



Universidad de las Ciencias
Informáticas

Universidad de las Ciencias Informáticas
Facultad 3

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Herramienta de gestión de información toxicológica para el CENATOX

Autor:

Susel Hernández Trejo

Tutores:

Ing. Víctor Alexis Mesa Cisneros

Dr.C. Arturo Orellana García.

La Habana, septiembrede 2020

Declaración de autoría

Declaro ser el único autor del trabajo de diploma "Herramienta de gestión de información toxicológica para el CENATOX", concedo a la Universidad de las Ciencias Informáticas y en especial al Centro de Informática Médica la autorización a hacer uso del mismo en su beneficio.

Para que conste firmo el presente documento a los ____ días del mes de septiembre del año 2020.

Susel Hernández Trejo

Firma del autor

Dr.C. Arturo Orellana García

Ing. Víctor Alexis Mesa Cisneros

Firma del tutor

Firma del tutor

Datos de contacto

Susel Hernández Trejo (strejo@estudiantes.uci.cu)

Dr. C. Arturo Orellana García (aorellana@uci.cu): graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2012. Se desempeña como líder del Grupo de Investigación de Minería de procesos y Asesor de Capacitación, Desarrollo e Investigación del Centro de Soluciones de Informática Médica. Ha liderado proyectos I+D+i de desarrollo de componentes de software a partir de minería de procesos para el análisis de procesos de negocio del entorno hospitalario. Investiga la Ingeniería de comportamiento, la medicina de precisión y el procesamiento de imágenes médicas. Tutora varias tesis de grado, maestrías y doctorados enfocados al análisis de procesos de negocio, la informática médica y otras áreas del conocimiento. Obtuvo el grado de Máster en Informática Aplicada en 2015 desarrollando una herramienta informática basada en técnicas de minería de procesos para identificar problemas en la ejecución de procesos de negocio. Doctor en Ciencias Técnicas desde 2016 presentando un modelo computacional para la detección de variabilidad en procesos de negocio del entorno sanitario aplicando minería de procesos.

Ing. Víctor Alexis Mesa Cisneros (vamesa@uci.cu)

Dedicatoria

A mi familia por apoyarme y guiarme siempre. A mis amigos por todo el apoyo que me ofrecieron. A mi tutor por siempre estar presente y alentarme. A todo aquel que alguna vez me brindó un consejo.

Resumen

La necesidad de automatización de procesos de negocio actualmente es un reto al que son enfrentados los centros médicos. En el Centro Nacional de Toxicología CENATOX los especialistas requieren tener una facilidad de revisión de la documentación necesaria para el análisis y la emisión de un diagnóstico que auxilie la vida del afectado. La presente investigación tiene como objetivo desarrollar una herramienta que permita la gestión de información toxicológica en dicho centro. Para darle solución al problema se realizó un análisis de las herramientas existentes que permiten la gestión de información toxicológica. Para desarrollar la solución se utilizó el lenguaje PHP y el framework Symfony. Se obtuvo como resultado una herramienta de gestión que realiza la gestión de información toxicológica necesitada. Esta solución propició que los especialistas tuvieran un mecanismo para agilizar su trabajo, haciendo posible que la duración del tiempo de ayuda toxicológica disminuyera considerablemente y a su vez la ocurrencia de errores humanos.

Palabras clave: gestión de información, herramienta de gestión, toxicológica.

Abstract

The need for automation of business processes is currently a challenge faced by medical centers. At the CENATOX National Toxicology Center, specialists need to have a facility to review the documentation necessary for the analysis and the issuance of a diagnosis that helps the life of the affected person. The present research aims to develop a tool that allows the management of toxicological information in said center. To solve the problem, an analysis of the existing tools that allow the management of toxicological information was carried out. To develop the solution, the PHP language and the Symfony framework were used. The result was a management tool that manages the needed toxicological information. This solution allowed specialists to have a mechanism to streamline their work, making it possible for the duration of the time of toxicological help to decrease considerably and in turn the occurrence of human errors.

Keywords: information management, management tool, toxicology.

Índice

Introducción	1
Capítulo 1. Fundamentación teórica de la investigación	6
1.1 La Toxicología en el mundo	6
1.2 El proceso de toxicología e información toxicológica en Cuba	9
1.2.1 Análisis de información toxicológica en Cuba.	11
1.3 Análisis del estado del arte	14
1.4 Comparación de los sistemas.....	16
1.5 Metodología de desarrollo.....	18
1.5.1 Metodología Programación Extrema (XP)	19
1.6 Herramientas para el modelado y representación de procesos de negocio	22
1.6.1 Herramientas	23
1.6.2 Lenguaje	24
1.6.3 Sistema gestor de base de datos.....	25
1.7 Conclusiones del capítulo.....	26
Capítulo 2. Propuesta de solución	27
2.1 Proceso de negocio a informatizar	27
2.2 Análisis de la solución.....	28
2.3 Fase de Planificación.....	32
2.3.1 Historias de Usuarios	32
2.3.2 Estimación del esfuerzo por HU	34
2.3.3 Plan de iteraciones	35
2.3.4 Plan de entregas	36
2.4 Fase de diseño	36
2.4.1 Arquitectura de Software.....	36
2.4.2 Tarjetas Clase-Responsabilidad-Colaboración.....	38
2.4.3 Patrones de Diseño.....	39
2.5 Fase de Implementación.....	41
2.5.1 Tareas de ingeniería	41
2.5.2 Estándares de codificación	42
2.6 Conclusiones del capítulo	43
Capítulo 3. Validación de la propuesta de solución	45
3.1 Fase de pruebas.....	45

3.2 Niveles de prueba	45
3.2.1 Pruebas de aceptación	46
3.2.2 Pruebas unitarias	47
3.3 Conclusiones del capítulo	50
Conclusiones	51
Recomendaciones	52
Bibliografía	53
Anexos	56

Índice de Figuras

Figura 1 Distribución de las intoxicaciones agudas según años y tipo de intoxicación. Fuente: (Dra Rodríguez Rubinos, et al., 2007).....	13
Figura 2 Proceso XP. Fuente: (Pressman, 2007)	21
Figura 3 Arquitectura MVC. Fuente: (Elaboración propia)	38
Figura 4 Pruebas de aceptación. Fuente: (Elaboración propia)	49
Figura 5 Pruebas unitarias. Fuente: (Elaboración propia)	50
Figura 6 Sistema de tablas de la herramienta. Fuente: (Elaboración propia)	56

