framework como CodeIgniter:

- Las páginas se procesan más rápido, el núcleo de CodeIgniter es bastante ligero.
- Es sencillo de instalar, basta con subir los archivos al ftp y manipular un archivo de configuración para definir el acceso a la bd.
- Reutilización de código, desarrollo ágil.
- Existe abundante documentación en la red.
- Facilidad de edición del código ya creado.
- Facilidad para crear nuevos módulos, páginas o funcionalidades.

- Acceso a librerías públicas y clases. Entre otras, hay librerías para el login, paginador, calendarios, fechas.
- Estandarización del código. Fundamental cuando hay que tocar código hecho por otra persona o cuando trabaja más de una persona en un mismo proyecto.
- URLs amigables con SEO.
- Separación de la lógica y arquitectura de la web, el MVC.
- CodeIgniter es menos rígido que otros framework. Define una manera de trabajar.
- Cualquier servidor que soporte PHP+SQL sirve para CodeIgniter.
- CodeIgniter se encuentra bajo una licencia open source, es código libre.
- CodeIgniter usa una versión modificada del Patrón de Base de Datos Active Record. Este patrón permite obtener, insertar y actualizar información in tu base de datos con mínima codificación. Permite consultas más seguras, ya que los valores son escapadas automáticamente por el sistema.

7.1.3 Selección de la Metodología Ágil

XP (Extreme Programming) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, Promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los Desarrolladores, y propiciando un buen clima de trabajo. XP se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación Fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos muy cambiantes y donde existe un alto riesgo técnico.⁶

⁶ cyta.com.ar, Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP), (2006), ISSN 1666-1680, disponible en: http://www.cyta.com.ar/ta0502/b_v5n2a1.htm [citado el 12 de septiembre de 2015]

- Tabla 1, Comparación Metodologías Ágiles y Tradicionales.

Metodología Ágil	Metodología Tradicional
Pocos Artefactos. El modelado es prescindible, modelos desechables.	Más Artefactos. El modelado es esencial, mantenimiento de modelos
Pocos Roles, más genéricos y flexibles	Más Roles, más específicos
No existe un contrato tradicional, debe ser bastante flexible	Existe un contrato prefijado
Cliente es parte del equipo de desarrollo (además in-situ)	El cliente interactúa con el equipo de desarrollo mediante reuniones
Orientada a proyectos pequeños. Corta duración (o entregas frecuentes), equipos pequeños (< 10 integrantes) y trabajando en el mismo sitio	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos
La arquitectura se va definiendo y mejorando a lo largo del proyecto	Se promueve que la arquitectura se defina tempranamente en el proyecto
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo	Énfasis en la definición del proceso: roles, actividades y artefactos
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Se esperan cambios durante el proyecto	Se espera que no ocurran cambios de gran impacto durante el proyecto

Referencia: http://www.cyta.com.ar/ta0502/b_v5n2a1.htm, Comparación Metodologías Ágiles y Tradicionales, Tabla 2. Diferencias entre metodologías ágiles y no ágiles.

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos y criterios para su uso para el desarrollo del software.

Pasos:

- El cliente define el valor de negocio a implementar.
- El programador estima el esfuerzo necesario para su implementación.
- El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- El programador construye ese valor de negocio.

Criterios:

- Software que funcione es más importante que documentación exhaustiva, es decir si el software no funciona los documentos no valen de nada.
- La colaboración con el cliente es más importante que la negociación de contratos.
- El éxito de los proyectos se basa en una realimentación permanente La respuesta ante el cambio es más importante que el seguimiento de un plan.
- Los clientes estuvieron involucrados directamente con el proyecto todo el tiempo.

El proyecto fue realizado bajo prácticas que fomentaron la comunicación, tanto cliente-desarrolladores como entre desarrolladores-desarrolladores, sencillez, el proyecto se realizaba según las necesidades del módulo que se fuera implementando, realimentación, lo que permitía corregir errores inmediatos para evitar complicaciones a futuro, y valentía que se refiere a la aceptación la cambio, ya que los requerimientos fueron variando a lo largo del proyecto, era necesario estar dispuestos a aceptar dichos cambios.

7.2 Proceso de Desarrollo

7.2.1 Los Roles

Programador: Ángel Andrés pulido Suarez, escribió las pruebas unitarias y produjo el código del sistema.

- Cliente: el desarrollador del proyecto y un integrante del área de logística inversa, escribieron las historias de usuario y las pruebas funcionales para validar su implementación. Pero sólo el desarrollador asignó la prioridad a las historias de usuario y decidió cuáles se implementaron en cada iteración.
- Encargado de pruebas (Tester): El integrante del área de logística inversa ayudó a escribir las pruebas funcionales. Ejecutó las pruebas regularmente, e informó los resultados y apreciaciones al equipo de desarrollo.
- Encargado de seguimiento (Tracker): El integrante del área de logística inversa proporcionó realimentación al equipo y realizó el seguimiento del progreso de cada iteración.
- Entrenador (Coach): Miguel Armando Hernández Bejarano fue el responsable del proceso global. Fue el encargado de proveer guías al equipo de forma que se apliquen las prácticas XP y se siguiera el proceso correctamente.

7.2.2 Planificación

7.2.2.1 Historias de usuarios.

Tipos de requerimientos.

- **RI: Requerimiento de interfaz.**
- **RNF: Requerimiento no funcional.**
- **RF: Requerimiento funcional.**

- **Requerimientos de interfaz**

El sistema será una aplicación interactuara con sistemas externos y por tal razón se implementan interfaces con otras aplicaciones (SAP).

- **Requerimientos no funcionales**

Los requerimientos no funcionales son impuestos por el cliente y detallan comportamientos del sistema en los que la interacción del usuario final no afecta directamente el funcionamiento de este, un claro ejemplo es el motor de base de datos, cantidad mínima de usuario que debe soportar la aplicación, lenguaje de programación entre otras. La Tabla 2 describe los requerimientos no funcionales solicitados por cliente incluyendo un edificador, la descripción, fecha de captura y prioridad del requerimiento.

Tabla 2. Requerimientos no funcionales

Código del Requerimiento	Descripción	Fecha	Prioridad
RNF-001		16/08/2015	ALTA
	La aplicación debe ser web para permitir su utilización de forma descentralizada.		
RNF-002		16/08/2015	ALTA
	La aplicación debe trabajar de forma dependiente va a interactuar con alguna de las aplicaciones existente en la empresa Amarey nova medical		
RNF-003		16/08/2015	ALTA
	La aplicación debe tener una base de datos en SQL.		

RNF-004			
	No se debe borrar información de la base de datos por cuestión de seguridad y de acceso al historial de datos, se debe tener un campo que indique la visibilidad del dato.	16/08/2015	ALTA
RNF-005			
	Toda la aplicación debe tener una interfaz gráfica en lenguaje español.	16/08/2015	MEDIA
RNF-006			
	La información almacenada en la aplicación debe ser coherente	16/08/2015	ALTA
RNF-007			
	La aplicación deberá ser sometida a una serie de pruebas para garantizar que esté acorde a los requerimientos solicitados por el cliente.	16/08/2015	ALTA
RNF-008		16/08/2015	ALTA

	<p>El código fuente de la aplicación debe tener buenas prácticas de desarrollo de software para garantizar su buen despliegue y mantenimiento.</p>	
--	--	--

- **Requerimientos funcionales**

Los requerimientos funcionales detallan las interacciones del sistema con los usuarios y en algunos casos con otros sistemas externos, en la Tabla 3 se muestra la plantilla que se utilizó para tabular dichos requerimientos, en esta se capturan minuciosamente los requerimientos del cliente, especificando datos básicos como nombre del proyecto, nombre y fecha en que se capturo le requerimiento, entre otros, además se realiza un detallado de los datos requeridos, resultados esperados, restricciones y funcionalidades específicas del requerimiento. En las Tablas 4 a la Tabla 23 se describen los requisitos funcionales.

Tabla 3. Plantilla de Historia de usuarios

PROYECTO REMISIONES		Código
Especificación de requerimientos		Código del requerimiento
Nombre	Nombre del requerimiento	Fecha
		Fecha de captura del requerimiento
Entradas	Datos requeridos del requerimiento	
Salidas	Resultados del proceso de las entradas	
Restricciones	Restricciones de la salida	
Proceso		
Descripción detallada de las funcionalidades del requerimiento		

- El requerimiento RF-001, trata de la autenticación de los usuarios en la aplicación, en general el sistema debe ofrecer un formulario de autenticación y recuperación de contraseña, la información detalla del requerimiento se muestra en la Tabla 4.

Tabla 4. Requerimiento autenticación de usuarios

PROYECTO REMISIONES		Código
Especificación de requerimientos		RF-001
Nombre	Autenticar usuarios de la aplicación	Fecha
		24/08/2015
Entradas	Nombre y Clave de usuario.	
Salidas	Usuario ingresa al sistema.	
Restricciones	Los campos son obligatorios. El usuario no podrá ingresar a la aplicación si no está creado previamente.	
Proceso		
<p>El sistema debe mostrar un formulario donde se autentique , en caso de autenticación el sistema debe validar la dupla usuario y clave, si es correcta debe permitir el acceso a la aplicación, en caso de que el usuario necesite recuperar la clave el sistema debe ofrecer la opción de recuperar contraseña por el nombre de usuario o dirección de correo electrónico, luego de que el usuario ingrese alguna de las opciones el sistema debe validar la existencia del usuario asociado a esto datos, si es válida la información el sistema debe enviar un correo electrónico al usuario con la información de su cuenta.</p>		

- El requerimiento RF-002, consisten principalmente en la gestión y administración de usuarios de la aplicación; el sistema debe ofrecer un formulario en el cual se pueda crear y editar los datos de los usuarios de la aplicación, la información detalla del requerimiento se muestra en la Tabla 5.

PROYECTO REMISIONES		Código
Especificación de requerimientos		RF-002
Nombre	Administrar Usuarios	Fecha
		24/08/2015
Entradas	Nombre de usuario, nombre y correo electrónico	
Salidas	Usuario creado o editado en el sistema.	
Restricciones	<p>Los campos nombre de usuario Y correo electrónico son obligatorios.</p> <p>El nombre y la dirección de correo son únicos en la aplicación. Este formulario solo está habilitado para el administrador.</p>	
Proceso		
El administrador puede ingresar los datos de usuario para ser creado o editado, el sistema valida los datos, si el usuario selecciona la opción de guardar, el sistema crea o edita los datos del usuario en la base de datos y confirma la transacción.		

- El requerimiento RF-003, consiste en la asignación de roles a un usuario de la aplicación, existen tres roles que son administrador y general, el software debe ofrecer un formulario donde se pueda asignar y desasignar los roles a un usuario por parte del administrador de la aplicación, la información detalla del requerimiento se muestra en la Tabla 5.

Tabla 5. Requerimiento asignación de roles

PROYECTO REMISIONES		Código
Especificación de requerimientos		RF-003
Nombre	Asignar Roles de Usuarios.	Fecha
		24/08/2015
Entradas	Id del usuario, tipos de roles.	
Salidas	Usuario asociado a un rol.	

Restricciones	El usuario debe estar creado previamente
Proceso	
El administrador puede asignar y desasignar los roles de usuario, se debe mostrar una lista de los roles existentes tales como administrador y general el sistema debe guardar esta asociación en la base de datos y confirmar la transacción.	

- El requerimiento RF-04, describe la gestión y administración de los Kits y sus elementos, esta se debe realizar a través de un formulario donde pueda crear y actualizar los Kits y sus elementos, la información detalla del requerimiento se muestra en la Tabla 6.

PROYECTO REMISIONES		Código
Especificación de requerimientos		RF-004
Nombre	Administrar kits y elementos	Fecha
		24/08/2015
Entradas	Nombre del kit, cargar los elementos asociados al kit	
Salidas	Materiales asociados a los kits	
Restricciones	El nombre del Kit es único.	
Proceso		
El administrador puede crear, editar y visualizar los datos de los kits y los Materiales asociados a ella. El sistema guarda la información en la base de datos y confirma la transacción.		

Tabla 6. Requerimiento Administración de Kits y Materiales.

- El requerimiento RF-005, trata de la gestión y administración de los datos básicos del Kit como los son descripción del material o eliminación de un material del kit, el sistema debe ofrecer un formulario para la eliminación y edición de los materiales, la información detalla del requerimiento se muestra en la Tabla 7.

PROYECTO REMISIONES		Código
Especificación de requerimientos		RF-005
Nombre	Administrar Kit	Fecha
		24/08/2015
Entradas	Descripción del Material. Elementos del Kit	
Salidas	Eliminación del material asociado.	
Restricciones	El nombre del kit es único. Los materiales deben estar creados previamente Un material solo puede pertenecer a varios kits	
Proceso		
El Administrador puede eliminar, editar y visualizar los Kits; se debe crear el nombre del Kit, se asocian varios materiales. El sistema debe validar la información y las asociaciones del kit si son válidas son guardadas en la base de datos y se confirma la transacción.		

Tabla 7. Administración de Kits

- El requerimiento RF-006, trata de la gestión y administración de los datos básicos de la remisión como los son Entidad (nombre pagador), Kit y motor si este aplica, el sistema debe ofrecer un formulario para la creación de las remisiones, la información detalla del requerimiento se muestra en la Tabla 8.

PROYECTO REMISIONES		Código
Especificación de requerimientos		RF-006
Nombre	Creación de las remisiones	Fecha
		24/08/2015
Entradas	Identificador de los Materiales	
Salidas	Institución al que se le realiza la Remisión Creación de las remisiones	

Restricciones	Los valores por defecto asignados a la remisión son Entidad, kit y motor, administrador tiene acceso a todas las remisiones El usuario tiene acceso a todos los kits que le han sido creados
Proceso	
El administrador o el usuario general pueden crear las remisiones, se selecciona una Entidad, kit y a continuación se muestran los materiales asociados a este, el usuario selecciona los elementos y luego genera la remisión. El sistema guarda la información del estado y confirma la transacción.	

Tabla 8. Creación de las remisiones

- En el requerimiento RF-007, el sistema debe ofrecer un formulario para la consulta de los datos básicos de la remisión, la información detalla del requerimiento se muestra en la Tabla 9.

PROYECTO REMISIONES		Código
Especificación de requerimientos		RF-007
Nombre	Datos Adicionales de la remisión	Fecha
		24/08/2015
Entradas	Destinatario, paciente, doctor y fecha cirugía	
Salidas	Visualización de los datos asociados a la remisión	
Restricciones	El administrador tiene acceso a todas las remisiones.	
Proceso		
El usuario selecciona un destinatario e ingresa los datos de historia clínica, paciente , doctor y fecha de la cirugía el sistema muestra los datos básicos del kit y valor en la pantalla, se genera pdf de la remisión		

Tabla 9. Datos Adicionales de la remisión

- En el requerimiento RF-008, el sistema debe ofrecer un formulario para la consulta de reportes del historial de remisiones, la información detalla del requerimiento se muestra en la Tabla 10

PROYECTO REMISIONES		Código
Especificación de requerimientos		RF-008
Nombre	Consultar reporte de historial de remisiones.	Fecha 24/08/2015
Entradas	Nombre del paciente, numero de remisión, paciente, historia clínica, paciente, doctor, fecha creación, fecha cirugía y cliente	
Salidas	Reporte del historial de la remisión en formato PDF	
Restricciones	El administrador y usuario que puede generar este reporte.	
Proceso		
El usuario selecciona un cliente, ingresa la fecha creación y la fecha de cirugía del filtro asociado a la fecha de creación de las remisiones, a continuación selecciona consultar el sistema generar un reporte con la o las remisiones, luego selecciona ver pdf, se descarga la remisión en formato PDF que contiene los datos básicos.		

Tabla 10. Consultar reporte de historial de remisiones

- En el requerimiento RF-009, el sistema debe actualizar la información automáticamente la información de clientes, materiales y precios, la información detalla del requerimiento se muestra en la Tabla 11.

PROYECTO REMISIONES		Código
Especificación de requerimientos		RF-009
Nombre	Actualización de la información básica	Fecha 24/08/2015
Entradas	archivo plano de SAP con la información básica de la base de datos	
Salidas	actualización y creación de la información de cliente, actualización y creación de nuevo materiales y actualización de precios	

Restricciones	El administrador es el único que puede validar esta información
Proceso	
el sistema mediante una interface con SAP, actualiza o inserta la información en la base de datos del proyecto mediante un archivo plano que se carga automáticamente cada 5 minutos	

Tabla 11. Actualización de la información básica

- En el requerimiento RF-010, el sistema debe permitir la carga masiva de los elementos que conforman un kit, la información detalla del requerimiento se muestra en la Tabla 12.

PROYECTO REMISIONES		Código
Especificación de requerimientos		RF-010
Nombre	Actualización de la información masiva del kit	Fecha
		24/08/2015
Entradas	archivo xls la información básica de los materiales del kit	
Salidas	actualización y creación de la información del kit	
Restricciones	El administrador es el único que puede cargar esta información	
Proceso		
El usuario por medio de una opción en el sistema, cargara en una plantilla en Excel la información de los elementos que conforman el kit, (material, ubicación, cantidad, si es material o no) para actualizar o complementar un kit.		

Tabla 12. Actualización de la información masiva del kit

7.2.2.2 Pruebas

XP enfatiza en la realización de un sin número de pruebas a lo largo del proyecto, con el fin de asegurar en todo momento la realización de lo planteado en el diseño. En este proceso no sólo participa el equipo de desarrollo, también es importante los aportes del cliente, sobre todo en las pruebas de aceptación. Cabe señalar que el diseño de pruebas se realiza para todas las partes del sistema como una práctica para garantizar el buen funcionamiento, independiente de la decisión que se tome sobre implementarlas en un framework como **CodeIgniter**.

- Pruebas unitarias: lo que dice XP Las pruebas deben ser escritas antes que los métodos. Su implementación y ejecución deben consumir el menor tiempo posible.
- Experiencia: La creación de pruebas fue una experiencia nueva para el equipo de trabajo al ser una de las reglas de la metodología XP que no se había llegado a utilizar en proyectos anteriores. Debido a esto, la realización de pruebas al principio del proyecto fue traumática y demandó más tiempo de lo planeado, lo cual no fue conveniente ya que la metodología intenta disminuir los cuellos de botella, no aumentarlos. El carácter obligatorio de la escritura de las pruebas antes del desarrollo de los métodos del sistema implica un proceso de diseño previo. Esto se considera una ventaja ya que se destina tiempo en la construcción de la prueba, pero al realizar la codificación del método, éste resultaba de manera casi inmediata. También se destaca la autonomía que deben tener dichas pruebas a la hora de su ejecución, lo que implicaba la manipulación de la base de datos y la recuperación de su estado inicial al finalizar la prueba.