Índice de tablas

Tabla 1 Fallecidos y tasas ajustadas por intoxicaciones agudas producidas con medicamentos. Fuente: (Mortalidad por intoxicaciones agudas producidas con medicamentos. Cuba, 1995-1996, 2000)	11
Tabla 2 Distribución de las intoxicaciones agudas según agente causal y tipo de intoxicación. Fuente: (Dra Rodríguez Rubinos, et al., 2007)	13
Tabla 3 Análisis de los sistemas propuestos. Fuente: (Elaboración propia)	16
Tabla 4 Personas que intervienen en el negocio. Fuente: (Elaboración propia)	28
Tabla 5 Descripción de los requisitos funcionales. Fuente: (Elaboración propia)	29
Tabla 6 Descripción de la HU 4 Consultar Ficha Manual. Fuente: (Elaboración propia).....	33
Tabla 7 Descripción de la HU 5 Visualizar Ficha Manual. Fuente: (Elaboración propia).....	34
Tabla 8 Estimación del esfuerzo por HU. Fuente: (Elaboración propia)	34
Tabla 9 Plan de duración de las iteraciones. Fuente: (Elaboración propia)	35
Tabla 10 Plan de Entrega. Fuente: (Elaboración propia).....	36
Tabla 11 Tarjeta CRC clase Datos. Fuente: (Elaboración propia)	39
Tabla 12 Tarjeta CRC clase PrincipalController. Fuente: (Elaboración propia)	39
Tabla 13 Distribución de tareas de Ingeniería por HU. Fuente: (Elaboración propia)	41
Tabla 14 Tarea de ingeniería Consultar Ficha Manual correspondiente a la búsqueda. Fuente: (Elaboración propia).....	42
Tabla 15 Tarea de ingeniería Visualizar Ficha Manual consultada. Fuente: (Elaboración propia)	42
Tabla 16 Caso de prueba Autenticar usuario. Fuente: (Elaboración propia).....	46
Tabla 17 Caso de prueba Consultar Ficha Manual. Fuente: (Elaboración propia).....	47
Tabla 18 Descripción de la HU 1 Gestionar usuario. Fuente: (Elaboración propia)	56

Tabla 19 Descripción de la HU 2 Autenticar usuario. Fuente: (Elaboración propia).....	56
Tabla 20 Descripción de la HU 3 Autocompletar consulta. Fuente: (Elaboración propia)	56
Tabla 21 Descripción de la HU 6 Exportar a PDF. Fuente: (Elaboración propia).....	57

Introducción

La palabra Toxicología, procede etimológicamente del término toxicoz, que significa propio para arco o flechas. En realidad, se refiere a la presencia de una sustancia nociva con la que se impregnaban las puntas de lanzas y flechas del Paleolítico con la finalidad de aumentar su capacidad mortífera. (La evolución de la Toxicología: de los venenos a la evaluación de riesgos, 2005) (Origen e historia de la Toxicología, 2014)

La historia de la Toxicología es muy antigua, desde los comienzos de la humanidad ante la necesidad de alimentarse el ser humano sintió la obligación de consumir los productos que encontraba a su alcance, con ello obtuvo la experiencia de que muchos resultaban perjudiciales para su salud, de manera que conoció los envenenamientos por sus efectos mortales. (La evolución de la Toxicología: de los venenos a la evaluación de riesgos, 2005)

Se puede afirmar que la Toxicología fue en sus inicios y durante un largo lapso temporal el estudio de los venenos, documentados por vez primera en el Papiro Ebers, datado hacia el 1500 antes de nuestra era y perteneciente al desarrollo de la compleja y fascinante XV^{LL}ª dinastía de Egipto, además se relata el poder de venenos para remedios y la fumigación como tratamiento. (La evolución de la Toxicología: de los venenos a la evaluación de riesgos, 2005)

En los siglos XIX y XX se producen los primeros intentos de crear modelos matemáticos de las relaciones entre dosis y respuesta, aparecen conceptos como 'dosis tolerada', 'concentración máxima permitida' o 'nivel de seguridad' y la experimentación animal para conocer cómo se desarrolla la respuesta del organismo. Fue entonces cuando se empezó a discutir la relación entre la frecuencia, la intensidad y la duración de una exposición, además del desarrollo de las interacciones químicas, intentando conseguir teorías tan pronto explicativas como predictivas. Dando paso al estudio de la Toxicología como una ciencia independiente en la era moderna. (La evolución de la Toxicología: de los venenos a la evaluación de riesgos, 2005)

Como toda disciplina científica, la Toxicología continúa en evolución, sufriendo transformaciones que determinan su objetivo y forma. Aparecen nuevas especialidades como la Biología Molecular y la Bioinformática, uno de los estudios específicos de esta última aborda la toxicidad bajo entornos virtuales computarizados que permiten un ahorro de tiempo y recursos económicos, lo que impulsa esta nueva rama de investigación toxicológica y crea la esperanza firme en el desarrollo de una toxicología predictiva. A la misma vez, la potente capacidad de computación y calidad en la

programación de nuestros días está dando lugar a una toxicología computacional que permite estudios de gran calidad y excelente capacidad predictiva; además la existencia de bases de datos en red está dando lugar a una dinámica de trabajo mucho más rápida y precisa. (La evolución de la Toxicología: de los venenos a la evaluación de riesgos, 2005)

Se destacan a escala mundial en la Información Toxicológica, el Programa Internacional de Seguridad de las Sustancias Químicas (IPCS), el paquete computarizado sobre intoxicaciones (INTOX/IPCS), el proyecto denominado INTOX, a nivel de región, la Biblioteca Virtual en Salud y Ambiente – Toxicología (BVSA-TOXICOLOGIA), todos estos con el objetivo de reforzar la capacidad de prevenir los accidentes, tratar los posibles afectados, vigilar y reducir las intoxicaciones como causa de enfermedad o muerte. (La Biblioteca Virtual de Salud-Toxicología, 1999) (1998)

Existen diversidad de productos tóxicos y circunstancias en que se pueden presentar intoxicaciones, el conocimiento de sus mecanismos de acción, metabolismo y neutralización, provoca que se haga más complejo el trato de casos intoxicados. Por estas razones se desarrollan los Centros de lucha contra las intoxicaciones o Centros antitóxicos (CAT), que son responsables de reducir la morbilidad y mortalidad asociada con las intoxicaciones. Estos brindan información toxicológica de urgencia y asesoramiento para el tratamiento por vía telefónica. (Pérez Barly, 2013)

El Centro Nacional de Toxicología CENATOX, creado en 1986 en La Habana, es el coordinador del sistema de tóxico-vigilancia en Cuba. Posee una red integrada por tres centros regionales localizados al oriente y centro de la isla. (2020) El CENATOX es la instancia de control y supervisión donde se gestiona la información sobre reacciones adversas a drogas de abuso, agroquímicos, animales tóxicos, fármacos, plantas tóxicas, productos del hogar y productos industriales.

Este centro tiene como visión lograr el fortalecimiento en la toma de decisiones cuando el personal médico y o la población se enfrentan a productos químicos de cualquier índole y a su vez retroalimentarlos, a partir de la base, en relación al comportamiento de los eventos toxicológicos, todo esto con el objetivo del mejoramiento de la calidad de vida de la población. (2020)

A diario aparecen en el mercado nuevos de estos reactores tóxicos y los profesionales de la salud, generalmente, desconocen sus peligros y mecanismos; por ello necesitan un asesoramiento para el tratamiento del paciente intoxicado, que resulte verídico, conciso y sobre todo rápido.

El proceso de ayuda actualmente se realiza en línea las 24 horas, todo el año, recibiendo interconsulta telefónica de todas las unidades de salud del país, brindando al personal consultante sugerencias de conductas terapéuticas a seguir, mediante fichas impresas, libros de texto, folletos, registro de

plaguicidas, entre otros; en formato impreso y digital. Un gran porcentaje de esta información data desde la fundación del centro. Este proceso resulta tedioso por el gran cúmulo de información en formato duro que conlleva y la propia ejecución de él de forma manual, desde que se recibe la llamada de urgencia hasta que se presenta un diagnóstico; esto provoca que se puedan cometer errores en la gestión de los datos por parte del personal médico y como consecuencia la pérdida de vidas humanas. Analizar esta gran cantidad de documentos para llegar a un posible diagnóstico hace que el proceso pueda tardar alrededor de 10 horas en la mayoría de los casos. (Dr. C. My. Fleites Mestres, 2019) A raíz de que el centro no posee un sistema automatizado y no cuenta con los existentes en el mundo debido a sus altos costos y al bloqueo, se hace necesario informatizar este proceso para así aportar mayor calidad y rapidez en el servicio.

A partir de la situación problemática descrita anteriormente se identifica como **problema a resolver**: ¿Cómo mejorar la recolección y gestión de los datos de los casos de intoxicaciones por emergencias en el CENATOX?

El problema se enmarca en el **objeto de estudio**: proceso de análisis de información epidemiológica en Cuba, que se enfoca en el **campo de acción**: análisis de la información de intoxicaciones en el CENATOX.

Para darle solución al problema planteado, se define como **objetivo general**: desarrollar una herramienta de gestión de información toxicológica, para generar información oportuna y rápida en la prevención y control de eventos en el CENATOX.

Tareas de investigación:

1. Elaboración del marco teórico metodológico referente a la toxicología derivada al uso de drogas de abuso, agroquímicos, animales tóxicos, fármacos, plantas tóxicas, productos del hogar y productos industriales en Cuba.
2. Análisis de las características, particularidades y procesos de negocio del Centro Nacional de Toxicología de Cuba.
3. Desarrollo de las funcionalidades identificadas para el sistema.
4. Implantación del sistema en el Centro Nacional de Toxicología de Cuba.
5. Validación de los resultados obtenidos a partir de los métodos definidos en la investigación.

Los **métodos de investigación** utilizados para darle solución al objetivo propuesto fueron:

Teóricos:

- **Analítico-sintético:** se utiliza en la fundamentación del objeto de estudio de la investigación a partir de los conceptos asociados y de la comprensión del negocio.
- **Inductivo-deductivo:** se utiliza en la aplicación de casos de pruebas al sistema, llegando a conclusiones a partir de las respuestas proporcionadas por este.
- **Histórico-lógico:** se utiliza para realizar un análisis crítico valorativo de la información contenida en las fuentes bibliográficas consultadas y para investigar acerca de otras soluciones similares como: PlaguitoX, Sistema Automatizado de Información Toxicológica de Urgencia en el CENATOX, entre otros.

Empíricos:

- **Observación:** se utiliza para observar la forma en que se realiza el proceso de recolección y gestión de los casos de intoxicaciones por drogas de abuso, agroquímicos, animales tóxicos, fármacos, plantas tóxicas, productos del hogar y productos industriales en el CENATOX. Brinda una comparación y un análisis de los resultados obtenidos en las pruebas realizadas a la aplicación desarrollada.
- **Entrevista:** se utiliza para la recopilación de información a través de conversaciones con el cliente, para de esta forma satisfacer sus necesidades y obtener un producto con la calidad requerida.
- **Análisis documental:** se utiliza para reconocer los elementos de las hojas de drogas de abuso, agroquímicos, animales tóxicos, fármacos, plantas tóxicas, productos del hogar y productos industriales que forman parte de la propuesta y definir así los criterios necesarios en la emisión de diagnósticos.
- **Análisis comparativo:** para detectar similitudes, diferencias e insuficiencias en cuanto a la interpretación que brindan las herramientas existentes para gestión y visualización de los procesos del negocio.

La estructura del documento se caracteriza por la presencia de: introducción, tres capítulos, conclusiones, bibliografías y los anexos. El contenido de cada capítulo es descrito a continuación:

Capítulo 1. Fundamentación teórica de la investigación: aborda los conceptos principales sobre los diferentes elementos en la que se basa esta investigación, tiene un análisis del estado del arte, a nivel nacional e internacional de sistemas que permitan otorgar información idónea y rápida para evitar y controlar eventos y se realiza una explicación de las herramientas y tecnologías utilizadas para el desarrollo del sistema propuesto.