Según XP, se deben crear todas las pruebas de una clase antes de comenzar a desarrollar los métodos. En la experiencia fue conveniente realizar las pruebas individualmente, debido a que se producían errores al ejecutar todas las pruebas en un solo llamado. Se descubrió que este inconveniente estaba relacionado con la base de datos y no con los métodos, lo que al principio del proyecto aplicó dificultad al evaluar si un método había pasado o no la prueba. Una vez que se descubrió este error se empezó a ejecutar las pruebas por grupos de procesos en lugar de ejecutarlas todas de una vez. De esta forma no había problemas de comunicación con la base de datos y se garantizaba que si una prueba fallaba era solo por errores de lógica. XP propone la utilización de una herramienta que realice de forma automática la implementación y ejecución de las pruebas. En este caso se recurrió a realizar el debug que proporción netbeans para validar y encontrar los errores.

- Pruebas de aceptación: Lo que dice XP. Se deben diseñar las pruebas de aceptación con base en las historias de usuario
- Experiencia

Tres elementos permitieron al desarrollador diseñar las pruebas de aceptación. En primer lugar el tipo de sistema implementado era suficientemente sencillo y conocido, En segundo lugar las reuniones de las cuales se obtuvieron las historias de usuario se aplicaron para la reconstrucción de las pruebas de aceptación por parte del desarrollador sin toda la intervención del cliente. En tercer lugar el cliente aceptó el delegar esta función de diseño de las pruebas debido que su disponibilidad de tiempo se lo impidió.

- Cuando se encuentra un error: Lo que dice XP, al encontrar un error debe escribirse primero la prueba antes que corregirlo.
- Experiencia

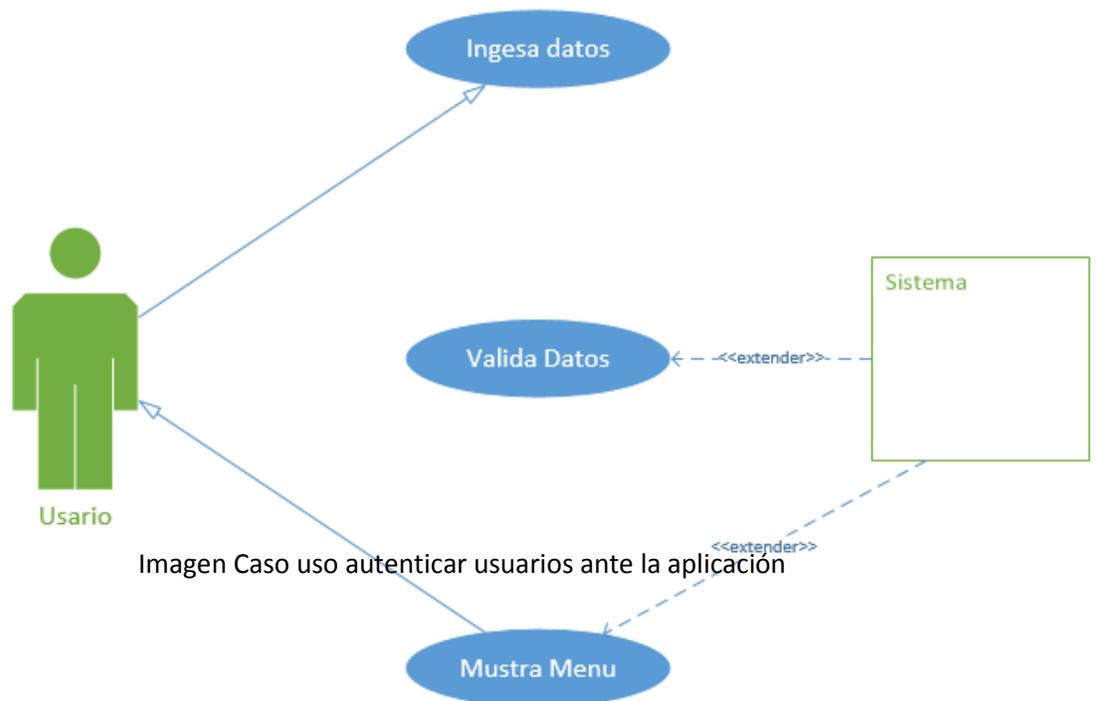
No se crearon unidades de prueba para los errores de las interfaces gráficas, estos tipos de inconvenientes fueron solucionados mediante pruebas manuales, es decir. Estas pruebas manuales consistían en cajas negras donde se verificaba la solución del problema mediante la ejecución del programa. Cuando el código de una de las clases principales pasaba la prueba que se había diseñado con anterioridad, pero el programador encontraban que la funcionalidad no era correcta, se modificaba la prueba y se corregía el código. La aplicación de esta idea en XP es relativa a dónde se halla encontrado el error. Evidentemente si el error fue encontrado a partir de la ejecución de una prueba unitaria no se hace necesario el escribir una nueva prueba ya que existe una para detectar el error (el caso más común en la realización del proyecto). Por otro lado, si el error se detectó por una inspección manual, efectivamente se debía elaborar una nueva prueba que detectara este error o revisar alguna que ya existiera, relacionada con este elemento de la aplicación, que detectara tal error.

7.2.2.3 Versiones e Iteraciones

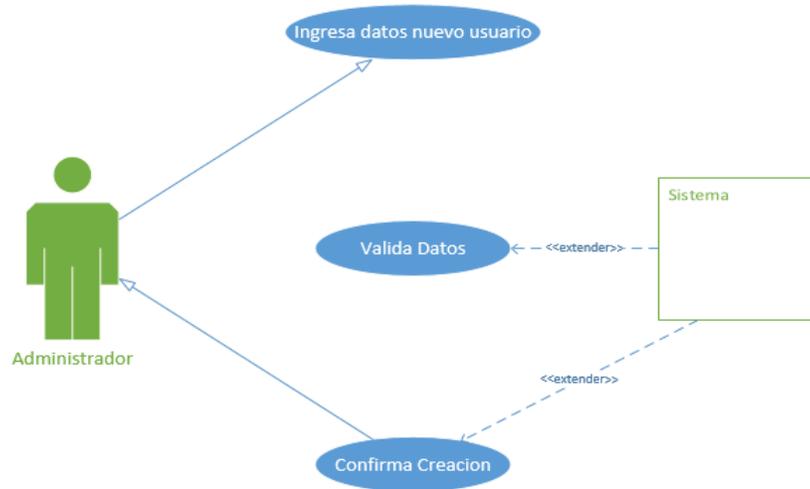
- **Versión 0.1**

Iteración 1: Para el modelado de la aplicación se diseñara dos diagramas para estructurar y entender el funcionamiento de la aplicación frente a los requerimientos solicitados por el cliente, los diagramas a realizar son Casos de Uso y Diagrama de Clases, los cuales se realizarán bajo la herramienta de Microsoft Visio Profesional 2013.

- **Casos de uso:** Los casos de uso documentan el comportamiento del sistema desde el punto de vista del usuario. A continuación se describen los casos de uso más relevantes del sistema, en estos se describe el requerimiento asociado, los actores implicados, las condiciones, las excepciones, los flujos principales. Para acceder a la información detallada de cada caso de uso identificado en el sistema véase el Anexo C.
- Autenticar usuarios ante la aplicación



- Creación de usuario



- Administrar kits y elementos

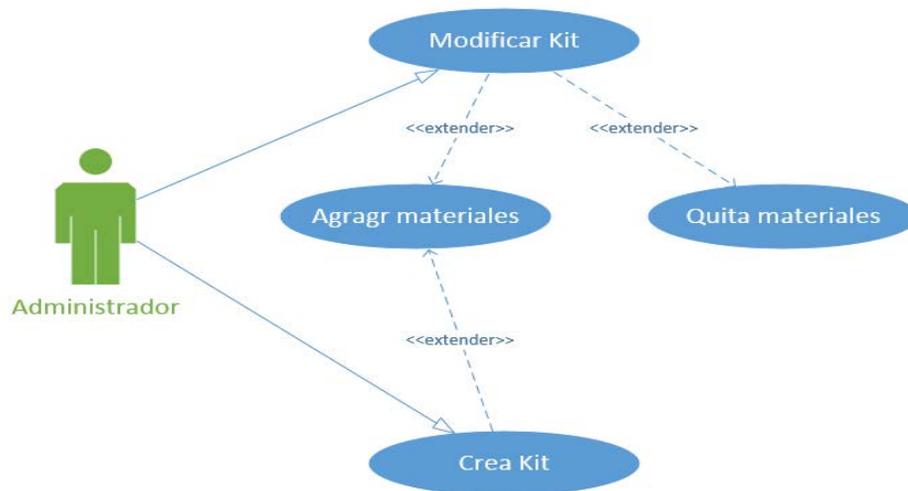


Imagen Caso uso administrar kits y elementos

- Administrar Kit

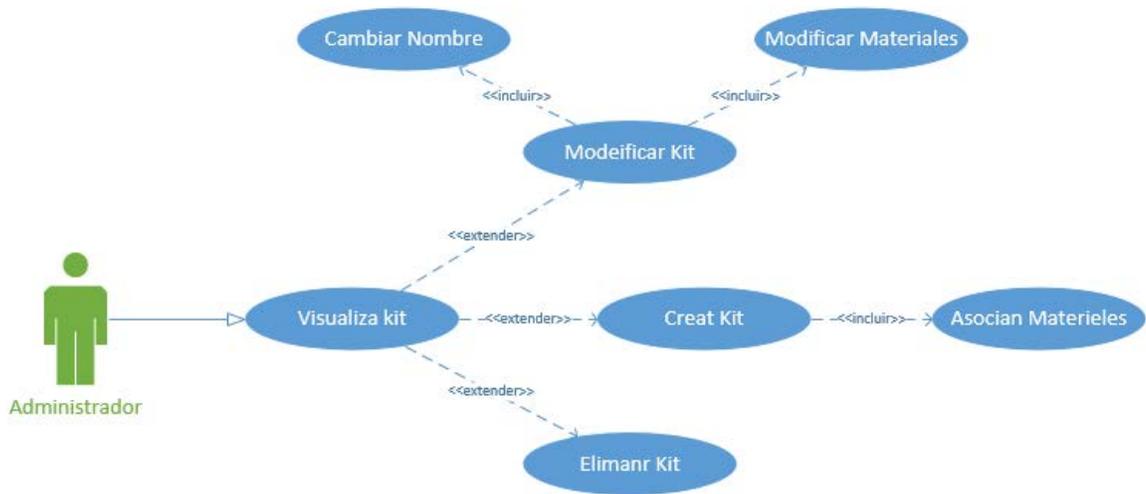


Imagen Caso uso administrar kit

- Creación de las remisiones y Datos Adicionales de la remisión

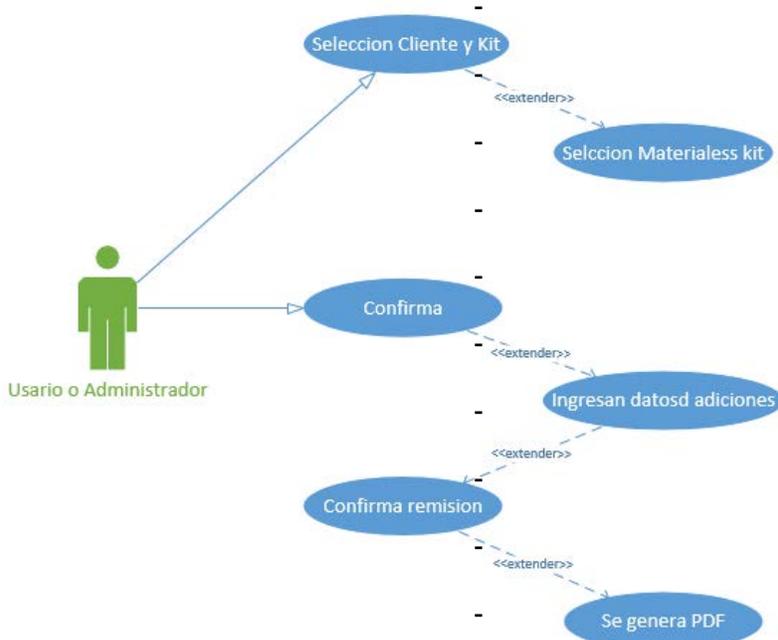


Imagen Caso uso creación de las remisiones y datos Adicionales de la remisión

- Consultar reporte de historial de remisiones.



Imagen Caso uso consultar reporte de historial de remisiones.

- Diagrama de Clases:** El diagrama de clase describe la estructura arquitectónica de la aplicación. En el diagrama de clases de la imagen Diagrama de Clases, se muestra la relación y la estructura de toda la aplicación a nivel de las clases, en esta grafica se puede observar la comunicación y dependencia de las clases. La imagen Diagrama de Clases, en el Anexo D se puede apreciar con mayor detalle y las tarjetas crc empleadas.

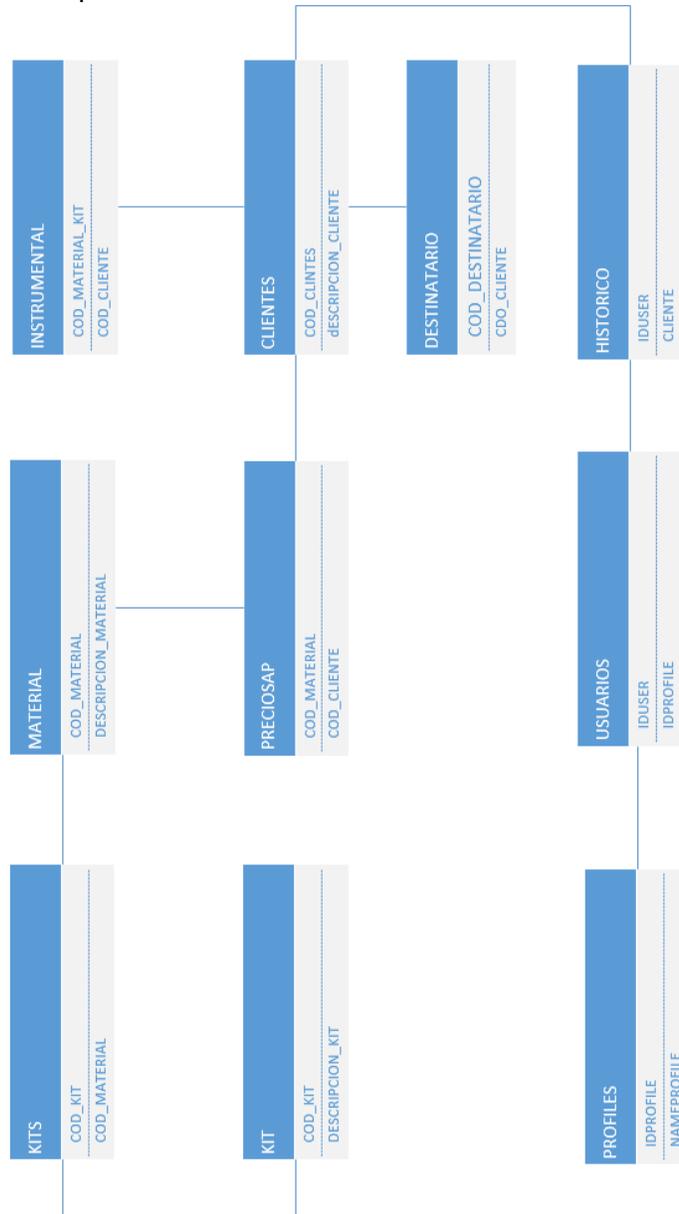


Imagen Diagrama de clases

- **Diagrama de Componentes:** El diagrama de componentes representa cómo el sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. La imagen Diagrama de componentes.

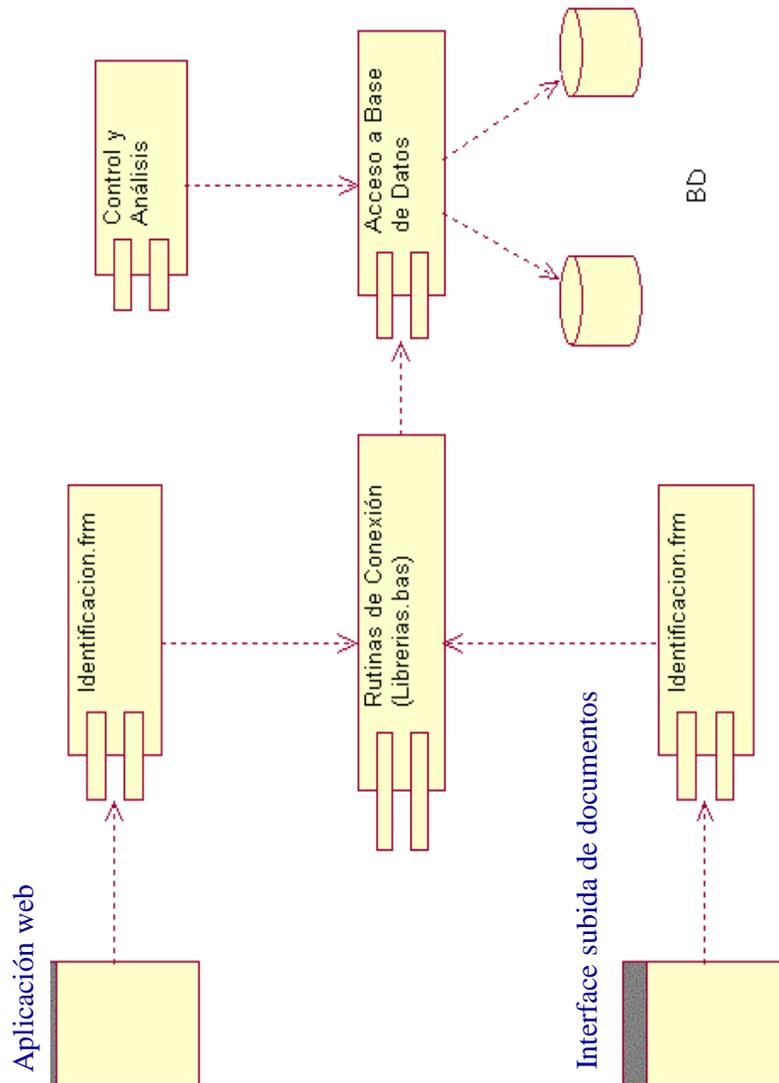


Imagen Diagrama de componentes

Iteración 3: El diseño de la Base de Datos es la parte primordial para empezar con el desarrollo de cualquier aplicación, ya que un buen diseño de ésta, es la base fundamental para el óptimo funcionamiento y éxito de cualquier software. Para definir el diseño de la base de datos se contó siempre con la ayuda de los clientes, quienes informaban qué datos serían relevantes para la remisión, en base a ello se estableció que datos básicos que se tendrían y como se alimentaría la información de la tablas de la base de datos, lo cual dio como resultado el modelo que se muestra a continuación, Ver (Imagen modelo ER). El diccionario de datos se puede ver en el anexo H.

Modelo entidad relación

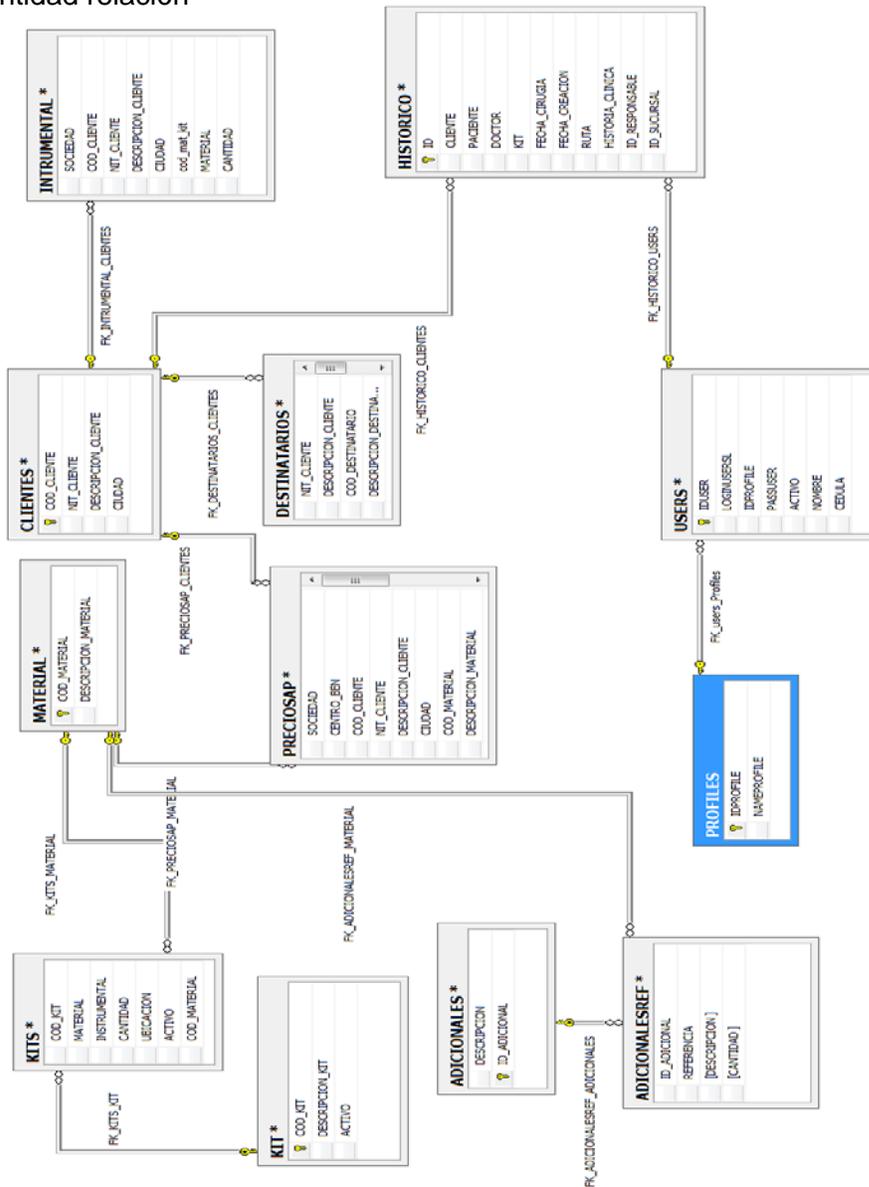


Imagen Modelo ER

A continuación se explican las tablas utilizadas.

- **Tabla Material:** Es la tabla es donde se guardan los materiales que son la referencia para los kits de las remisiones.

Los campos más importantes son:

- ID_MATERIAL: es la llave principal de la tabla.
- DESCRIPCION_MATERIAL: es este campo se almacena el nombre del material.

- **Tabla Kit:** es esta tabla se almacena en listado de kits que se utilizan en logística inversa para las remisiones.

Los campos más importantes son:

- COD_KIT: es la llave principal de la tabla.
- DESCRIPCION_KIT: es este campo se almacena el nombre del kit.

- **Tabla Kits:** en esta tabla se guarda la relación de cada material por cada kit

Los campos más importantes son:

- COD_KIT: es la llave foránea que hace relación hacia la tabla de kit.
- COD_MATERIAL: es la llave foránea que hace relación hacia la tabla de material.
- INSTRUMENTAL: en este campo se almacena la información que indica si el material del kit hace parte a un instrumental o no, dependiendo de esto tiene un valor referencia o no en la generación de la remisión.
- UBICACIÓN: en este campo se almacena la información que indica en qué posición de la caja el kit se encuentra el material
- ACTIVO: en este campo se almacena la información que indica si el material esta activo o no, ya que los materiales por condiciones de proveedores o por evolución de los procesos de cirugía pueden salir del inventario de la compañía.

- **Tabla Adicionales:** esta tabla se almacena en listado de materiales no comerciales que se utilizan en logística inversa para las remisiones

Los campos más importantes son:

- ID_ADICIONAL: es la llave principal de la tabla

- DESCRIPCION: en este campo se almacena el nombre del grupo de elementos adicionales.
- **Tabla AdicionalesRef:** esta tabla se almacena en listado de materiales que conforman el grupo de adicionales que se utilizan en logística inversa para las remisiones.

Los campos más importantes son:

- ID_ADICIONAL: es la llave foránea que hace relación hacia la tabla adicional.
- REFERENCIA: en este campo se almacena el código del material y es la llave foránea que hace relación hacia la tabla material.
- DESCRIPCION: en este campo se almacena el nombre del material.
- CANTIDAD: en este campo se almacena la cantidad mínima disponible del material.
- **Tabla PreciosSap:** en esta tabla se guarda la relación entre cliente, producto y precio para cada uno de los materiales, dado que para cada cliente existe un precio por producto, esta tabla es alimentada por una interface que carga un archivo plano proveniente de SAP. En esta tabla se actualiza la información o se inserta en caso que no exista, a su vez por medio de unos triggers actualiza la tabla de clientes y material, y relaciona al cliente con todos los productos.

Los campos más importantes son:

- SOCIEDAD: en este campo se almacena la referencia a la sociedad que corresponde a la empresa por la cual se factura al cliente.
- COD_CLIENTE: en este campo se almacena el código del cliente.
- DESCRIPCION_CLIENTE: es este campo se almacena la descripción del cliente.
- COD_MATERIAL: en este campo se almacena el código del material.
- DESCRIPCION_MATERIAL: en este campo se almacena la descripción del material.
- PRECIO: en este campo se almacena el precio del material por cada cliente.
- VALIDO_D y VALIDO_A: en este campo se almacena la fecha de vigencia del precio del material por cliente.

- **Tabla Clientes:** Es la tabla es donde se guarda el listado de clientes.

Los campos más importantes son:

- ID_COD_CLIENTE: es la llave principal de la tabla.
- DESCRIPCION_CLIENTE: es este campo se almacena el nombre del cliente.
- CIUDAD: en este campo se almacena el lugar de origen del cliente

- **Tabla Destinatarios:** Es la tabla es donde se guarda el listado de sucursales o lugares de despacho de las remisiones relacionados al cliente principal.

Los campos más importantes son:

- COD_DESTINATARIO: es la llave principal de la tabla.
- DESCRIPCION_DESTINATARIO: en este campo se almacena el nombre del destinatario.
- CIUDAD: en este campo se almacena el lugar de origen del destinatario
- DIRECCION: en este campo se almacena la dirección de despacho del destinatario.
- COD_CLIENTE: es la llave foránea que hace relación hacia la tabla cliente.

- **Tabla Instrumental:** esta tabla se almacena en listado de instrumental complementario que conforman los kits que se utilizan en logística inversa para las remisiones por cliente.

Los campos más importantes son:

- COD_CLIENTE: es la llave foránea que hace relación hacia la tabla clientes.
- COD_MAT_KIT: en este campo se almacena el código del instrumental.
- MATERIAL: en este campo se almacena el nombre del instrumental.
- CANTIDAD: en este campo se almacena la cantidad mínima disponible del instrumental.

- **Tabla Histórica:** en esta tabla se guarda la información de la remisión generada.

Los campos más importantes son:

- CLIENTE: en este campo se guarda el código del cliente la cual se generó la remisión.
- PACIENTE: en este campo se guarda el nombre del paciente para el cual se generó la remisión
- DOCTOR: en este campo se guarda el nombre del médico que va a realizar la operación.
- KIT: en este campo se guarda el código del kit que se despachó en la remisión.
- FECHA_CIRUGIA: en este campo se guarda la fecha de ejecución de la cirugía.
- FECHA_CREACION: en este campo se guarda la fecha de creación de la remisión.
- RUTA: en este campo se guarda la ruta del archivo pdf generado para la remisión.
- HISTORIA_CLINICA: en este campo se guarda el número de historia clínica a la cual se vincula la remisión.
- ID_RESPONSABLE: en este campo se guarda el id del usuario que genero la remisión y es llave foránea a la tabla de usuarios.
- ID_SUCURSAL: en este campo se guarda el id de la sucursal a la cual se despacha la remisión.
- **Tabla Users:** en esta tabla se guarda la información de los usuarios que tienen acceso a la aplicación.

Los campos más importantes son:

- IDUSER: en este campo se guarda el id del usuario y es la llave principal de la tabla.
- LOGINUSERSL: en este campo se guarda el nombre del usuario.
- ACTIVO: en este campo se guarda el estado del usuario
- IDPROFILE:
- **Tabla Profiles:** en esta tabla se guarda el perfil o prioridad del usuario.

Los campos más importantes son:

- IDPROFILE: este campo es la llave principal de la tabla y hace referencia a la prioridad del usuario.

- NAMEPROFILE: es este campo se almacena el nombre del perfil

- **Versión 0.2**

Iteración 1:

- **Prototipo:** Como la aplicación debía estar en la red para poder ser consultada desde cualquier parte, se seleccionó WAMP SERVER, NETBEANS Y PHP, ya que es uno de los software de desarrollo para aplicaciones web más utilizados y de gran difusión, por estas razones se Selección de Herramientas, se consideró una herramienta ideal para el desarrollo de esta aplicación.

Iteración 2: Como se buscaba tener un diseño amigable, intuitivo y de fácil uso para la aplicación se creó una página principal (Master Page), la cual es una plantilla de la que heredarán todas las demás páginas de la aplicación web.

En la página principal (“Plantilla. Master”) se definió la estructura común para toda la aplicación web, su diseño incluye un banner donde se encuentra el escudo de la empresa y el menú de navegación de la aplicación, el menú de la página es de fácil acceso, también se buscó que la combinación de los colores de la página fuera agradable a la vista del usuario (Imagen 8).



[Agregar Nuevo Kit](#) [Modificar Kits](#) [Cargar Archivo](#) [Nuevo Usuario](#) [Generar Remisión](#) [Historial Remisión](#) [Cambio Contraseña](#)

Imagen 8: página maestra

Iteración 3: Se implementa el módulo de consultas. Para ello se realizaron funciones que consultan a la base de datos.

Se descargan los controles que proporciona la conexión entre Wampserver y SQL Server, pudieron crear los enlaces al motor de base de datos y a sus diferentes tablas, para así hacer las consultas respectivas, ver anexo F.

Por ejemplo, para el caso de consulta por Kit se utilizó un control jqGrid enlazado a las tablas Kit, el cual muestra todos los kits existentes, y haciendo el filtrado respectivo en la sentencia SQL, se pueden visualizar los datos esperados.

Las demás consultas se hicieron de forma similar, por ejemplo para la consulta por Cliente simplemente se cambió la consulta de la tabla kit por la tabla Cliente y haciendo su respectivo filtrado se visualizan los datos deseados.

- **Versión 0.3**

Iteración 1: En esta versión se implementó los módulos para la inserción de datos.

En estos módulos se hizo la aplicación que permitiera ingresar nuevos datos a la base de datos, por medio de la carga de un archivo plano por medio de las funciones JavaScript que proporciona CodeIgniter y de los controles que proporciona php.

De esta manera se termina la versión 0.3, cumpliendo con las tareas de la historia de usuario número 4.

Esta versión se le enseñó a los clientes, los cuales dieron su visto bueno al sentirse conformes con los avances presentados.

- **Versión 0.4**

Iteración 1: En esta versión se implementó el módulo para la edición de datos. Este módulo fue muy similar al módulo de ingresar nuevos datos, porque por medio de los controles que proporciona la herramienta se podían editar los datos que existían en la base de datos.

De esta manera se terminó la versión 0.4, cumpliendo con las tareas de la historia de usuario número 5.

Esta versión se le enseñó a los clientes, los cuales dieron su visto bueno al sentirse conformes con los avances presentados.

- **Versión 0.5 hasta la 0.9**

De la versión 0.5 hasta la versión 0.9 se trabajó el diseño estético de la aplicación. En cada una de las iteraciones que involucraba la entrada de datos por parte del usuario de la aplicación, se diseñaron e implementaron los diferentes tipos de validaciones respectivas para cada uno de los campos que lo requerían.

En este punto fue primordial validar los campos requeridos y el formato de los archivos a cargar, los cuales no debían quedar vacíos o extensiones incorrectas ya que representarían un error al momento de ser cargados en la base de datos.



9. PRESUPUESTO

Tal como se observa en la Tabla 13, el cálculo del presupuesto de esta aplicación se basa en el análisis de las horas de desarrollo consumidas en los 70.8 días planteados en el cronograma, para este caso tenemos que el costo promedio de una hora de desarrollo es 40.000 pesos colombianos, a razón de 16 horas trabajadas los fines de semana y 20 horas promedio entre semana, esto nos da un total de 324 horas, para un valor aproximado del desarrollo de \$ 12.960.000 pesos colombianos, es de aclarar que en el valor de hora de desarrollo se incluyen los gastos adjuntos que conllevan las reuniones de la recolección de información, investigación, pruebas, instalación y despliegue, entre otros.

Por otro lado, No se incluye en el presupuesto el valor de las licencias Wampserver y Netbeans ya que son de uso libre. En caso de ser publicada la aplicación en un Hosting privado, su valor oscila \$256.000 pesos colombianos, con las siguientes características generales: 1 Gb de espacio, 50 Gb de tráfico mensual.

También se debe incluir el diseño gráfico de la página web, el cual está compuesto de dos propuestas de diseño y tiene un valor aproximado de \$1.000.000 de pesos colombianos.

Para resumir, el presupuesto del desarrollo de la aplicación tiene un valor aproximado de \$14.256.000 pesos colombianos.

Tabla 13. Presupuesto para el desarrollo de la aplicación.

Días	Horas de trabajo por semana	Total horas	Valor hora desarrollo	Valor total horas desarrollo	Diseño gráfico	Hosting privado	Valor total del desarrollo
70,8 d	36 h	324 h	\$ 40.000 COL\$	\$ 12.960.000 COL\$	\$ 1.000.000 COL\$	\$ 256.000 COL\$	\$ 14.256.000 COL\$

10. DISEÑO

Para el diseño de la aplicación se siguió las recomendaciones de la metodología ágil XP, siempre se trató de evitar las soluciones complejas, y se trabajó en una sola iteración, sin pensar en las que vendrían más adelante.

Otro aspecto importante en el Diseño, es la constante reestructuración del código. El principal objetivo de la Reestructuración es evitar la duplicación de código, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. Esto se hizo constantemente en la programación de la aplicación por ejemplo, en el módulo de Modificación de kits se hizo Editar un elemento del kit, y para hacer los demás editar solo se cambiaba la instrucción.

10.1 Codificación

10.1.1 Cliente siempre disponible

XP recomienda como factor de éxito que el cliente esté involucrado en toda la etapa del desarrollo, esto se cumplió satisfactoriamente ya que como se explicó anteriormente el desarrollador eran al mismo tiempo clientes.

10.1.2 Estándares de codificación

La metodología XP aconseja seguir estándares de codificación para que cualquier integrante del equipo de desarrollo pueda entender y asimilar fácilmente código escrito por otro integrante. Seguir esta recomendación fue fácil para el desarrollador ya que venían trabajando de tiempo atrás en otros desarrollos y esta buena práctica siempre se tuvo presente. (Ver Anexo A. Estándares).

10.1.3 Pruebas

XP recomienda diseñar las pruebas antes de codificar los módulos, esta práctica represento un cambio drástico en la forma de desarrollar del equipo de trabajo ya que como no se había utilizado en proyectos anteriores se tuvo algunos inconvenientes y por esta razón se llevó más tiempo de lo deseado. (Ver Anexo B. Pruebas).

10.1.4 Instalación

Para implementar esta fase se debe analizar los resultados obtenidos de la etapa de pruebas, luego se debe corregir los errores, realizar mejoras o cambios sobre la aplicación acorde a dicho análisis.

En esta etapa el software se instala en el ambiente de producción, también se implementan de ser necesario el hardware y software adicional que se necesite para el funcionamiento de nueva aplicación. A lo largo de esta etapa se documenta los posibles errores que ocurran y las mejoras o cambios solicitados por el cliente para la aplicación, tanto en diseño, como en la lógica del negocio. Al finalizar esta etapa se debe crear un documento donde se especifique el estado del software y la aprobación del cliente frente a los requisitos solicitados. Ver anexo E.

10.1.5 Mantenimiento

En muchos casos el software requiere ajuste o correcciones de errores, también en ocasiones las funciones que presta el software, no cumple con las expectativas de cliente y se debe acudir a mejoras o cambio en la aplicación, todo esto puede ser producido por la implementación de nuevo hardware, cambio en las interfaces externas a las que se hace referencia dentro del proyecto o por cambios, o nuevas necesidades del cliente. A esta etapa se le denomina "Mantenimiento" y muy ligada a la etapa de codificación, ya que si se tiene un código fuente de alta calidad se obtendrá una buena escalabilidad del software.