El **Capítulo 2. Propuesta de solución:** reseña la propuesta de solución, además de la modelación del flujo de información de la herramienta desarrollada, se describen los patrones arquitectónicos y los de diseños empleados; se realiza una explicación de la técnica empleada y del procedimiento para el desarrollo del sistema.

Finalizando en el **Capítulo 3. Validación de la propuesta de solución:** contempla la evaluación de los resultados alcanzados a lo largo de la investigación y son descritas las validaciones realizadas a la herramienta para medir su adecuado rendimiento.

Capítulo 1. Fundamentación teórica de la investigación

Se analizan los elementos de la base teórico-conceptual de la investigación para darle desarrollo a la solución propuesta, apoyándose en el análisis y valoración de forma crítica, de las tendencias, tecnologías actuales y antecedentes que se asocian al campo de acción. Se escoge la metodología de desarrollo que guiará el proceso de desarrollo del software, además se realiza una caracterización de las soluciones distintas existentes que se corresponden con el objeto de estudio de la investigación. Se describen y justifican las características principales de las herramientas y tecnologías de software empleadas para el desarrollo del sistema.

1.1 La Toxicología en el mundo

Conceptos Principales

Toxicología: Es la ciencia que estudia las sustancias químicas y los agentes físicos, en cuanto son capaces de producir alteraciones patológicas a los seres vivos, a la par que estudia los mecanismos de producción de tales alteraciones y los medios para detectar, identificar y determinar tales agentes y valorar su grado de toxicidad. (Suárez Escandón, 2007)

Tecnologías de Información y las Comunicaciones: Las TIC pueden ser tradicionales, como la radio, la televisión y los medios impresos; o nuevas (NTIC), un conjunto de medios y herramientas como los satélites, la computadora, la Internet, el correo electrónico, los celulares, los robots entre otros.(Nuevas tecnologías de la información: problemas éticos fundamentales, 2005)

Drogas de abuso: Aún y cuando existen definiciones de toda índole y en todos los idiomas, la formulada por Gisbert Calabuig y recogida en el Manual de drogodependencia resulta concreta y futurista, a saber: "Drogas son todas aquellas sustancias que producen dependencia y que se emplean voluntariamente para provocarse determinadas sensaciones o estados psíquicos no justificados terapéuticamente".(Consideraciones generales sobre drogas de abuso, 2002)

Agroquímicos: Estos productos también conocidos como fitosanitarios o plaguicidas son insumos que previenen, repelen o controlan cualquier plaga de origen animal o vegetal durante la producción, almacenamiento, transporte y distribución de productos agrícolas.(2019)

En los últimos 100 años la Toxicología se ha desarrollado rápidamente, pero su crecimiento exponencial data de la era de la posguerra (después de la II Guerra Mundial), cuando se dispara la producción de moléculas orgánicas tales como drogas, pesticidas y sustancias químicas de uso industrial y militar. Hacia 1955, el toxicólogo estadounidense Arnold Lehman escribió otro adagio que haría historia en la Toxicología: *Usted también puede ser un toxicólogo en dos sencillas lecciones, cada una de 10 años.* La primera revista científica sobre Toxicología, *Toxicology and Applied Pharmacology*, fue creada por Coulston, Lehman y Hayes, en Estados Unidos. (Origen e historia de la Toxicología, 2014)

En América Latina cabe destacar a Emilio Astolfi (1930-1995), médico argentino, que puede considerarse el padre de la Toxicología en este continente. Se formó en la escuela francesa y fue el pionero, fundador y formador de la primera cátedra de Toxicología en la Facultad de Medicina de la Universidad de Buenos Aires (Argentina), donde actualmente se ofrecen cursos a nivel de pregrado y posgrado de especialización y maestrías desde hace muchos años. Waldemar Almeida, médico brasileño, pionero de la Toxicología en Brasil, fallecido en 1996, hizo grandes aportes en el campo de la investigación y creó la cátedra de Toxicología y el Centro de Estudios Toxicológicos en el Instituto Oswaldo Cruz, de Río de Janeiro. (Origen e historia de la Toxicología, 2014)

En la era moderna es evidenciado un incremento en la disponibilidad de sustancias químicas, vegetales, animales, que ponen en constante riesgo para la salud a la población mundial. Las intoxicaciones en la población en general son frecuentes, y en el área pediátrica lo son aún más, sobre todo en las edades de 1 a 5 años, esto es debido a la curiosidad que presenta el niño y el descuido del adulto al dejar agentes químicos, fármacos de uso habitual al alcance de los niños, en la mayoría siendo accidental. En los adolescentes son debido a intentos suicidas y en los adultos por el manejo inapropiado al manipular las sustancias sin la debida precaución. Todo lo antes abordado ha influenciado la necesidad de crear los Centros de Información Toxicológica en diversas partes del mundo. (Jiménez Rangel, 2006)

La diversidad de productos tóxicos y las circunstancias en que se pueden presentar las intoxicaciones, así como el conocimiento de sus mecanismos de acción, metabolismo y neutralización, hacen más

complejo el manejo de los casos intoxicados. Por estas razones se desarrollan los Centros de lucha contra las intoxicaciones o Centros Antitóxicos (CAT), que se inician por vez primera en Chicago, Estados Unidos, en el año 1952, desarrollándose paulatinamente en otros países europeos y luego en Latinoamérica. (Suárez Escandón, 2007)

Estos centros son los responsables de reducir la morbilidad y mortalidad asociada con las intoxicaciones, al brindar información toxicológica de urgencia y asesoramiento para el tratamiento por vía telefónica. Los mismos cuentan al menos, con ficheros impresos en papel, una biblioteca básica y una computadora, que facilita el acceso a la información toxicológica vía internet o mediante discos compactos. (Suárez Escandón, 2007)(La gestión de información como herramienta fundamental en el desarrollo de los centros tecnológicos, 2003)(1998)

Las Tecnologías de la Información y las Comunicaciones (TIC) han impactado en distintas áreas de la sociedad, facilitando la informatización de diversos sectores como el de la salud. El uso de las TIC en los sistemas de salud, está destinado a mejorar la efectividad y eficiencia del sector, tanto dirigido a una mejor atención de los pacientes como a la toma de decisiones de la administración y la dirección. (Informatización en el Sistema Nacional de Salud. Enfoques hacia la dirección en salud, 2013)

A escala mundial, se destacan grandes esfuerzos de Información Toxicológica, mediante el uso de las TIC:

IPCS: Es un Programa Internacional de Seguridad de las Sustancias Químicas (IPCS), fundado en 1980, fruto de la acción conjunta de tres organizaciones internacionales interesadas en la seguridad química de todos los países del mundo. Ellas son: Organización Internacional del Trabajo (OIT), PNUMA y la Organización Mundial de la Salud (OMS).