Para el desarrollo del proyecto esta fase no se tendrá en cuenta, pero es de aclarar que para efectos de algún mantenimiento se entregara el código fuente de la aplicación y el script de la base de datos.

11. INTERFACES

11.1 Interfaces parte pública

La página de inicio de la aplicación es la página de login, tiene el logo de la empresa, además de contar con un mensaje de bienvenida para los usuarios y un formulario para ingresar el Nombre de Usuario del y la Contraseña. (Ver Imagen 9)



Imagen 9. Iniciar sesión

11.1.1 Inicio

Tiene un menú superior desde donde se da acceso a las diferentes páginas de la aplicación web (Ver Imagen 10). Este menú consta de:



11.1.2 Agregar Nuevo Kit

En la parte de nuevo kit el usuario administrador puede visualizar el nombre de los kit que existe en una grilla que aparecerá abajo del formulario de ingreso de nuevo kit, en el formulario de creación de kit, el administrador ingresa el nombre del kit y a continuación clic en el botón enviar para registrar en el sistema la descripción del nuevo kit, ver (imagen 11).



Administrador [Cerrar Sesio](#)

[Agregar Nuevo Kit](#)

[Modificar Kits](#)

[Cargar Archivo](#)

[Generar Remisión](#)

[Historial Remisión](#)

Nuevo Kit

INTRODUCE DESCRIPCION_KIT

enviar

Referencia	Descripcion	Accion
2	AGUJAS COLORADO	
7	APEX_HIDROXIAPATITA	
61	ARTROSCOPIA_HOMBRO	

Imagen 11. Agregar Nuevo kit.

11.1.3 Modificar Kits

En esta página se visualizan los materiales que conforma un kit, se escoge del control de selección un kit y a continuación la grilla que está en parte inferior se carga con los elementos, está a su vez tiene una columna con las opciones de editar y eliminar los materiales para el kit, ver (Imagen 12).



Administrador [Cerrar Sesion](#)

[Agregar Nuevo Kit](#)
 [Modificar Kits](#)
 [Cargar Archivo](#)
 [Generar Remisión](#)
 [Historial Remisión](#)

Lista de Materiales

Kits

Cod Kit	Referencia	Descripcion	Accion
7	501431805	APEX AUTOPER +y6 x 180 mm LR 30 S	 
7	501452005	APEX AUTOPER +y6 x 200 mm LR 50 S	 
7	501741805	APEX AUTOPER +y5 x 180 mm LR 40 S	 
7	501442005	APEX AUTOPER +y6 x 200 mm LR 40 S	 
7	501751805	APEX AUTOPER +y5 x 180 mm LR 50 S	 

Imagen 12. Modificar kit.

- **Editar:** en la opción editar de la grilla, al dar clic sobre el icono  , se despliega una ventane emergente, en esta solo permite cambiar el nombre al material que hace

relación al kit, luego del cambio clic en el botón guardar para registrar el cambio ver (imagen 12.1).

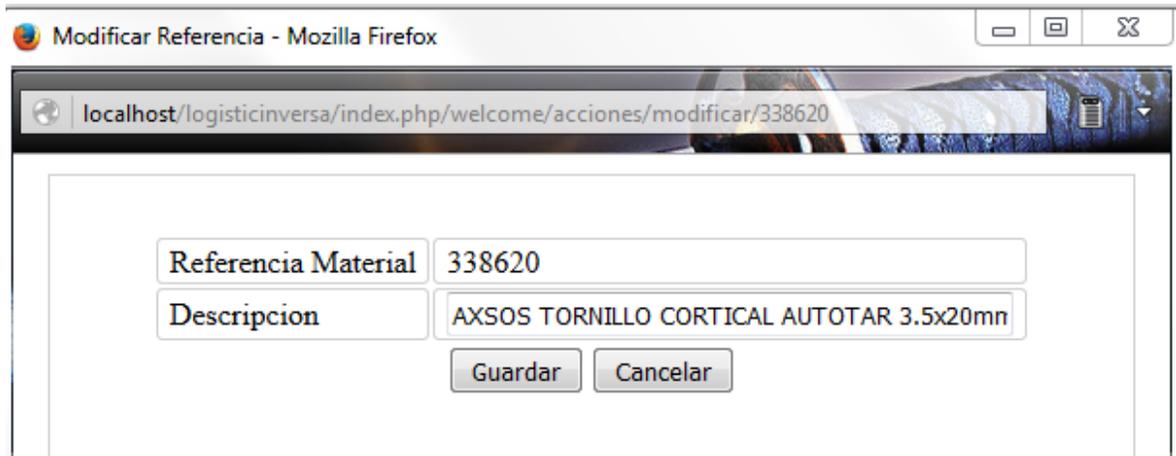


Imagen 12.1. Editar

- **Eliminar:** en la opción eliminar de la grilla al dar clic sobre el icono  se elimina el material que hace relación al kit, se despliega una ventana emergente, clic en el botón Aceptar para registrar el cambio ver (imagen 12.2).

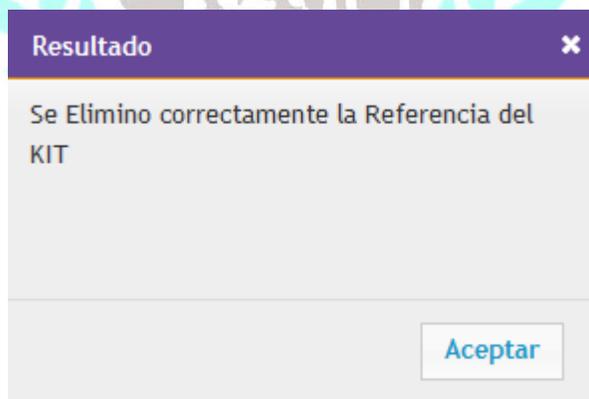


Imagen 12.2. Eliminar

11.1.4 Cargar Archivo

En esta página se visualizan los un formulario para la cargar de los materiales que harán parte del nuevo kit creado o para actualizar un kit existente, este contiene el botón examinar para buscar el archivo .CSV, el botón enviar que carga la información en una tabla temporal y el botón actualizar que activa un MERGE ver anexo G; que actualiza la información en la tabla kits y en la tabla material si existe, en caso contrario inserta los datos en la base de datos ver (Imagen 13).



Administrador [Cerrar Sesión](#)

[Agregar Nuevo Kit](#) [Modificar Kits](#) [Cargar Archivo](#) [Generar Remisión](#) [Historial Remisión](#)

Cargar Archivo

No se ha seleccionado ningún archivo.

Para descargar la plantilla presione [AQUI](#)

Imagen 13. Carga archivo

11.1.5 Generar Remisión

En esta página se visualizan los un formulario para la creación de la remisión este consta de del campo Nombre Pagador que es de selección, así como el campo Kits y el Campo

Motor que es opcional y el botón , al dar clic sobre el último, el sistema llena la grilla con la información del kit, teniendo como prioridad el precio unitario, si tiene IVA y la cantidad minina del material que está configurado para el kit, se seleccionan los materiales que se van a enviar en la remisión y a continuación clic en el botón

Diligenciar, ver (Imagen 14), se despliega una ventana emergente para diligenciar los datos adicionales de la remisión, ver (Imagen 14.1)

30 Años
GRUPO AMAREY
NOVA MEDICAL

Administrador [Cerrar Sesión](#)

Agregar Nuevo Kit Modificar Kits Cargar Archivo **Generar Remisión** Historial Remisión

Remisión

Nombre Pagador: ADMINISTRADORA COUNTRY S A Kits: AGUJAS COLORADO
Motor: -

Consultar

Referencia	Descripcion	Cantidad	Iva	Precio Unitario
<input checked="" type="checkbox"/> N103A	MAC AGUJA COLORADO 30MM DE LONG. AISLADA	1	16%	171.186,00
<input checked="" type="checkbox"/> N104A	MAC AGUJA COLORADO 40MM DE LONG. AISLADA	4	NA	171.186,00

Mostrando 1 - 2 c

Diligenciar

Chalecos - Bomba Irrigación Grande - Bomba Irrigación Pequeña - Consola Shaver - Compas RTC

Imagen 14. Generar Remisión

- En la ventana emerge que se genera se ingresan los datos adicionales como son; Institución a entregar (destinatario), paciente (Nombre del paciente), Doctor (Nombre del Doctor), Historia clínica y Fecha de la cirugía, y una lista de elemento extras que son opciones

Y nombre del pagador, kits y la lista de materiales que se seleccionaron en el formulario inicial, por último el botón **Generar PDF** que guarda la información en la tabla historico y genera un pdf con la remisión para el cliente, ver (Imagen 14.1, 14.2 y 14.3).

Lista de Materiales

Pagador:	ADMINISTRADORA COUNTRY S A	Kits:	AGUJAS COLORADO
Institucion a Entregar:	-		
Paciente:		Historia Clínica:	
Doctor:		Fecha de Cirugia:	
<input type="checkbox"/> Chalecos - <input type="checkbox"/> Bomba Irrigación Grande - <input type="checkbox"/> Bomba Irrigación Pequeña - <input type="checkbox"/> Consola Shaver - <input type="checkbox"/> Compas RTC			

Referencia	Descripcion	Cantidad	Iva	Valor	Sub Total
N104A	MAC AGUJA COLORADO 40MM DE LONG. AISLADA	4	NA	171.186,00	684.744
N103A	MAC AGUJA COLORADO 30MM DE LONG. AISLADA	1	0.16	171.186,00	198.57576

Oservaciones:

Generar PDF

Imagen 14.1 datos adicionales.

Validación de campos diligenciados, ver imagen (Imagen 14.2)

Lista de Materiales

Pagador:	ADMINISTRADORA COUNTRY S A	Kits:	AGUJAS COLORADO
Institucion a Entregar:	-		
Paciente:		Historia Clínica:	
Doctor:		Fecha de Cirugia:	
<input type="checkbox"/> Chalecos - <input type="checkbox"/> Bomba Irrigación Grande - <input type="checkbox"/> Bomba Irrigación Pequeña - <input type="checkbox"/> Consola Shaver - <input type="checkbox"/> Compas RTC			

Referencia	Descripcion	Cantidad	Iva	Valor	Sub Total
N104A	MAC AGUJA COLORADO 40MM DE LONG. AISLADA	4	NA	171.186,00	684.744
N103A	MAC AGUJA COLORADO 30MM DE LONG. AISLADA	1	0.16	171.186,00	198.57576

Oservaciones:

Campos Requeridos

Los siguientes campos son requeridos:

- => Paciente
- => Historia Clínica
- => Doctor
- => Fecha Cirugia
- => Institucion a Entregar

OK

Generar PDF

Imagen 14.2 Validación de datos adicionales.

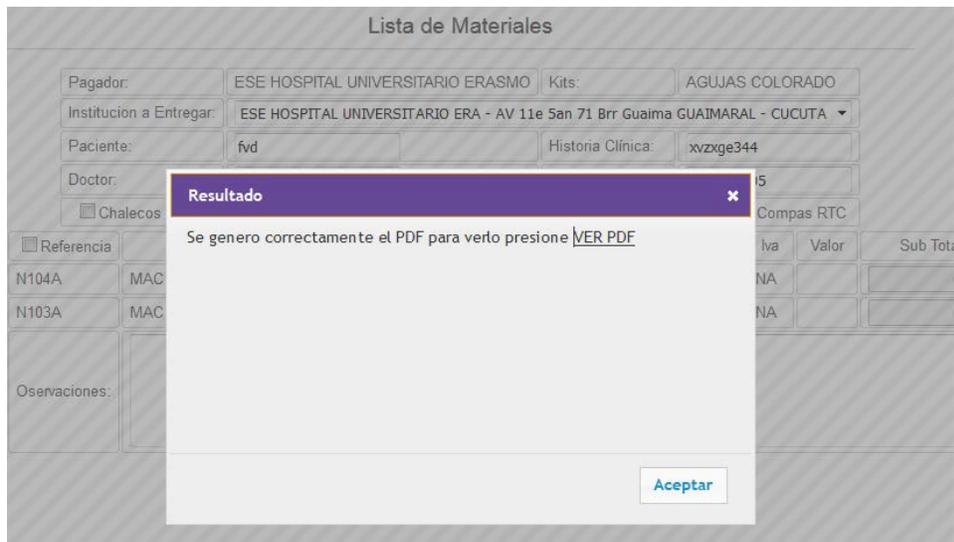


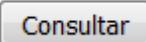
Imagen 14.3 Generación pdf.

- PDF generado de la remisión que se entrega al cliente, ver (Imagen 14.4)



Imagen 14.4 pdf

11.1.6 Historial Remisión

En esta página se visualizan los un formulario para la búsqueda de la remisiones generadas en el sistema, los campos de filtro son: N° remisión, paciente, historia clínica, fecha de creación y fecha de cirugía, y los botones:  que envía el parámetro a la consulta a la base de datos para devolver la información encontrada y poblar la grilla, y el botón  que limpia el contenido de la grilla, ver (Imagen 15).

Agregar Nuevo Kit Modificar Kits Cargar Archivo Generar Remisión **Historial Remisión**

Historial Remisión

N° Remisión:	Paciente:	Historia Clínica:	Doctor:	Cliente:
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Fecha de Creacion:			Fecha de Cirugia:	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Consultar"/>		<input type="button" value="Limpiar"/>		

Id	Fecha Creacion	Cliente	Paciente	Historia Clínica	Doctor	Fecha Cirugia	Ver PDF



Imagen 15. Historial remisión

- Resultado de la consulta, la grilla se carga con la información solicitada, id de la remisión(automático), fecha de creación, paciente, historia clínica, doctor, fecha de cirugía y ver pdf que es la columna que permite ver y descargar el pdf de la remisión, ver (Imagen 15.1).

Historial Remisión

N° Remisión:	Paciente:	Historia Clínica:	Doctor:	Cliente:
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Fecha de Creacion:		Fecha de Cirugia:		
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Consultar"/>		<input type="button" value="Limpiar"/>		

Id	Fecha Creacion	Cliente	Paciente	Historia Clínica	Doctor	Fecha Cirugia	Ver PDF
1	2015-07-03	ADMINISTRADORA COUNTRY S A	ARTURO MEJIA	55454544	ESTEBAN LIZARAZO	2015-07-04	
2	2015-07-03	ADMINISTRADORA COUNTRY S A	GERARDO BEDOYA	454858586	ESPERANZA	2015-07-10	
3	2015-07-03	ADMINISTRADORA COUNTRY S A	GERMAN MERCHAN	123456	ALEXIS VILLAMIZAR	2015-07-04	
4	2015-07-06	HOSPITAL LA VICTORIA III NIVEL ESE	LUIS MIGUEL MORENO	1031150607	JOSE FABIO VELA	2015-07-07	
5	2015-07-06	HOSPITAL DE SUBA II NIVEL E S E	RUTHER GASCA	16845439	DAVID ALEJANDRO BORDA	2015-07-07	
6	2015-07-06	MEDICOS Y SERVICIOS INTEGRALES ESPE	RUTHER GASCA	16845439	DAVID ALEJANDRO BORDA	2015-07-07	
7	2015-07-06	HOSPITAL SANTA CLARA	BENITO CAMELAS	1234	TAZO	2015-07-07	
8	2015-07-08	CORPORACION HOSPITALARIA JUAN CIUD	YOLANDA AVILA	39529500	JORGE BARBOSA	2015-07-09	
9	2015-07-08	HOSPITAL SANTA CLARA	JOSEPH MIÑÓZ	100061675	MANUEL IGNACIO GUTIERREZ	2015-07-10	

Mostrando 1 - 500 de 585

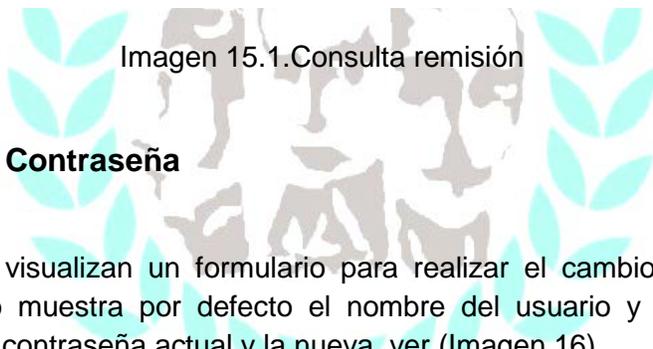


Imagen 15.1. Consulta remisión

11.1.7 Cambio de Contraseña

En esta página se visualizan un formulario para realizar el cambio de contraseña, al cargar el formulario muestra por defecto el nombre del usuario y su identificador, el formulario solicita la contraseña actual y la nueva, ver (Imagen 16).

Cambio Contraseña

Usuario:	admin
ID:	1
Password Actual:	<input type="password"/>
Password:	<input type="password"/>
<input type="button" value="Guardar"/>	

Imagen 16. Cambio contraseña

- Validación de la contraseña, el formulario valida que la contraseña ingresada en password actual sea el que se encuentra en el sistema, no permite que la contraseña nueva sea 0 o vacío y que no se igual a la existente, ver (Imagen 16.1)



Usuario:	admin
ID:	1
Password Actual:	<input type="text"/>
Password:	<input type="text"/>

Imagen 16.1. Validación contraseña

- Si se cumple con los requisitos se realiza el cambio de la contraseña, ver (Imagen 16.2)

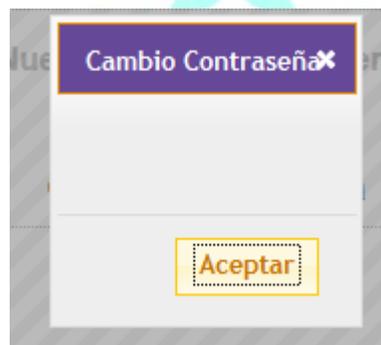


Imagen 16.2. Confirmación cambio de contraseña

11.1.8 Nuevo Usuario

En esta página se visualizan un formulario para la creación de un nuevo usuario que estará vinculado al cliente principal de la cuenta que se seleccione, adicional tiene un agrilla con la información de los usuarios existentes la cual permite su edición, ver (Imagen 17).

Crear Usuario

Formulario de registro

Usuario:	<input type="text"/>
Nombre:	<input type="text"/>
Cedula:	<input type="text"/>
Correo:	<input type="text"/>
Password:	<input type="text"/>
Estado:	Categoría:
<input type="text"/>	<input type="text"/>

Modificar Usuario

CODIGO	USUARIO	ESTADO	NOMBRE	CEDULA	CORREO
1	admin	A	Administrador	123456789	45465@hotmail.com
3	cristian.porras	A	Cristian Leonardo Porras Aldana	80162866	

Imagen 17. Nuevo usuario

- el formulario de creación de usuario contiene los siguientes campos: usuario, nombre, cedula, correo, password, estado (activo, inactivo) y categoría (general, administrador). Al crearse el usuario se muestra un mensaje de confirmación, ver (Imagen 17.1)

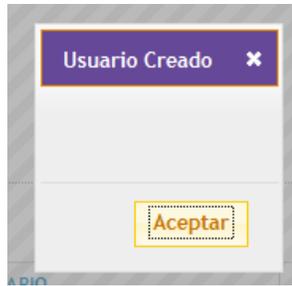


Imagen 17.1. Usuario creado

- En la parte inferior del formulario de la página de nuevo usuario, se encuentra una grilla con la información de los usuarios existentes en el sistema, al dar clic sobre cualquiera de ellos se despliega un formulario para su edición, ver (Imagen 17.1 y 17.2)

Modificar Usuario

CODIGO	USUARIO	ESTADO	NOMBRE	CEDULA
1	admin	A	Administrador	123456789
3	cristian.porras	A	Cristian Leonardo Porras Aldana	80162866
9	danny.echeverry	A	Danny Echeverry Poveda	79967469
4	diego.nunez	A	Diego Andres Nuñez Hernandez	80241694

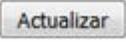
Imagen 17.2. Lista usuarios

- Formulario edición usuario, permite cambiar estado, categoría, correo y contraseña.

Usuario:	<input type="text" value="diego.nunez"/>
ID:	<input type="text" value="4"/>
Estado:	Categoría: estandar
<input type="text"/>	<input type="text"/>
Correo:	<input type="text"/>
Password:	<input type="text"/>

Actualizar

Imagen 17.3. Formulario edición usuario

Una vez se modifique el usuario y se da clic en el botón  , se muestra el siguiente mensaje, ver (Imagen 17.4)

El Usuario fué actualizado correctamente!



Imagen 17.4. Confirmación edición usuario

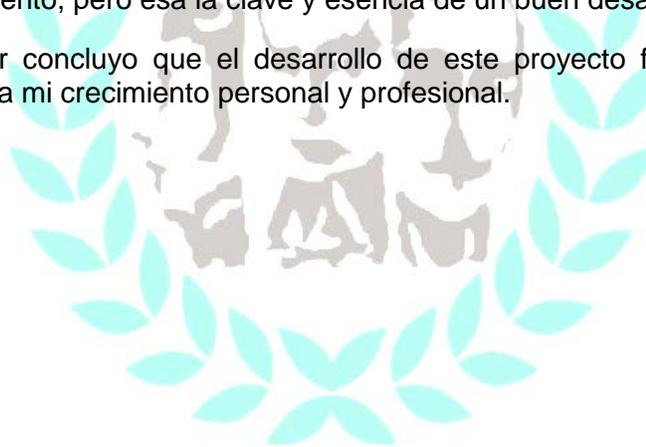


12. CONCLUSIONES

- La principal conclusión del proyecto es el beneficio y el gran impacto que genera la aplicación en la empresa Amarey nova medical, Es evidente que en la empresa Amarey nova medical no existe alguna aplicación que lleve el control sobre las remisiones generadas a los clientes. Mediante el uso de la aplicación de remisiones se puede obtener los tiempos de ejecución, control sobre lo enviado, en si toda la traza de la remisión, por tal se deduce que el proyecto realizado es de vital importancia para la empresa Amarey nova medical. También es de alto impacto ya que se tienen una herramienta que acerca al área de logística inversa y a las demás áreas de la empresa aún más a la utilización de las TIC's.

La necesidad y la ambición de resolver problemas es el combustible para todo desarrollador de software, pero vale aclarar que su talón de Aquiles es el perfeccionismo improductivo, tales casos se vivieron en todo el ciclo de desarrollo del proyecto, a veces se excedían los tiempos de desarrollo por tratar de brindar una aplicación amigable, con interfaces gráficas agradables, fáciles de manejar y buen rendimiento, pero esa la clave y esencia de un buen desarrollo.

Para finalizar concluyo que el desarrollo de este proyecto fue una experiencia fructífera para mi crecimiento personal y profesional.



13. REFERENCIAS BIBLIOGRAFICAS

- CA-DE-LI-G-001-V3 CARACTERIZACIÓN LOGISTICA INVERSA.pdf
- F. KORTH Henry y SILBERSHATZ Abraham, Fundamentos de Base de Datos, España. Segunda Edición. McGraw Hill 1993
- JOHANSEN. Introducción a la teoría general de sistemas. 1997.
- KENDALL & KENDALL, Análisis Estructurado Moderno, Primera Edición 1993. Editorial McGraw Hill.
- <http://www.monografias.com/trabajos/gis/gis.shtml>
- <http://es.slideshare.net/scourge/logistica-inversa-17191095>
- http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html#.VeWglpdSLwA
- Introducción Métrica: <https://www.google.com/search?q=M%C3%A9trica+3+&ie=utf-8&oe=utf-8#q=metodolog%C3%ADa+m%C3%A9trica+v+3>MACCONELL Roger S, Ingeniería de Software un Enfoque Práctico, Madrid; Santa fe de Bogotá; McGraw Hill, 1997
- OSCAR JOHANSEN BERTOGLIO, Introducción a la Teoría General de Sistemas. 1997, Editorial Limusa.
- SENN James. Análisis y diseño de sistemas de información, Colombia. 1998.
- Rafael Lapiedra Alcamí, Carlos Devece Carañana, Joaquín Guiral Herrando, Introducción a la gestión de sistemas de información en la empresa, Publicacions de la Universitat Jaume I. Servei de Comunicació i Publicacions Campus del Riu Sec. Edifici Rectorat i Serveis Centrals. 12071 Castelló de la Plana, 2010.
- <https://technet.microsoft.com/es-es/library/ms187875%28v=sql.105%29.aspx>
- <http://www.adwe.es/codigo/codeigniter-framework-php-desarrollo-aplicaciones-web>
- <http://www.cyta.com.ar/ta0502/v5n2a1.htm>
- Sociedad Latinoamérica para la calidad – Análisis Costo/Beneficio, Copyright 2000.
- El lenguaje unificado de modelado - Grady Booch, James Rumbaugh, Ivar Jacobson- El libro introductorio a UML-Addison Wesley

- Metodología Xp, IUTY 072301 PNF en Informática ,<http://ingsoftware072301.obolog.es/metodologia-xp-2012877>
- Rumbaugh, J. (1999). El Lenguaje Unificado del Modelado. Madrid, España: Addison Wesley.
- Stevens, P. (2002). Utilización de UML en ingeniería de Software con Objetos y Componentes. Madrid: Pearson Education.



GLOSARIO

- **La Nota de Remisión:** es aquel documento que se utiliza casi exclusivamente a instancias de un contexto mercantil para acreditar o dejar constancia de la entrega de un pedido. *Referencia: Definiciones ABC, Definición de Nota de Remisión, (2007) disponible en: <http://www.definicionabc.com/general/nota-de-remision.php> [citado el 9 de septiembre de 2015].*
- **Sistema de información:** es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una necesidad o un objetivo. *Referencia: wikipedia, sistema de información, (2007) disponible en: https://es.wikipedia.org/wiki/Sistema_de_información, [citado el 21 de septiembre de 2015].*
- **Software:** al equipo lógico o soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware. *Referencia: wikipedia, Software, (2005) disponible en: <https://es.wikipedia.org/wiki/Software>, [citado el 21 de septiembre de 2015].*
- **Interfaz:** es lo que conocemos en inglés como interface (“superficie de contacto”). En informática se utiliza para nombrar a la conexión física y funcional entre dos sistemas o dispositivos de cualquier tipo dando una comunicación entre distintos niveles. *Referencia: wikipedia, Interfaz, (2005) disponible en: <https://es.wikipedia.org/wiki/Interfaz>, [citado el 21 de septiembre de 2015].*
- **Base de datos:** na base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico. *Referencia: masadelante., Definición de Base de datos, (2005) disponible en: <https://www.masadelante.com/faqs/base-de-datos>, [citado el 21 de septiembre de 2015].*
- **Metodología xp:** Metodología liviana de desarrollo de software
 - Conjunto de prácticas y reglas empleadas para desarrollar software
 - Basada en diferentes ideas acerca de cómo enfrentar ambientes muy cambiantes
 - Originada en el proyecto C3 para Chrysler
 - En vez de planificar, analizar y diseñar para el futuro distante, hacer todo esto un poco cada vez, a través de todo el proceso de desarrollo
 - Objetivos:
 - + Establecer las mejores prácticas de Ingeniería de Software en los desarrollo de proyectos.
 - + Mejorar la productividad de los proyectos.

+Garantizar la Calidad del Software desarrollando, haciendo que este supere las expectativas del cliente. *Referencia: Universidad Unión Bolivariana., programación extrema xp, (2015) disponible en: http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html, [citado el 21 de septiembre de 2015].*

- **CodeIgniter:** CodeIgniter se basa en el patrón de desarrollo Model-View-Controller. MVC es un enfoque de software que separa la lógica de aplicación de la presentación. En la práctica, permite que las páginas web que contienen secuencias de comandos mínima ya que la presentación está separado del scripting PHP.

- El modelo representa sus estructuras de datos. Normalmente las clases del modelo contendrán las funciones que le ayudan a recuperar, insertar, y actualizar la información en su base de datos.

- La vista es la información que se presenta a un usuario. Una visión será normalmente una página web, pero en CodeIgniter, una opinión también puede ser un fragmento de la página como un encabezado o pie de página. También puede ser una página de RSS, o cualquier otro tipo de "página".

- El controlador actúa como un intermediario entre el Modelo, la Vista, y todos los demás recursos necesarios para procesar la petición HTTP y generar una página web. *Referencia: ellislab., CodeIgniter User Guide Version 2.2.0, (2014) disponible en: <https://ellislab.com/codeigniter/user-guide/overview/mvc.html>, [citado el 21 de septiembre de 2015].*

- **Framework:** Un Framework es un conjunto estandarizado de conceptos, prácticas y criterios para hacer frente a un tipo común de problema, que puede ser usado para ayudarnos a resolverlo de forma rápida y eficaz.

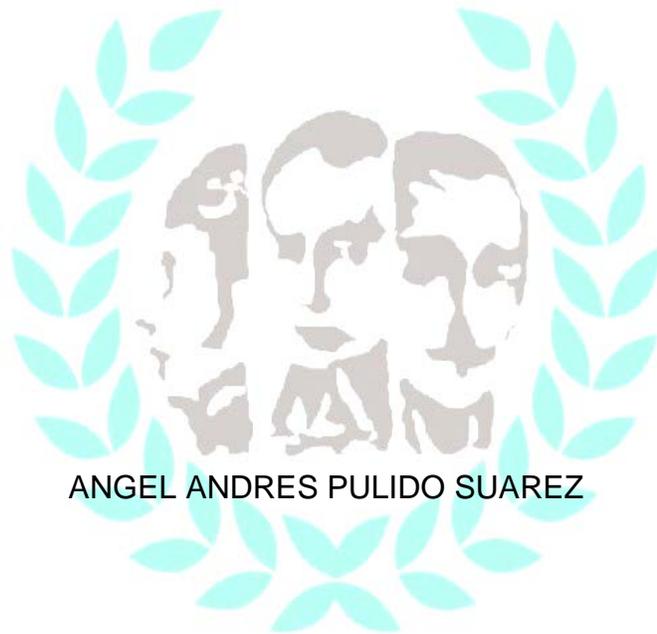
El objetivo de los Frameworks es proporcionar una estructura común, de modo que los desarrolladores no tienen que hacer el código de cero cada vez y puede volver a utilizar la gran mayoría. *Referencia: Edgar Gómez servicios informáticos., Qué es un Framework, (2013) disponible en: <http://edgargomez.es/que-es-un-framework/#.Vkysnb8tPaU>, [citado el 21 de septiembre de 2015].*

- **Tarjetas CRC:** La utilización de tarjetas CRC (Class-Responsibility-Collaboration) es una técnica de diseño orientado a objetos propuesta por Kent Beck (introducido de la metodología de programación extrema) y Ward Cunningham.

El objetivo de la misma es hacer, mediante tarjetas, un inventario de las clases que vamos a necesitar para implementar el sistema y la forma en que van a interactuar, de esta forma se pretende facilitar el análisis y discusión de las mismas por parte de varios actores del equipo de proyecto con el objeto de que el diseño sea lo más simple posible verificando las especificaciones del sistema. . *Referencia: Jummp., Gestión de proyectos y desarrollo de software (2012) disponible en: <https://jummp.wordpress.com/2012/01/10/desarrollo-de-software-tarjetas-crc/>, [citado el 1 de octubre de 2015].*



SISTEMA DE INFORMACION PARA EL CONTROL DE LAS REMISIONES
DE LA EMPRESA GRUPO AMAREY



FUNDACION UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERIA DE SISTEMAS
INGENIERIA DE SISTEMAS
BOGOTA D.C.
2015

SISTEMA DE INFORMACION PARA EL CONTROL DE LAS REMISIONES
DE LA EMPRESA GRUPO AMAREY

ANGEL ANDRES PULIDO SUAREZ

PROYECTO PRESENTADO PARA OPTAR AL TÍTULO DE INGENIERO DE
SISTEMAS



ASESOR
MIGUEL ARMANDO HERNÁNDEZ BEJARANO
INGENIERO DE SISTEMAS

FUNDACION UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERIA DE SISTEMAS
INGENIERIA DE SISTEMAS
BOGOTA D.C.
2015

TABLA DE IMÁGENES

Imagen 1. Validación automática de perfil de usuarios.....	92
Imagen 1.3, Inserción base de datos.....	93
Imagen 1.4, Vista navegadores.....	94
Imagen 1.5, Validación usuarios.....	94
Imagen 1.6, Debug a las funciones.....	95
Imagen 2. Configuración url.....	110
Imagen 2.1, Conexión base de datos.....	110
Imagen 3. Requerimientos.....	111
Imagen 3.1. Archivos dll.....	111
Imagen 3.2. Activación del servicio.....	112
Imagen 4. Trigger tabla actualiza.....	114
Imagen 4.1, trigger tabla actualizakits.....	114
Imagen 4.2, trigger tabla instrumental.....	114
Imagen 4.3, trigger tabla kits.....	115
Imagen 4.4, trigger tabla precio_intru.....	115
Imagen 5. Tabla actualizakits.....	116
Imagen 5.1, tabla actualiza.....	117
Imagen 5.3, tabla AdicionalesRef.....	117
Imagen 5.4, tabla cargue.....	118
Imagen 5.5, tabla clientes.....	118
Imagen 5.6, tabla destinatarios.....	119
Imagen 5.7, tabla histórico.....	119
Imagen 5.8, tabla instrumental.....	120
Imagen 4.9, tabla Kit.....	120
Imagen 5.10, tabla Kits.....	120
Imagen 5.11, tabla material.....	121
Imagen 5.12, tabla precio_intru.....	121
Imagen 5.13, tabla preciosap.....	122
Imagen 5.14, tabla profiles.....	122
Imagen 5.15, tabla users.....	123

ANEXOS

ANEXO A ESTÁNDARES

Los estándares son una buena práctica de programación que no sólo se debe utilizar cuando se trabaja con la metodología XP sino con cualquier clase de metodología que se aplique.

Lo que se buscó con los siguientes estándares fue facilitar la comprensión del código y así lograr la propiedad colectiva de este.

- Estándares en la base de datos
 - Los nombres de las tablas se escriben en mayúscula.
 - Los nombres de los campos de cada tabla se escriben en mayúscula, iniciando con el nombre del campo, seguidos por guion bajo y el nombre de la tabla a la cual pertenece.

Ejemplo: Tabla: CLIENTE

Campos: ID_CLIENTE

DESCRIPCION_CLIENTE

- Estándares en el código
 - Nombres de variables: los elementos visuales comienzan con las iniciales del tipo de objeto.

Ejemplo:

```
name="DESCRIPCION_KIT"
```

```
var cliente = $("#cliente option:selected").val();
```

```
jQuery("#listmaterial")
```

- El código debe estar indentado correctamente.

- Las páginas tienen como convención de nombres “historial”, siendo la palabra Función reemplazada por la correspondiente tarea (busca, editar, etc.) y la palabra tabla reemplazada por el nombre de la tabla a la cual se aplica dicha tarea.

Nota: Todos los nombres de las variables, tablas, funciones, etc. Son muy nemónicos e intuitivos acogiéndose así a la simplicidad según plantea XP.



ANEXO B PRUEBAS

Las pruebas son uno de los elementos más importantes de XP y en tal sentido surgieron algunas consideraciones relacionadas con sus características, las cuales se describen a continuación.

- **Pruebas autónomas.**

Uno de los requisitos de XP para las pruebas unitarias se refiere a que éstas deben correr en forma autónoma, es decir que no requieran la intervención humana para determinar si han sido exitosas.

Existen una variedad de rutinas que por su forma de funcionar se hace complejo hacerle pruebas. Entre ellas cabe resaltar la obtención de la hora del sistema. En este caso hacer una prueba que evalúe su éxito requerirá bien sea la participación humana para determinar que la hora sea correcta o la ejecución de la misma rutina dentro del código de la prueba, lo que no tiene sentido. Elementos visuales como ventanas, botones representan un reto adicional para ser probados de forma autónoma, ya que la intervención humana se hace indispensable para determinar el éxito de la prueba. En estos casos es mucho más práctico hacer las pruebas manualmente. Ejemplo que se cargó el menú de forma automática dependiendo del perfil del usuario. Ver imagen 1.



Imagen 1, Validación automática de perfil de usuarios

- **Prueba antes que código.**

Realizar la prueba antes que el código hace que éste sea más fácil de construir, se identifiquen con mayor claridad los casos especiales y quede más elegante Al construir una prueba queda más clara la funcionalidad que debe implementarse por lo cual la construcción de la rutina requiera de menos tiempo. Por otro lado y producto de dicha claridad se evidencian los casos especiales que debe considerar dicha rutina lo cual la hace más completa y tolerante a fallos, requisito indispensable de una rutina bien hecha. Finalmente, gracias a lo anterior, el resultado del proceso de codificación es más organizado, haciéndolo fácil de entender.