INTOX/IPCS: Es un proyecto del IPCS, que ha desarrollado un paquete computarizado sobre intoxicaciones (INTOX/IPCS), disponible en español, francés, inglés y portugués.

Existe además otro proyecto a escala mundial, denominado **INTOX**, desarrollado por el IPCS/OMS

A nivel de la región se ha creado La Biblioteca Virtual en Salud y Ambiente - Toxicología (**BVSA-TOXICOLOGIA**), que se desarrolla por una acción conjunta

del CEPIS/OPS y el Sistema Latinoamericano y del Caribe de Información en Ciencias de la Salud (BIREME).

1.2 El proceso de toxicología e información toxicológica en Cuba

Cuba posee referencia de manifestaciones tóxicas en el ser humano, ocurridas estas en el siglo xv entre los aborígenes, ocasionadas por la inhalación de gran cantidad de tabaco, lo cual producía embriaguez y alucinaciones. (Origen e historia de la Toxicología, 2014)

La bibliografía médico-legal-toxicológica comienza en la Habana desde 1839, con las publicaciones y trabajos realizados por reconocidos médicos de la época como: doctor José de Lletor y Castroverde, profesor de Medicina Legal; doctor Ángel J. Cowley, profesor de Methodus Mendendi (Terapéutica); doctor Carlos Donoso Lardier, profesor de la Facultad de Farmacia; doctor Ramón Zambrana Valdés, considerado el precursor de la Medicina Legal en Cuba, entre otros. (Origen e historia de la Toxicología, 2014)

A pesar de contar en Cuba con una Universidad desde 1728, no fue hasta 1842 que se impartió la Medicina Legal y Toxicología como asignatura de una de las cátedras de la Facultad de Medicina de la Universidad de la Habana, la cual evolucionó con el paso de los años. (Fraga Álvarez, et al., 1996)

Uno de los pioneros de estas ciencias fue el doctor cubano Ramón Zambrano, primer especialista de Toxicología del siglo XIX en Cuba. A este le seguirían en el siglo XX médicos y profesores como Raimundo Fernández y Francisco Lancís, este último una personalidad mundialmente conocida de la Medicina Forense y formador de una escuela nacional que alcanzó un merecido reconocimiento a nivel internacional. (Pérez Pérez, et al., 2010)

Sin embargo, la Toxicología Médica alcanzó la mayoría de edad en el país en las postrimerías del siglo XX, cuando logró establecerse como una entidad independiente al ser creado el Centro Nacional de Toxicología en el Hospital Militar Central "Dr. Carlos J. Finlay". (Pérez Pérez, et al., 2010)

El 15 de noviembre de 1986, se fundó en Cuba el Centro Nacional de Toxicología (CENATOX). Entre sus principales funciones incluyen: brindar información toxicológica y asesoramiento en el manejo de los pacientes intoxicados, proporcionar servicio de laboratorio analítico, toxicovigilancia, desarrollo de estudios pre-clínicos y clínicos, docencia y entrenamiento en el campo de la prevención y el manejo del paciente intoxicado. (Pérez Barly, 2013)

El CENATOX como centro de control de intoxicaciones cumplía los requisitos establecidos por la OMS y el IPCS, nombrándolo representante de la actividad por Cuba ante estos organismos, convirtiéndose en uno de los centros más completos de su tipo en América Latina y el Caribe, condición reconocida en eventos internacionales. (Moya Díaz, et al., 2003)

El CENATOX es miembro de la Asociación Latinoamericana de Toxicología (ALATOX) y preside la Sociedad Cubana de Toxicología. Ha auspiciado diferentes eventos nacionales e internacionales entre los que se encuentran: Congreso Internacional de Toxicología del 88, X Congreso Latinoamericano de Toxicología de 1998, Primer Curso de Fortalecimiento del Liderazgo de Centros de Toxicología en Centroamérica y IV Congreso Internacional de Toxicología de 2007. (Dra García Rosales, 2020)

Algunas de las misiones que busca cumplir el CENATOX son (Dra García Rosales, 2020):

- Organizar y asesorar la asistencia médica toxicológica en el país en los niveles de atención de salud primario, secundario y terciario.
- Actuar en conjunto con el Estado Mayor Nacional de la Defensa Civil en la prevención y enfrentamiento de las emergencias o catástrofes químicas.
- Asesorar metodológicamente la preparación y funcionamiento de las Brigadas Toxicológicas del país en caso de accidentes químicos o intoxicaciones masivas.
- Mantener bajo vigilancia la incidencia de intoxicaciones agudas en el país con el objetivo de eliminar o disminuir las posibles causas que generen intoxicaciones masivas y desastres tecnológicos de objetivos químicos.
- Brindar información toxicológica de urgencia por vía telefónica al Sistema Nacional de Salud, instituciones del estado, población, etc.

Cuba cuenta con tres Centros de Información, Asesoramiento y Asistencia Toxicológica, que cuentan con personal especializado para asesorar sobre el tratamiento y la prevención de las intoxicaciones y facilitar así información sobredrogas de abuso, agroquímicos, animales tóxicos, fármacos, plantas tóxicas, productos del hogar y productos industriales. Los especialistas de estos centros están capacitados para recomendar las acciones a ejecutar cuando las personas entran en contacto con algunos de estos: (2020)

- Centro Nacional de Toxicología, con el Grupo de Estadística y Toxicovigilancia como Coordinador de la actividad. La Habana.

- Centro de Toxicología CENTOX. Villa Clara.
- Centro para la Salud y el Desarrollo. Ciego de Ávila.
- Centro de Toxicología y Biomedicina. Santiago de Cuba.