- **Base de datos.**

Al terminar el diseño y la implementación de la base de datos, se insertaron datos directamente en ésta y se realizaron consultas por medio del código de SQL para verificar que el modelo entidad relación estaba bien diseñado. Ejemplo: creación de usuario, se valida en la base de datos users, ver imagen 1.3.

IDUSER	LOGINUSERSL	IDPROFILE	PASSUSER	ACTIVO	NOMBRE	CEDULA	CORREO
22	DSFSDF	1	SDFDF	A	dfsdfs	4444444	dsfsd@fe.vom
21	YO	2	999999	A	YO	99999999	99@9.COM
19	FGHFG	2	789+9	A	GHHFGHHFGHH	9999999	GHHFGH@MKSDFMK.COM
18	JAJ	2	4545465	A	NULL	NULL	jaj@jaj.com
17	PEPE	2	4444444	A	NULL	NULL	pepe@pepe.com

Imagen 1.3, Inserción base de datos

- **Módulos.**

Al terminar cada módulo el equipo de desarrollo realizó las pruebas correspondientes para verificar que cada módulo cumpliera con sus tareas de forma correcta, fue así como por ejemplo en el módulo de inserción de datos se verificó que los datos ingresados quedaran correctamente guardados en la base de datos.

Las pruebas de aceptación se hicieron cada vez que se sacó una nueva versión, consistía en entregar al cliente y/o terceras personas dicha versión para que estos ingresaran, datos realizaran consultas y/o ediciones según fuera el caso para ver si la aplicación respondía bien.

Por tratarse de una aplicación web una de las pruebas más importantes fue la navegación, y los tiempos de respuesta, estas pruebas se hicieron en tres de los navegadores más conocidos e importantes como lo son Chrome, Safari y Mozilla Firefox. Y dieron como resultado un funcionamiento óptimo en los tres navegadores con una con una resolución 1024 x 768 pixeles. Ver(Imagen 1.4).

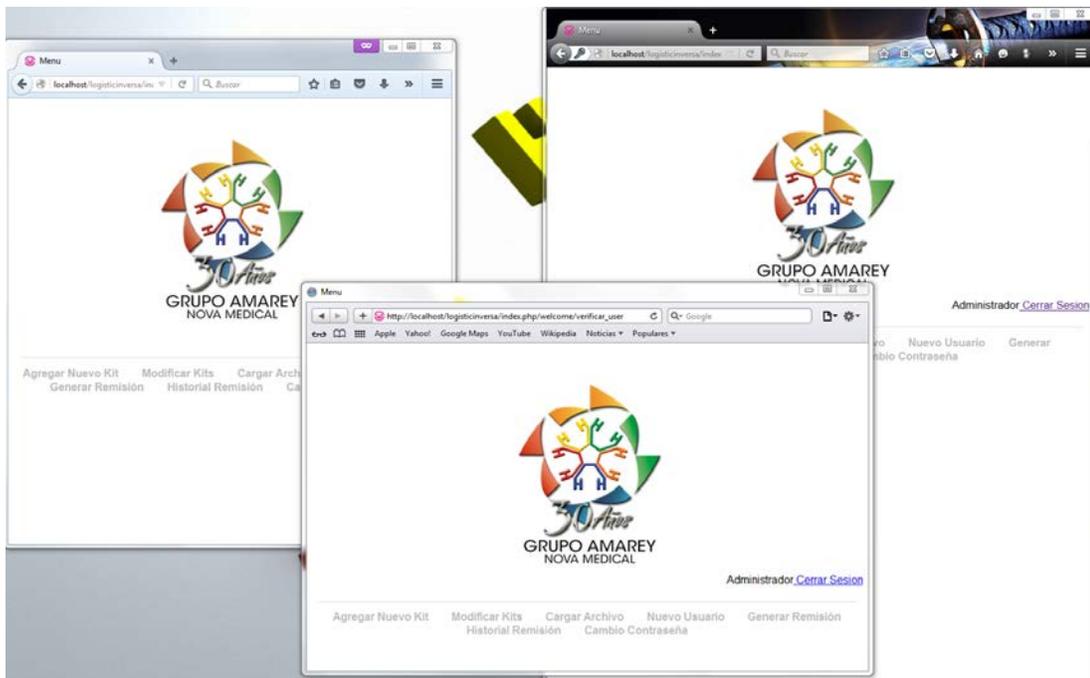


Imagen 1.4, Vista navegadores.

- **Probar gradualmente.**

Teniendo en cuenta lo anteriormente dicho, las pruebas deben ser elaboradas antes que el código, pero debe considerarse la conveniencia de la realización de la totalidad de las pruebas de una clase o la realización de cada una de ellas con su respectiva comprobación. Esta última posición presenta la ventaja de una constante corroboración del código de modo tal que se puede trabajar con la seguridad de que se está implementando sobre un código que funciona. Ejemplo: que al ingresar el usuario valide las credenciales, ver imagen 1.5.

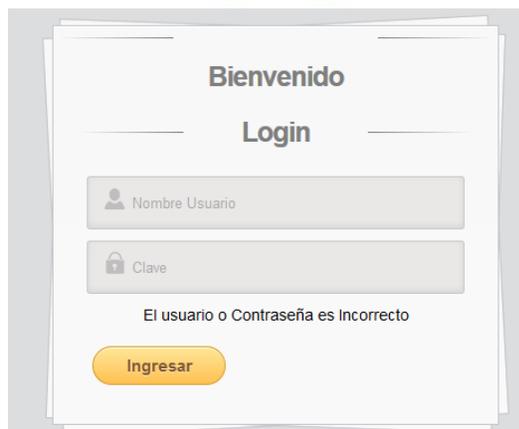


Imagen 1.5, Validación usuarios

- Probar al encontrar un error.

Aunque XP plantea la realización de pruebas en la medida en que surgen errores, poner en marcha esta práctica consume demasiado tiempo. Resulta más conveniente la revisión de las pruebas existentes para determinar a cuál de éstas le incumbe el error que se presenta. En caso de no encontrar dicha prueba, se recurrirá a la realización de una prueba especial para el error. De esta forma se evita la aparición de una gran cantidad de pruebas aisladas, preservando la organización de las mismas.

Por ejemplo al momento de cargar la lista en el formulario de nuevo kit, se hace seguimiento a la función que llena la grilla o tabla en la vista, en buscar e erros de carga o envió de paramentos, ver imagen 1.6.

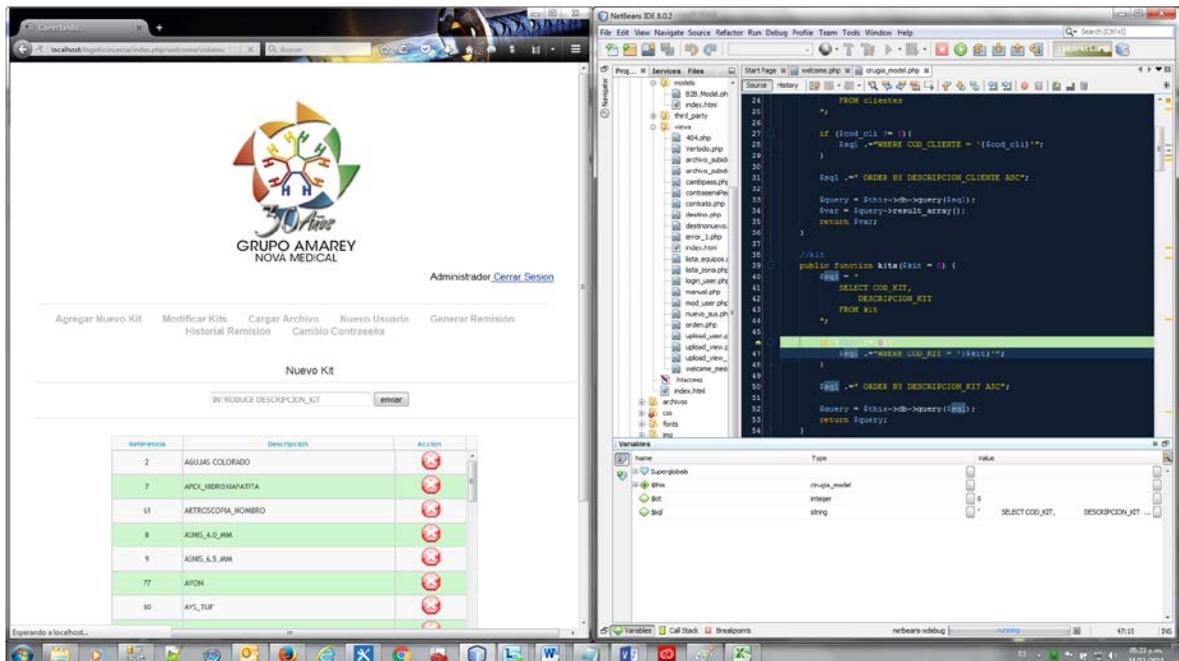


Imagen 1.6, debug a las funciones

ANEXO C CASOS DE USO

En el siguiente documento se describen los casos de uso identificados para el software de REMISIONES.

Los casos de uso documentan el comportamiento del sistema desde el punto de vista del usuario. El primer paso para realizar los casos de uso es identificar los actores que están asociados a una serie de roles, entendiendo como actor entidades que interactúan con el sistemas realizando actividades específicas, estos actores pueden ser personas, dispositivos de hardware u otros sistemas tales como un Webservice.

A continuación se describen los casos de uso más relevantes del sistema, en estos se describe el requerimiento asociado, los actores implicados, las condiciones previas y posteriores, las excepciones, los flujos principales.

- **Administrar usuario**

Nombre:	
Nombre:	Administrar Usuarios.
Actores:	Administrador
Precondiciones:	<ul style="list-style-type: none"> · El actor se ha autenticado en el sistema. · El actor debe estar creado en la aplicación con el rol de administrador.
Poscondiciones:	<ul style="list-style-type: none"> · El usuario ha sido creado en el sistema. · El usuario ha sido actualizado en el sistema.
Flujo normal de los eventos:	
<ol style="list-style-type: none"> 1. El sistema muestra el formulario para crear o actualizar usuarios. 2. El actor selecciona la opción de nuevo. 3. El sistema muestra los campos que van a estar asociados al usuario. 4. El actor selecciona guardar. 5. El sistema almacena la información. 6. El sistema confirma la transacción. 	

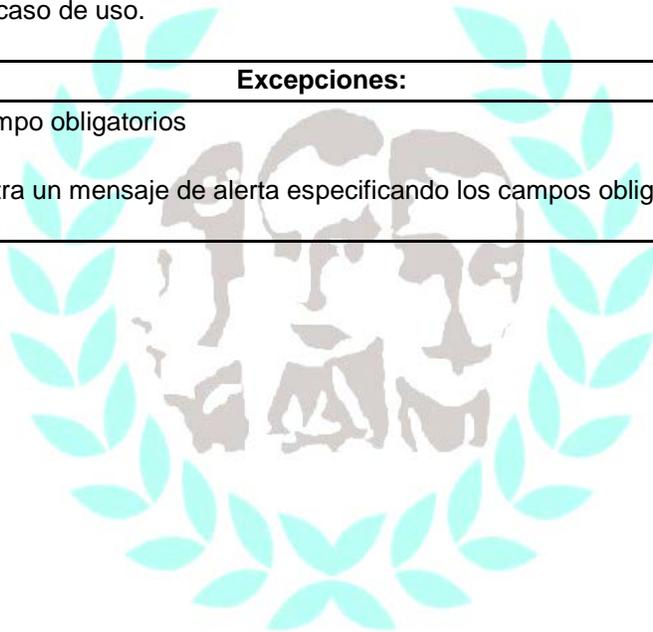
7. El actor sale del caso de uso.

Flujo alternos:

- A. El actor selecciona la opción editar usuario.
 - 1. El sistema muestra los datos que se pueden actualizar.
 - 2. El actor cambia los campos que requiera.
 - 3. El actor selecciona guardar.
 - 4. El sistema almacena los cambios.
 - 5. El actor sale del caso de uso.

Excepciones:

- A. Validación de campo obligatorios
 - 1. El sistema muestra un mensaje de alerta especificando los campos obligatorios.



• **Autenticación del usuario**

Nombre:	Autenticacion Usuarios.
Actores:	Administrador
Precondiciones:	<ul style="list-style-type: none"> · El actor se ha autenticado en el sistema. · El actor debe estar creado en la aplicación con un rol
Poscondiciones:	<ul style="list-style-type: none"> · El usuario ha sido creado en el sistema. · El usuario ha sido actualizado en el sistema.
Flujo normal de los eventos:	
<ol style="list-style-type: none"> 1. El sistema muestra el formulario para ingreso al sistema. 2. El sistema muestra los campos asociados al usuario. 3. El actor ingresa los datos 4. El actor selecciona ingresar. 5. El sistema valida la información. 6. El sistema confirma la transacción. 7. El actor sale del caso de uso. 	
Flujo alternos:	
Excepciones:	
<p>A. Validación de campo obligatorios</p> <ol style="list-style-type: none"> 1. El sistema muestra un mensaje de alerta especificando los datos incorrectos. 	

• Administrar kits y elementos

Nombre:	Administracion Kits y elementos
Actores:	Administrador
Precondiciones:	<ul style="list-style-type: none"> · El actor se ha autenticado en el sistema. · El actor debe estar creado en la aplicación con un rol
Poscondiciones:	<ul style="list-style-type: none"> · El usuario ha sido creado en el sistema. · El usuario ha sido actualizado en el sistema.
Flujo normal de los eventos:	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción modificar kit 2. El sistema muestra el formulario 3. El actor selecciona el kit 4. El sistema muestra los materiales del kit 5. El actor selección la opción, editar o eliminar 6. El sistema confirma la transacción. 7. El actor selección la opción, editar, cambia solo el nombre del material 8. El actor selecciona guardar. 9. El sistema almacena la información. 10. El sistema confirma la transacción. 11. El actor selección la opción, eliminar, se quita el material del kit 12. El actor selecciona guardar. 13. El sistema almacena la información. 	

<p>14. El sistema confirma la transacción.</p> <p>15. El actor sale del caso de uso.</p>
Flujo alternos:
<p>A. El actor selecciona la opción cargar archivo si desea una modificación masiva.</p>
Excepciones:

• **Administrar kit**

Nombre:	Administracion Kit
Actores:	Administrador
Precondiciones:	<ul style="list-style-type: none"> · El actor se ha autenticado en el sistema. · El actor debe estar creado en la aplicación con un rol
Poscondiciones:	<ul style="list-style-type: none"> · El usuario ha sido creado en el sistema. · El usuario ha sido actualizado en el sistema.
Flujo normal de los eventos:	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción agregar nuevo kit 2. El sistema muestra el formulario y la lista de kits existentes 3. El actor ingresa el nombre del kit 4. El actor selecciona guardar. 5. El sistema almacena la información. 6. El sistema confirma la transacción. 7. El usuario selecciona la opción cargar archivo 8. El sistema muestra el formulario para cargar el archivo y un botón para descargar las plantilla aceptada por el sistema 9. El actor selecciona el archivo con la información delo materiales que van a estar asociados al kit (cod_kit, material, cod_material, cantidad ubicación, si es instrumental o no, 	

etc)
<ol style="list-style-type: none"> 10. El actor selecciona enviar. 11. El sistema almacena la información. 12. El sistema confirma la transacción. 15. El actor sale del caso de uso.
Flujo alternos:
Excepciones:

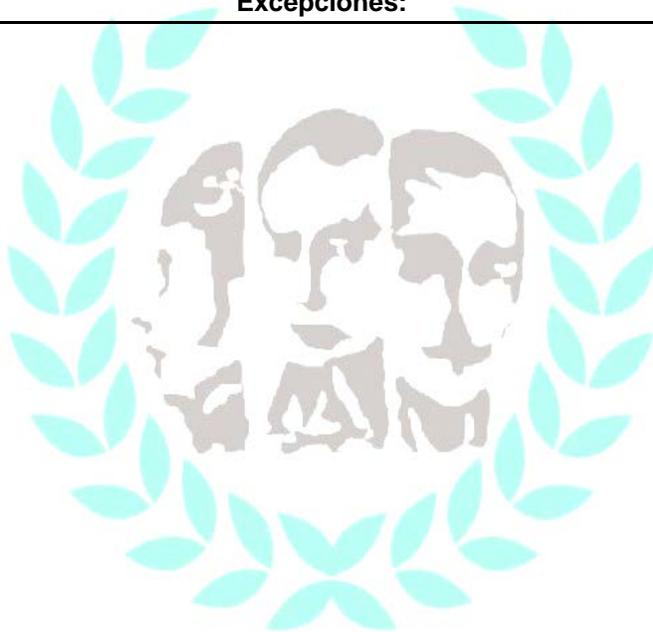
• **Creación de las remisiones y Datos Adicionales de la remisión**

Nombre:	Creación de las remisiones y Datos Adicionales de la remisión
Actores:	Usuario
Precondiciones:	<ul style="list-style-type: none"> · El actor se ha autenticado en el sistema. · El actor debe estar creado en la aplicación con un rol
Poscondiciones:	<ul style="list-style-type: none"> · El usuario ha sido creado en el sistema. · El usuario ha sido actualizado en el sistema.
Flujo normal de los eventos:	
	<ol style="list-style-type: none"> 1. El usuario selecciona la opción genera remisión 2. El sistema muestra el formulario 3. El actor selecciona institución, kit y motor que es opcional 4. El actor selecciona consultar 5. El sistema muestra la información 6. El actor selecciona los elementos del kit 7. El actor selecciona diligenciar 8. El sistema procesa la información y muestra un nuevo formulario 9. El actor diligencia los campos, destinatario, paciente, doctor, historia clinica, fecha cirugía, y adicionales que es opcional 10. El actor selecciona generar pdf

11. El sistema almacena la información.
12. El sistema confirma la transacción.
13. El actor selecciona ver
14. El sistema muestra el pdf
15. El actor selecciona aceptar
16. El actor sale del caso de uso.