1.2.1 Análisis de información toxicológica en Cuba.

Con el objetivo de una mejor comprensión del entorno toxicológico del país y la necesidad de recopilar esta información, se muestran los datos obtenidos en dos estudios publicados: "Mortalidad por intoxicaciones agudas producidas con medicamentos. Cuba, 1995-1996" que detalla estadísticas sobre la mortalidad por este tipo de intoxicaciones en Cuba realizado en el CENATOX (Mortalidad por intoxicaciones agudas producidas con medicamentos. Cuba, 1995-1996, 2000) y "Caracterización clínica epidemiológica de las intoxicaciones agudas en pediatría" que aborda estadísticas de intoxicaciones en edades tempranas en la provincia de Camagüey. (Dra Rodríguez Rubinos, et al., 2007)

Mortalidad por intoxicaciones agudas producidas con medicamentos. Cuba, 1995-1996

Con el comportamiento toxicológico entre los años 1995 y 1996 fue realizado un estudio que muestra las siguientes estadísticas: los años potencialmente perdidos (AVPP) permitieron valorar el impacto de esos decesos por su ocurrencia a temprana edad, con un cúmulo de 7430 años (a) para una tasa de 0,7 a por 1000 habitantes; es en el sexo femenino donde aparecen las mayores pérdidas con 74% de los AVPP. Las edades adultas que aportan la mayor cantidad de pérdidas son las comprendidas entre 20 y 44 a y con mayor predominio aparecen los grupos de 20-24 y de 25-29 a con 12,8 y 13,2 % respectivamente. Este mismo comportamiento se observa en las mujeres, mientras que en los hombres los grupos de más pérdidas son los de 30-34 y de 35-39 a con 13,4 y 14,9% respectivamente, y las edades de 20-29 a son las menores pérdidas. En los niños las edades que predominaron son las comprendidas hasta 4 a con 6,4% del total de AVPP y 53% del aportado por estos menores. Esto es evidenciado es la siguiente tabla:

Tabla 1 Fallecidos y tasas ajustadas por intoxicaciones agudas producidas con medicamentos. Fuente: (Mortalidad por intoxicaciones agudas producidas con medicamentos. Cuba, 1995-1996, 2000)

Grupos de edad	Sexo masculino		Sexo femenino		Totales		Razón por sexo F/M
	Fallecido	Tasa	Fallecido	Tasa	Fallecido	Tasa	

<1	1	0,6	1	0,6	2	0,6	1,0
1 a 4	3	0,4	2	0,3	5	0,4	0,7
5 a 14	2	0,2	5	0,0	7	0,2	1,5
15 a 24	5	0,3	29	1,7	34	1,0	5,7
25 a 34	10	0,4	36	1,6	46	1,0	4,0
35 a 44	16	1,8	38	2,5	54	1,8	1,4
45 a 54	16	1,3	26	2,0	42	1,6	1,5
55 a 64	13	1,5	43	4,8	56	3,1	3,2
65 a 74	6	1,0	28	4,7	34	2,9	4,7
>75	13	3,3	21	4,7	34	4,0	1,4
Totales	85	0,5	229	2,1	3,14	1,4	4,2

Caracterización clínica epidemiológica de las intoxicaciones agudas en pediatría

De el 1ro de enero de 2001 al 31 de diciembre de 2005, con el objetivo de caracterizar el comportamiento clínico epidemiológico de las intoxicaciones agudas. Fueron revisadas un total de 576 historias clínicas que cumplían los criterios de inclusión y se estudiaron las siguientes variables: año, tipo de intoxicación, edad, sexo, agente casual, complicaciones y estadía. Predominaron las intoxicaciones accidentales en varones menores de cinco años sobre las voluntarias en féminas adolescentes, con una tendencia al incremento del número de casos en los dos últimos años para casi la totalidad de los grupos etéreos, los medicamentos fueron los agentes causales más involucrados en ambos tipos de intoxicaciones y en el 89.7 % de los casos cursan sin complicaciones con una estadía promedio general de 3.3 días.

De un total de 576 pacientes egresados con diagnóstico de intoxicación aguda en el período de estudio se observó una tendencia ascendente con el paso de los años, se atendieron en el último bienio 279 casos que representa el 48.4 % del total, por otra parte, se observó que las intoxicaciones accidentales (379) casi duplicaron las voluntarias (197) que representan el 65.8 % y el 34.2% respectivamente, se destacó a partir de 2003 el incremento del número de casos con intoxicaciones voluntarias.

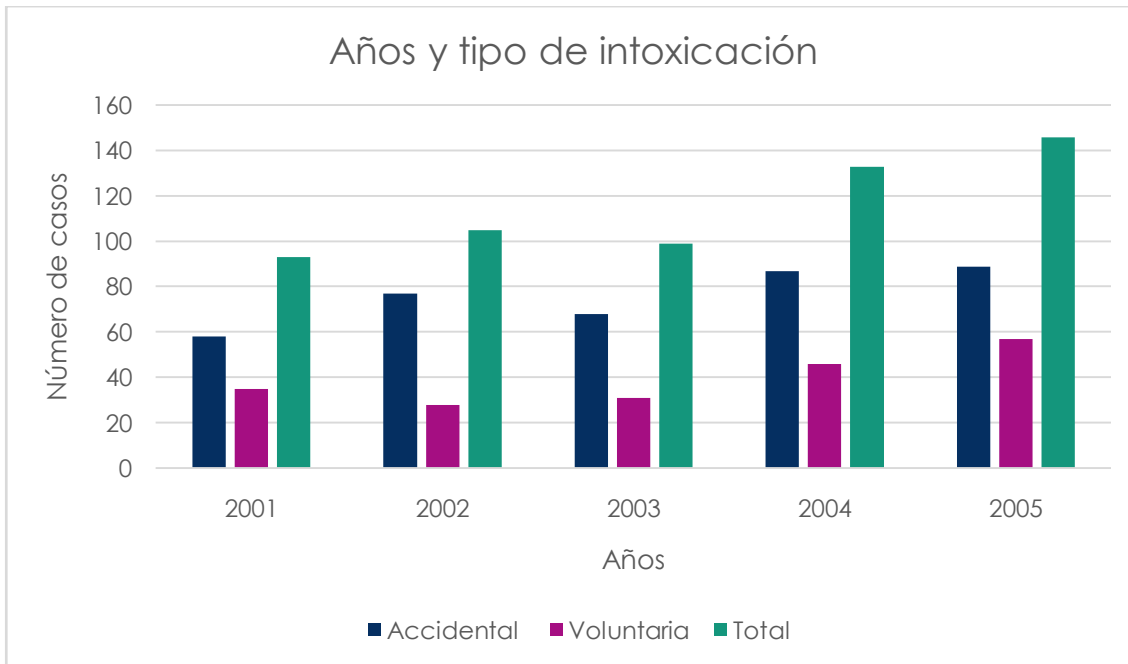


Figura 1 Distribución de las intoxicaciones agudas según años y tipo de intoxicación. Fuente: (Dra Rodríguez Rubinos, et al., 2007)

En el estudio predominaron tanto en las intoxicaciones accidentales como en las voluntarias los medicamentos con 111 casos y 131 casos respectivamente, que representan en su totalidad el 42.01% de los pacientes. Para las intoxicaciones accidentales le siguen en orden de frecuencia los productos de uso industrial y los derivados del petróleo con 92 pacientes cada uno. En relación a las intoxicaciones voluntarias el alcohol ocupó la segunda causa más frecuente con 28 casos como muestra la siguiente tabla.

Tabla 2 Distribución de las intoxicaciones agudas según agente causal y tipo de intoxicación. Fuente: (Dra Rodríguez Rubinos, et al., 2007)

Agente causal	Accidental		Voluntaria		Total	
	No.	%	No.	%	No.	%
Medicamentos	111	19.27	131	22.74	242	42.01
Producto uso industrial	92	15.97	4	0.69	96	16.67
Derivados del petróleo	92	15.97	2	0.35	94	16.32

Plantas tóxicas	39	6.77	1	0.17	40	6.94
Alcohol	10	1.74	28	4.86	38	6.60
Plaguicidas	10	1.74	7	1.22	17	2.95
Productos del hogar	13	2.28	2	0.35	15	2.60
Otros productos químicos	5	0.87	9	1.56	14	2.43
Desconocido	7	1.22	6	1.04	13	2.26
Combinado	0	0.00	6	1.04	6	1.04
Drogas de abuso	0	0.00	1	0.17	1	0.17
Total	379	65.80	197	34.20	576	100.00

Resultados similares se han encontrado en otros estudios epidemiológicos realizados en diferentes provincias del país como Cienfuegos y Ciego de Ávila con un evidente incremento del número de pacientes con el paso de los años y un predominio de las intoxicaciones accidentales en la población infantil. (Dra Rodríguez Rubinos, et al., 2007)

1.3 Análisis del estado del arte

Se realiza un análisis sobre las herramientas y software existentes utilizados en centros hospitalarios teniendo en cuenta cinco aspectos fundamentales: si emiten diagnósticos, si filtran información, si su utilización resulta fácil, si brindan soporte y si se integran a otros sistemas de salud. Con el objetivo de determinar si pueden ser utilizados en el CENATOX o es necesaria la implementación de una nueva herramienta para la gestión de información de los casos de intoxicaciones en el mismo.

IPCS: Este programa proporciona una base científica, evaluada internacionalmente, para establecer normas de seguridad química. Se propone reforzar, a nivel de cada país participante, la capacidad de prevenir los accidentes químicos, tratar a los afectados, en caso de que se produzcan accidentes, realizar preparativos para casos de accidente químico, vigilar y reducir la morbilidad y mortalidad de las intoxicaciones por productos químicos. (La gestión de información como herramienta fundamental en el desarrollo de los centros tecnológicos, 2003) (1998)

Los centros de control de intoxicaciones y las redes nacionales de centros de toxicología brindan un servicio incalculable en respuesta a todos estos problemas. El IPCS ha elaborado directivas para el establecimiento y la gestión de estos servicios y facilita datos evaluados para el diagnóstico y tratamiento de intoxicaciones por productos químicos. (La gestión de información como herramienta fundamental en el desarrollo de los centros tecnológicos, 2003)

INTOX/IPCS: Este está conformado por el sistema IPCS/INTOX, programa para el manejo de la base de datos de información toxicológica, y el banco de datos del mismo nombre (disponible en disco compacto y en Internet) es una colección de documentos sobre sustancias tóxicas. Ambos proporcionan información sobre sustancias industriales, farmacéuticas, de uso doméstico, agroquímicas y toxinas de plantas, hongos y animales, así como otros agentes que causan intoxicaciones. (La gestión de información como herramienta fundamental en el desarrollo de los centros tecnológicos, 2003) (Organización Mundial de la Salud, 1998) (1998)

INTOX: Permite el intercambio de información entre los expertos, así como la respuesta rápida ante emergencias químicas específicas. El intercambio se realiza mediante una lista de discusión que agrupa a los especialistas de todo el mundo. Algunos países han comenzado a utilizar esta herramienta. A partir del formato establecido para el informe de los casos, países como Argentina, México y Brasil han desarrollado su propio software. (La gestión de información como herramienta fundamental en el desarrollo de los centros tecnológicos, 2003)

BVSA-TOXICOLOGIA: Esta biblioteca es un recurso de información para el progreso de la gestión de información en el espacio virtual que facilita el acceso a: bases de datos bibliográficas y a texto completo, directorios de instituciones y sociedades, revistas, documentos a texto completo (libros, artículos), búsquedas en Internet con control de calidad, páginas web institucionales, materiales educativos, estudios de casos clínicos y ambientales, así como la disseminación selectiva de la información a los especialistas interesados en el tema. (La gestión de información como herramienta fundamental en el desarrollo de los centros tecnológicos, 2003) (La Biblioteca Virtual de Salud-Toxicología, 1999)

La BVSA-Toxicología permite:

- Organizar la información acerca de la especialidad (que es abundante y dispersa).
- Agilizar la búsqueda de información (se accede a una sola dirección electrónica al inicio de la búsqueda).

- Ofrecer una información selectiva y confiable.
- Agrupar instituciones, sociedades y profesionales de acuerdo con categorías y especialidades dentro de la temática escogida.
- Brindar ayuda para la educación de todos.
- Intercambiar información mediante una lista de discusión (RETOXLAC).
- Contactar a los diferentes profesionales en tiempo real (sala de charla).

La OPS, por intermedio de CEPIS, ha comenzado a desarrollar la iniciativa de implementar un Sistema Regional de Información Toxicológica, sustentado en la promoción de la armonía en los informes de casos, así como en la recolección de la información que procesan los CIAAT. Su información estará disponible mediante la BVSA-Toxicología. (La gestión de información como herramienta fundamental en el desarrollo de los centros tecnológicos, 2003)

Sistema Automatizado de Información Toxicológica de Urgencia en el CENATOX: Este sistema, según los profesionales que lo utilizaron resulta difícil y tedioso el trabajo con él, ellos agregan que su uso no les resulta factible por su poco entendimiento y que ya no cumple con los parámetros que se necesitan en la actualidad para el servicio en línea que se presta. (Dr. C. My. Fleites Mestres, 2019)

Plaguix Software para la gestión de información de los casos de intoxicaciones por plaguicidas en el CENATOX: Este software cumple con los aspectos fundamentales a evaluar en este epígrafe, pero su desventaja es que solo emite esta información para los casos de plaguicidas. (Mesa Cisneros, et al., 2018)

1.4 Comparación de los sistemas.

La siguiente tabla muestra un análisis de las herramientas anteriores teniendo en cuenta diferentes aspectos.

Tabla 3 Análisis de los sistemas propuestos. Fuente: (Elaboración propia)

Softwares	Emite diagnósticos	Filtra información	Interfaces amigables	Brinda Soporte	Se integra a otros sistemas de salud
Plaguix	Sí (pero solo de plaguicidas)	Sí	Sí	No	No

Sistema Automatizado de Información Toxicológica de Urgencia en el CENATOX	Sí	Sí	No	No	No
IPCS	Sí	Sí	Sí	Sí	Sí, pero al cubano no
INTOX/IPCS	Sí	Sí	Sí	Sí	Sí, pero al cubano no
INTOX	Sí	Sí	Sí	Sí	Sí, pero al cubano no
BVSA-TOXICOLOGIA	Sí	Sí	Sí	Sí	Sí, pero al cubano no

Existen diversidad de herramientas y softwares para la visualización y/o gestión de las documentaciones de casos de intoxicaciones en el país y en el mundo. Entre todas tienen muchas similitudes y a la vez notorias diferencias. Dentro de ellas hay unas que resultan más profesionales que otras, pero a su vez, estas generalmente son más caras. La mayoría de las herramientas analizadas no son de origen nacional, por lo que no se garantiza licencias o soporte para ellas. La consulta debe ser en línea, por lo que la imposibilita mientras no exista conexión a internet. Los sistemas más actualizados están en idioma inglés. Todos dirigen la consulta hacia la consulta telefónica con expertos. Estos programas al ser de otros países no tienen en cuenta la situación toxicológica de nuestro medio. Las que son de origen cubano no abarcan las necesidades actuales, una solo incluye los casos de intoxicaciones por plagas (PlaguitoX), mientras que la otra no les resulta de ayuda a los especialistas por las diversas razones planteadas.

Una vez finalizado el estudio de los sistemas de gestión de información epidemiológica e intoxicaciones se concluye que se tiene que implementar un nuevo software que sea capaz de dar solución a los requisitos exigidos, pues las herramientas y softwares existentes, en general, no son capaces de contribuir a la recolección de datos acerca de casos de intoxicaciones presentando un diagnóstico rápido y conciso.

1.5 Metodología de desarrollo

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software. (2005). Estas a su vez se dividen en ágiles y tradicionales.

Las metodologías tradicionales se enfocan en documentación, planificación y procesos, plantillas, técnicas de administración, revisiones, entre otros. Ejemplos de estas metodologías son: Proceso Racional Unificado (RUP) y Microsoft Solution Frame (MSF).

Las metodologías ágiles ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan. Para muchos clientes esta flexibilidad será una ventaja competitiva y porque estar preparados para el cambio significa reducir su coste. Ejemplos de estas metodologías son Programación Extrema (XP) y Proceso Unificado Ágil (AUP). (Pressman, 2007)

Principios de agilidad

Se definen 12 principios de agilidad para aquellos que la quieran alcanzar (Pressman, 2007):

1. La prioridad más alta es satisfacer al cliente a través de la entrega pronta y continua de software valioso.
2. Son bienvenidos los requerimientos cambiantes, aun en una etapa avanzada del desarrollo. Los procesos ágiles dominan el cambio para provecho de la ventaja competitiva del cliente.
3. Entregar con frecuencia software que funcione, de dos semanas a un par de meses, de preferencia lo más pronto que se pueda.
4. Las personas de negocios y los desarrolladores deben trabajar juntos, a diario y durante todo el proyecto.
5. Hay que desarrollar los proyectos con individuos motivados. Debe darse a éstos el ambiente y el apoyo que necesiten, y confiar en que harán el trabajo.

6. El método más eficiente y eficaz para transmitir información a los integrantes de un equipo de desarrollo, y entre éstos, es la conversación cara a cara.
7. La medida principal de avance es el software que funciona.
8. Los procesos ágiles promueven el desarrollo sostenible. Los patrocinadores, desarrolladores y usuarios deben poder mantener un ritmo constante en forma indefinida.
9. La atención continua a la excelencia técnica y el buen diseño mejora la agilidad.
10. Es esencial la simplicidad: el arte de maximizar la cantidad de trabajo no realizado.
11. Las mejores arquitecturas, requerimientos y diseños surgen de los equipos con organización propia.
12. El equipo reflexiona a intervalos regulares sobre cómo ser más eficaz, para después afinar y ajustar su comportamiento en consecuencia.

1.5.1 Metodología Programación Extrema (XP)

Programación extrema, XP por sus siglas en inglés se basa en una serie de reglas y principios que se han ido gestando a lo largo de toda la historia de la ingeniería del software. Usadas conjuntamente proporcionan una nueva metodología de desarrollo de software que se puede englobar dentro de las metodologías ligeras y se clasifica como evolutiva. (Programación Extrema: Prácticas, Aceptación y