Flujo alternos:

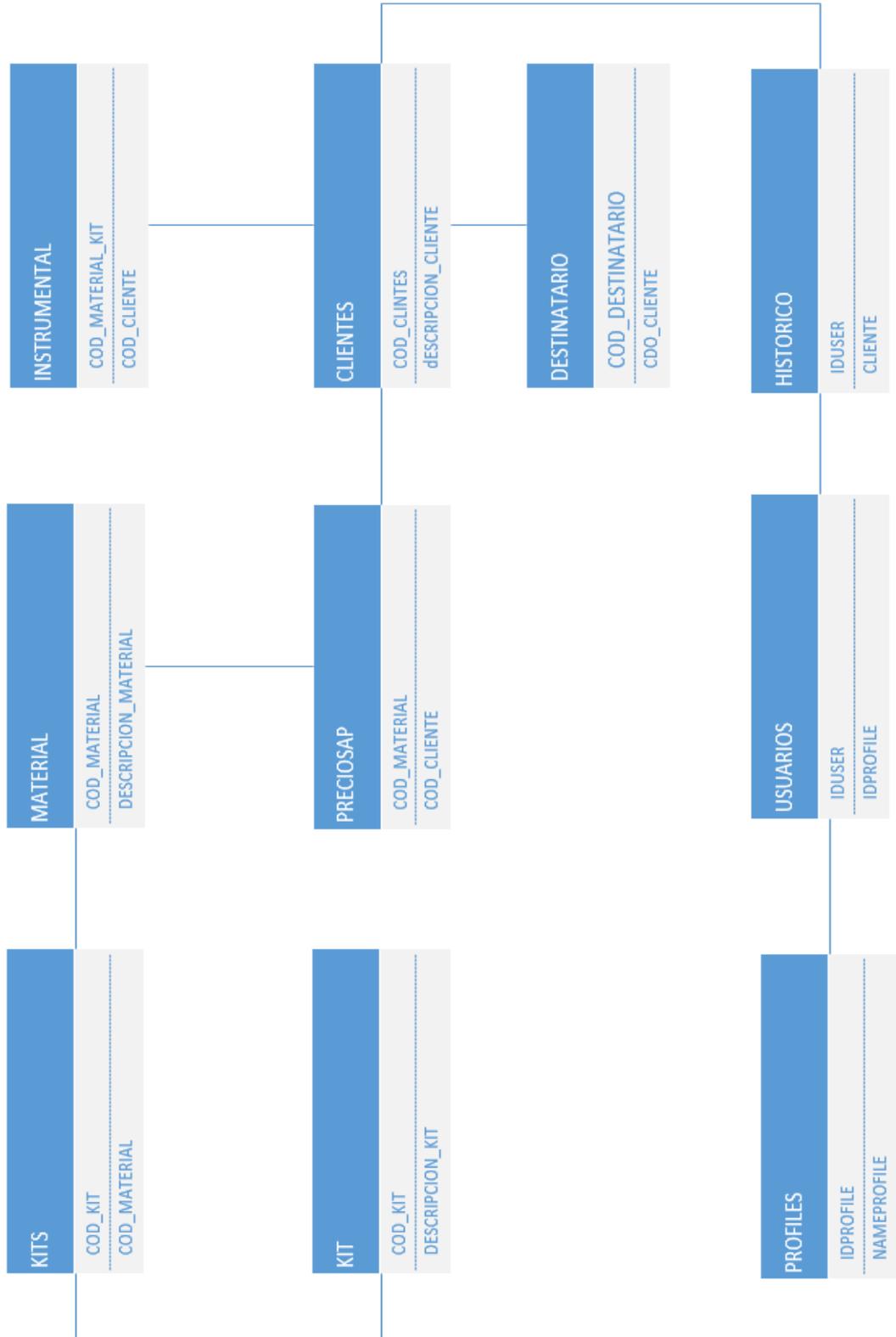
Excepciones:



- Consultar reporte de historial de remisiones.

Nombre:	Consultar reporte de historial de remisiones.
Actores:	Administrador - usuario
Precondiciones:	<ul style="list-style-type: none"> · El actor se ha autenticado en el sistema. · El actor debe estar creado en la aplicación con un rol
Poscondiciones:	<ul style="list-style-type: none"> · El usuario ha sido creado en el sistema. · El usuario ha sido actualizado en el sistema.
Flujo normal de los eventos:	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción Historial remisiones 2. El sistema muestra el formulario 3. El actor selecciona cualquiera de los parámetros de filtro (Numero remisión, cliente, fecha cirugía, fecha creación, paciente, doctor) 4. El actor selecciona consultar. 5. El sistema valida la información. 6. El sistema muestra la información. 7. El usuario selecciona la información buscada. 8. El usuario selecciona ver pdf, para ver la remisión. 9. El sistema muestra el pdf. 10. El actor selecciona limpiar. 11. El actor sale del caso de uso. 	
Flujo alternos:	
Excepciones:	

ANEXO D DIAGRAMA DE CLASES



- **Tarjetas CRC**

KIT	
RESPONSABILIDAD	COLABORACION
validar la existencia del kit mostrar el kit eliminar kit crear kit	administrador administrador

KITS	
RESPONSABILIDAD	COLABORACION
validar la existencia del material para el kit mostrar los materiales del kit modificar los materiales Del kit eliminar los materiales del kit insertar materiales al kit relacionar material con kit	administrador administrador administrador trigger SQL cliente cliente cliente

MATERIAL	
RESPONSABILIDAD	COLABORACION
validar la existencia del material insertar materiales	trigger SQL trigger SQL

PRECIOSAP	
RESPONSABILIDAD	COLABORACION
muestra precio del producto por cliente actualizar precio del material por cliente insertar nuevo material en tabla material insertar cliente en tabla cliente	interface SAP trigger SQL trigger SQL

CLIENTE	
RESPONSABILIDAD	COLABORACION
mostrar cliente relacionar cliente con destinatario relacionar cliente con todos los materiales	administrador administrador

INSTRUMENTAL	
RESPONSABILIDAD	COLABORACION
mostrar instrumental relacionar instrumental por cliente relacionar material con grupo de instrumental	trigger SQL trigger SQL

DESTINATARIO	
<p>RESPONSABILIDAD</p> <p>relacionar cliente con destinatario actualizar destinatario con cliente</p>	<p>COLABORACION</p> <p>administrador administrador</p>

HISTORICO	
<p>RESPONSABILIDAD</p> <p>mostrar historial de remisiones mostrar pdf relacionada a la remisión</p>	<p>COLABORACION</p> <p>módulo histórico módulo histórico</p>

USUARIOS	
<p>RESPONSABILIDAD</p> <p>mostrar usuario crear usuario cambiar contraseña inactivas usuario</p>	<p>COLABORACION</p> <p>administrador administrador administrador</p>

PROFILES	
RESPONSABILIDAD asignar perfil al usuario mostrar opciones del menú según perfil	COLABORACION administrador



ANEXO E INSTALACIÓN

A continuación se describe los requisitos mínimos para el funcionamiento de la aplicación, tales como los sistemas operativos compatibles, los requisitos de hardware, motor de base de datos, versión de Framework y navegadores web compatibles entre otros

Tabla 1 Requisitos mínimos para el funcionamiento de la aplicación.

Sistema operativo compatibles	
•	Windows Server 2003 SP2
•	Windows Server 2008 (no admitido en el rol Server Core)
•	Windows 7
•	Windows Server 2008 R2 (no admitido en el rol Server Core)
•	Windows 7 SP1
•	Windows Server 2008 R2 SP1
Requisitos de hardware	
•	Mínimo recomendado: Pentium 4 o superior con 2GB de RAM o más
•	Espacio en disco mínimo para la publicación de la aplicación web: 50MB
Motor de base de datos	
•	Microsoft SQL Server 2008 R2
Navegadores Compatibles	
•	Firefox Mozilla
•	Windows Internet Explorer
•	Google Chrome
•	Safari
Entorno de desarrollo	
•	Wampserver 2.4
•	Netbeans
Framework	
•	CodeIgniter-2.2.2
otros	
•	extension=php_sqlsrv_54_ts.dll
•	extension=php_pdo_sqlsrv_54_ts.dll
•	ntwdblib.dll
•	sqlncli.msi

El procedimiento para instalar el programa es el siguiente:

- Una vez instalados los programas y configurado los sistemas, como se indica en el anexo f.

- Se debe pegar la carpeta del proyecto en C:\wamp\www.
- Al abrir netbeans se debe ingresar a la carpeta config, luego a config.php y cambiar la ruta de acceso a la aplicación, ver (Imagen 2)

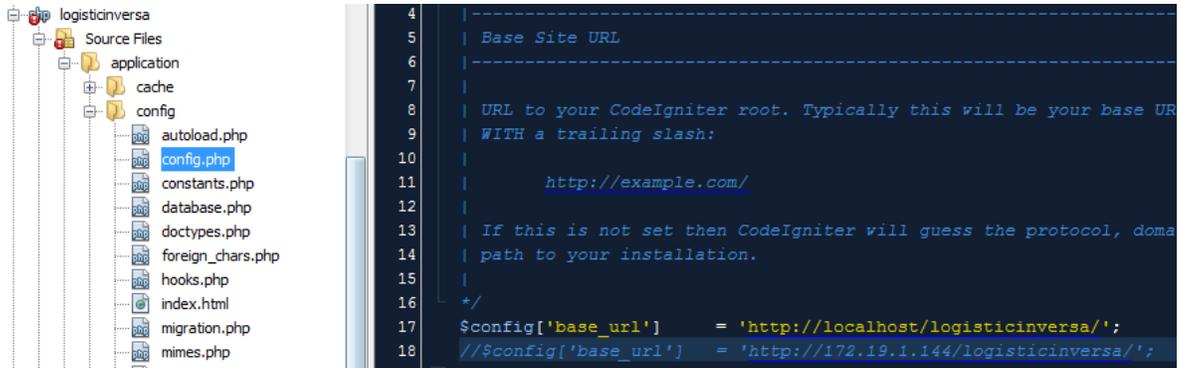


Imagen 2. Configuración url.

- En netbeans se debe ingresar a la carpeta config, luego a database.php y cambiar nombre del servidor, usuario y contraseña de acceso a la base de datos, ver (Imagen 2.1)

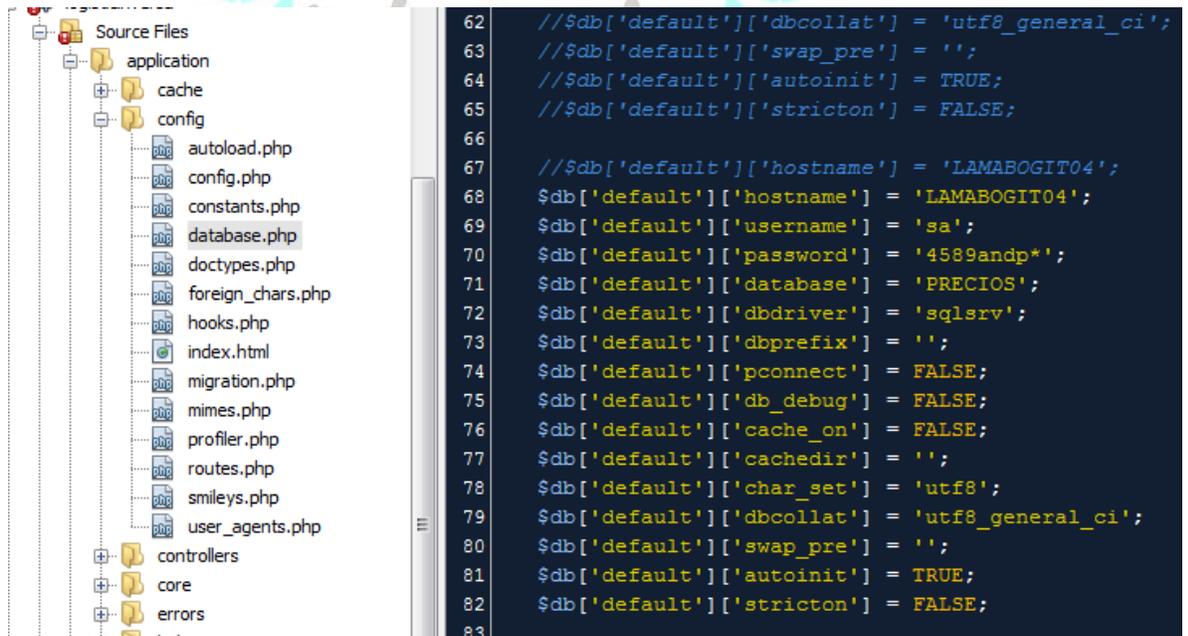


Imagen 2.1, Conexión base de datos.

ANEXO F CONEXIÓN SQL SERVER – WAMPSERVER

Se Debe descargar el driver oficial ofrecido por Microsoft, Lo pueden buscar en google o pueden acceder en el siguiente link <http://www.microsoft.com/en-us/download/details.aspx?id=20098>.

Existen 2 versiones la 2.0 y la 3.0, cuál usar depende del Sistema Operativo que se esté utilizando, a continuación los requerimientos para el driver versión 3.0. Ver (Imagen 3)

System Requirements

Supported Operating System

Windows 7, Windows Server 2008 R2, Windows Server 2008 Service Pack 2, Windows Vista Service Pack 2

Windows 7, Windows Server 2008 R2, Windows Server 2008 Service Pack 2, Windows Vista Service Pack 2

- PHP 5.3.6 or PHP 5.4. For more information about downloading and installing PHP, visit PHP on Windows.
- Microsoft SQL Server 2012 Native Client available in the SQL Server 2012 Feature Pack.
- Any edition of SQL Server 2005 or later.
- A Web server configured to run PHP.

Imagen 3. Requerimientos

Después de descargar los driver pasamos a descomprimir la carpeta, la cual contiene 8 archivos con extensión .dll para la versión 5.3 y 5.4, además de unos pequeños instructivos. Ver (Imagen 3.1)

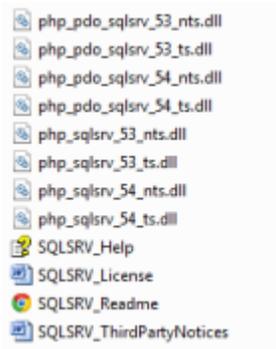


Imagen 3.1. Archivos dll

Elegimos los dos archivos dll: php_pdo_sqlsrv_54_ts.dll y php_sqlsrv_54_ts.dll, porque la versión PHP del proyecto es la 5.4; Ahora que ya sabemos que archivos servirán se deben de pegar en la carpeta ext de php.

Ahora se debe habilitar los drivers en el archivo de configuración de PHP, el archivo php.ini, nos situamos al final del documento y agregamos las siguientes líneas.

```
extension=php_sqlsrv_54_ts.dll
```

```
extension=php_pdo_sqlsrv_54_ts.dll
```

Hecho esto se guardan los cambios en el archivo php.ini y se reinician los servicios de Apache. Si se consulta la información mediante phpinfo() debe de aparecer el driver ya habilitado, de lo contrario se debe de revisar si se ha elegido bien los archivos .dll y haber escrito bien el archivo de configuración. Ver (Imagen 3.2).

pdo_sqlsrv

pdo_sqlsrv support		enabled
Directive	Local Value	Master Value
pdo_sqlsrv.client_buffer_max_kb_size	10240	10240
pdo_sqlsrv.log_severity	0	0

Imagen 3.2. Activación del servicio

Adicional se debe actualiza la versión de SQL server 2008 r2, que es la que se tiene para el proyecto y descargar el cliente nativo de SQL Server 2012

 Microsoft SQL Server 2008 R2 Native Client

ANEXO G MERGE

En la base de datos se desarrollaron varios trigger sobre las tablas para actualizar información de manera automática entre las diferentes tablas, para realizar la validación de la información se utilizaron los merge: “Realiza operaciones de inserción, actualización o eliminación en una tabla de destino según los resultados de una combinación con una tabla de origen. Por ejemplo, puede sincronizar dos tablas insertando, actualizando o eliminando las filas de una tabla según las diferencias que se encuentren en la otra.” Referencia MERGE (Transact-SQL) <https://msdn.microsoft.com/es-es/library/bb510625%28v=sql.120%29.aspx>

Dentro de lo trigger que están en la base de datos se imprimió los merge utilizando una estructura como la siguiente:

```
USE [PRECIOS]
GO
/***** Object: Trigger [dbo].[ACTUALIZA.CLI]  Script Date: 11/13/2015
14:29:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-----
ALTER TRIGGER [dbo].[ACTUALIZA.CLI]
ON [dbo].[ACTULIZA]
AFTER INSERT
AS
BEGIN

MERGE dbo.CLIENTES as target
using (SELECT DISTINCT codcliente,nit,nombre,ciudad FROM
[PRECIOS].[dbo].[ACTULIZA])as source
    on (target.COD_CLIENTE = source.codcliente)

when matched then
    update
    set NIT_CLIENTE = source.nit

    when not matched then

        insert ( COD_CLIENTE ,NIT_CLIENTE ,DESCRIPCION_CLIENTE ,CIUDAD)
        values (
source.[codcliente],source.[nit],UPPER(source.[nombre]),UPPER(source.[ciudad]))

OUTPUT $action, inserted.*;

END
```

Lista de tablas que contienen trigger

- ACTUALIZA, actualiza o inserta la información en la tabla, clientes, materiales y presiosap ver imagen 4.

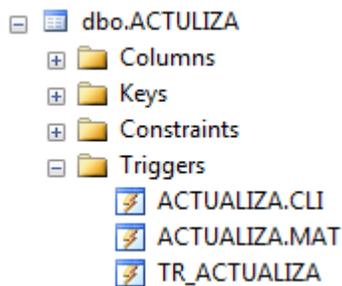


Imagen 4. trigger tabla actualiza

- ACTUALIZAKITS, actualiza o inserta los materiales de los kits, ver imagen 4.1

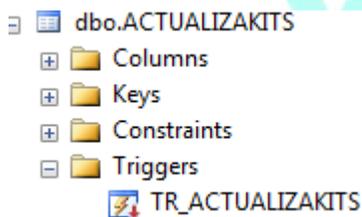


Imagen 4.1, trigger tabla actualizakits

- INSTRUMENTAL, inserta o actualiza en la tabla presiosap los precios para los instrumentales por cliente, ver imagen 4.2

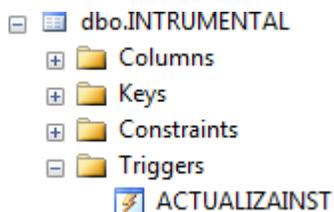


Imagen 4.2, trigger tabla instrumental

- KITS, BUSCANOREGISTRO, busca nuevos registros y los inserta en la tabla precio_intru, ver imagen 4.3

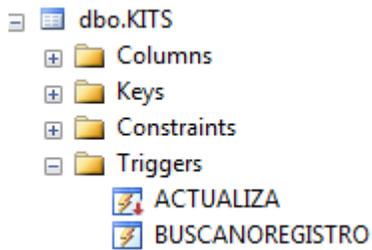


Imagen 4.3, trigger tabla kits

- PRECIO_INTRU, inserta en la tabla instrumental los instrumentales por cliente, ver imagen 4.4

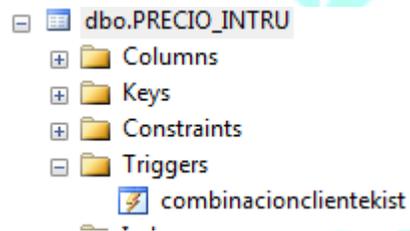


Imagen 4.4, trigger tabla precio_intru

ANEXO H DICCIONARIO DE DATOS

En el diccionario de datos se describen todas las tablas de la base de datos, dicha descripción contiene una breve explicación de los datos que debe almacenar la tabla, también contiene los nombre de los campos con sus tipos de datos, restricciones, valores por defecto, además indica si los campos son llaves primarias o foráneas, si el campo recibe datos nulos, si son únicos o si el campo se genera de forma auto numérica, en resumen el diccionario de dato contiene la descripción de la estructura de cada tabla de la base de datos.

- ACTUALIZAKITS: en esta tabla se cargar el archivo csv, con los nuevos kits o actualización que se van a realizar sobre este, la estructura de la tabla se ve en la imagen 5.

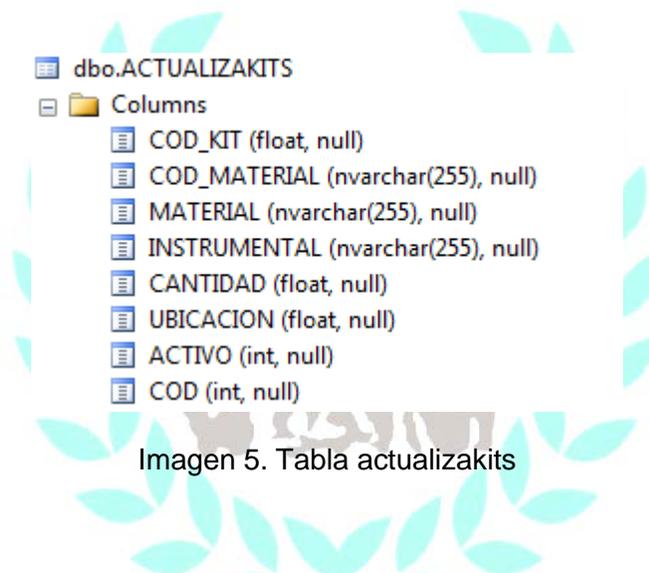


Imagen 5. Tabla actualizakits

- ACTUALIZA: en esta tabla se actualiza con la información que proviene de la tabla carga. En esta tabla se valida la información y se disparan los trigger que actualizan e insertan la data en la diferentes tablas relacionadas., la estructura de la tabla se ve en la imagen 5.1.

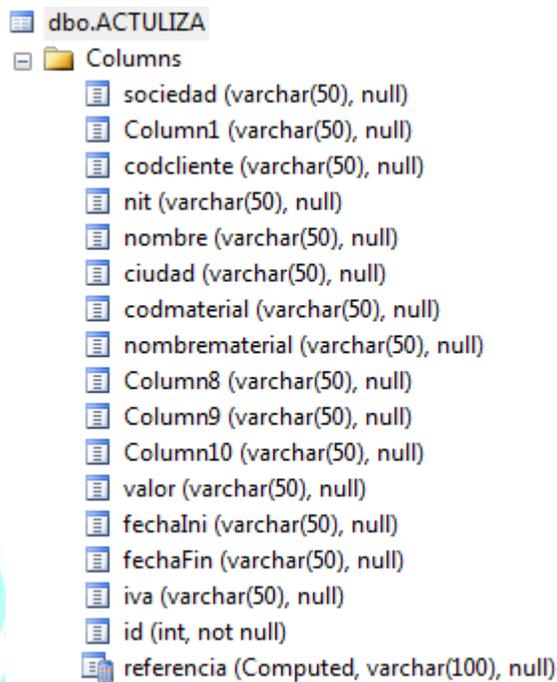


Imagen 5.1, tabla actualiza

- **ADICIONALES:** en esta tabla se cargar la categoría de los materiales adiciones que son opciones en las remisiones, la estructura de la tabla se ve en la imagen 5.2.

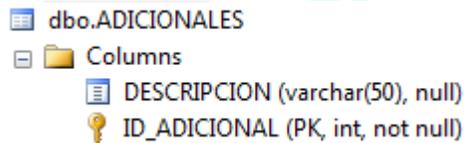


Imagen 5.2, tabla adicionales

- **ADICIONALESREF:** en esta tabla se cargar los materiales adiciones que son opciones en las remisiones, la estructura de la tabla se ve en la imagen 5.3.

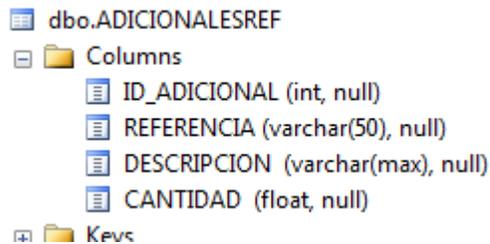


Imagen 5.3, tabla adicionalesref

- CARGUE: en esta tabla se carga el archivo plano .txt que viene de la interface entre SAP y remisiones, la estructura de la tabla se ve en la imagen 5.4.

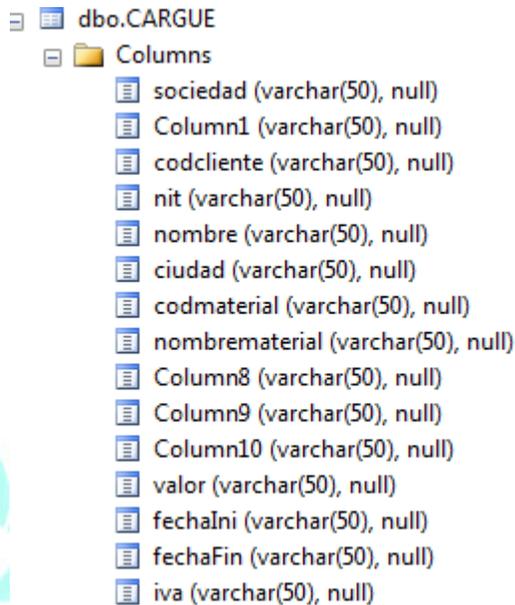


Imagen 5.4, tabla cargue

- CLIENTES: en esta tabla se encuentra toda la información relacionada a los clientes a que tienen algún tipo de relación comercial con la empresa y a los cuales se les puede generar una remisión, la estructura de la tabla se ve en la imagen 5.5.

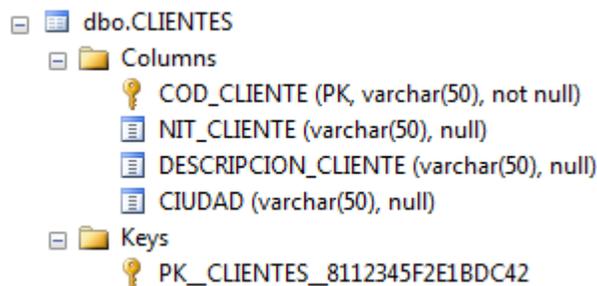


Imagen 5.5, tabla clientes

- DESTINATARIOS: esta tabla contiene los destinatarios, es decir las posibles sucursales a los cuales se puede despachar la remisión, estos están relacionados al cliente principal de la tabla cliente, la estructura de la tabla se ve en la imagen 5.6.

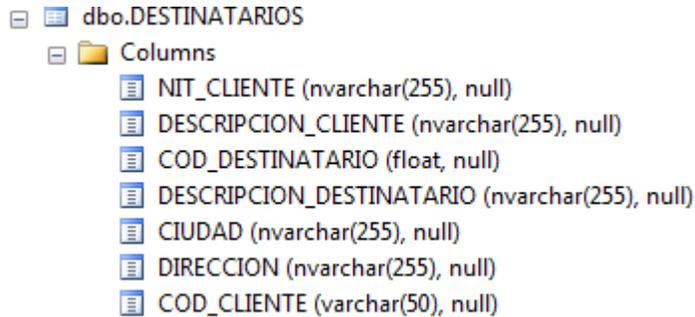


Imagen 5.6, tabla destinatarios

- HISTORICO: en esta tabla se guarda e historial de las remisiones generadas y la ruta del pdf generado, la estructura de la tabla se ve en la imagen 5.7.

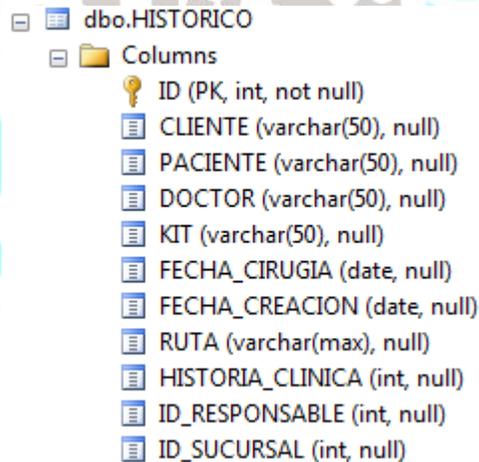


Imagen 5.7, tabla histórico

- INSTRUMENTAL: en esta tabla se guarda la información de los instrumentales relacionados a los clientes para los procedimientos que son enviados en las remisiones, la estructura de la tabla se ve en la imagen 5.8.

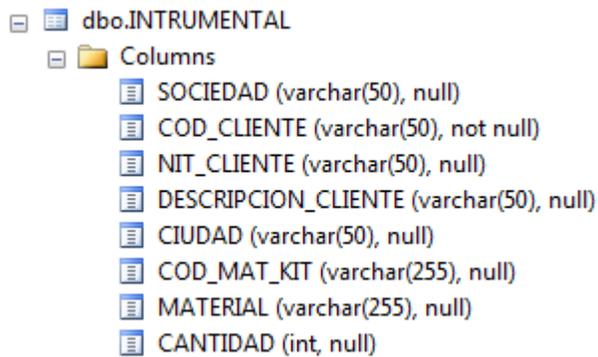


Imagen 5.8, tabla instrumental

- KIT: en esta tabla se encuentra la descripción de los kits, la estructura de la tabla se ve en la imagen 5.9.

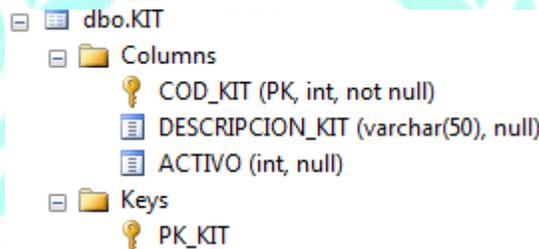


Imagen 4.9, tabla Kit

- KITS: en esta tabla se encuentra la relación de materiales por cada kit, la estructura de la tabla se ve en la imagen 5.10.

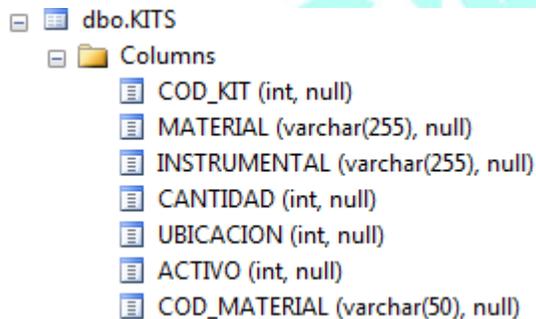


Imagen 5.10, tabla Kits

- MATERIAL: en esta tabla contiene todos los materiales que actualmente se encuentran en inventario de la compañía y es alimentada por los triggers de la tabla actualiza, la estructura de la tabla se ve en la imagen 5.11.

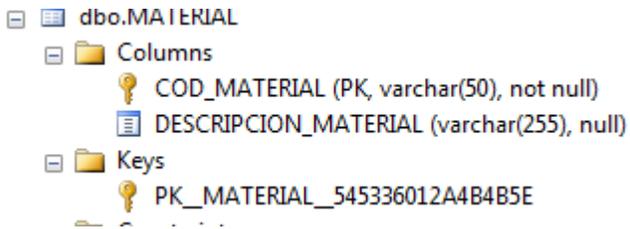


Imagen 5.11, tabla material.

- PRECIO_INTRU: en esta tabla se cargan los instrumentales, los culés por medio de un trigger se actualizan o insertan en la tabla instrumentos, la estructura de la tabla se ve en la imagen 5.12.

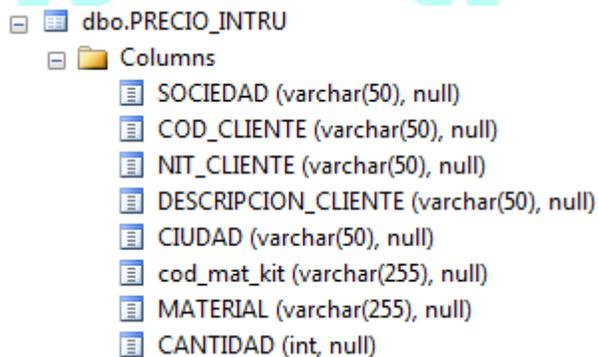


Imagen 5.12, tabla precio_intru.

- PRECIOSAP: en esta tabla se encuentra la información de precio por cliente por material, ya que la empresa maneja valores diferentes con sus clientes según las negociaciones que se tengan con ellos, esta información se actualiza por medio de la tabla actualiza, la estructura de la tabla se ve en la imagen 5.13.

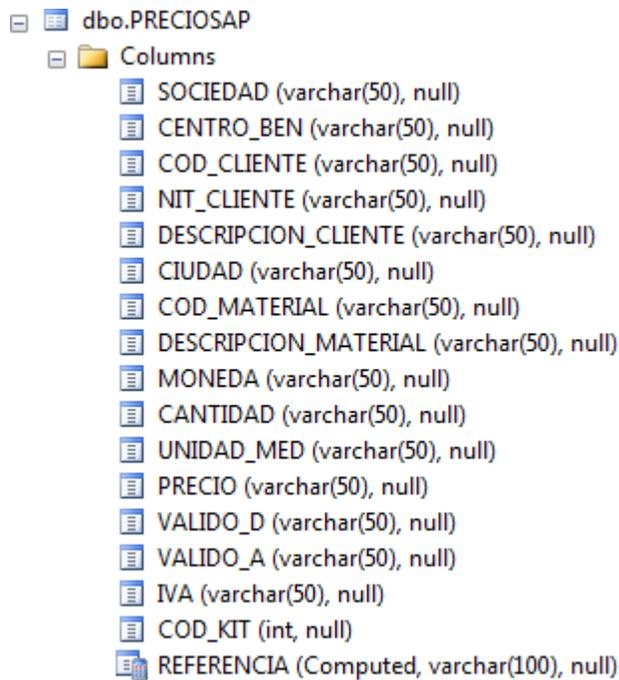


Imagen 5.13, tabla preciosap.

- **PROFILES:** en esta tabla se encuentra los perfiles que se manejan en la base de datos que dan el acceso a los menús de la aplicación de remisiones, la estructura de la tabla se ve en la imagen 5.14



Imagen 5.14, tabla profiles.

- **USERS:** esta tabla contiene a todos los usuarios de la aplicación con sus datos básicos, está relacionada con la tabla de Profiles, la estructura de la tabla se ve en la imagen 5.15

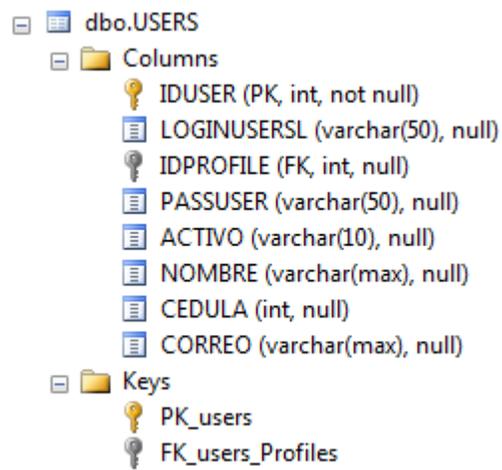


Imagen 5.15, tabla users.



ANEXO I CARTA DE CESION DE DERECHOS

Bogotá 26 de noviembre de 2015

Señores:
FUNDACION UNIVERSITARIA LOS LIBERTADORES
Facultad de Ingeniería de sistemas
Programa de Ingeniería de Sistemas
Ciudad.

Autorizo a los usuarios interesados para consultar solo con fines académicos, el contenido del trabajo de grado titulado:

Sistema de Información para el Control de las Remisiones de la empresa Grupo Amarey

Elaborado y presentado por el estudiante Ángel Andrés Pulido Suarez como requisito para optar al título de Ingeniero de sistemas.

Siempre y cuando se haga alusión de alguna parte o nota del trabajo se debe tener en cuenta la correspondiente citación bibliográfica para darle crédito al trabajo y su autor.

El autor manifiesta que la obra objeto de la presente autorización es original y la realizo sin violar o usurpar derechos de autor de terceros, por lo tanto la obra es de exclusiva autoría y tiene la titularidad sobre la misma, PARAGRAFO: en caso de presentarse cualquier reclamación por parte de un tercero en cuanto a los derechos de autor sobre la obra en cuestión, el autor, asumirá toda las responsabilidad, y saldrá en defensa de los derechos aquí autorizados, para todos los efectos la universidad actúa como tercero de buena fe, EL AUTOR, autoriza a la Fundación Universitaria los Libertadores, para que los términos establecidos en la ley 23 de 1982, la Ley 44 de 1993, Decisión andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia, utilice y use la obra objeto de la presentación.

Nombre completos	Documento de Identidad	Firma

DEDICO ESTE LOGRO:

A Dios que me bendijo con el don de la constancia y la perseverancia, además por poner en mi camino la ingeniería de sistema, porque a través de ella puedo explotar las habilidades y destrezas que él me brinda.

A mi Esposa Ederlet Santacruz por tanto amor y apoyo incondicional que me ha dado, a ella le dedico este triunfo con todo el amor del mundo.

A mi abuela Rosana Medina, a ella le debo todo lo que soy.

A mi Padre Miguel Ángel Pulido un gran hombre que fue testigo de tantos esfuerzos y sacrificios.



AGRADECIMIENTOS

Me place agradecer:

Al cuerpo de docentes y a la Fundación universitaria los Libertadores, por compartir sus conocimientos y brindarnos una educación de alta calidad.

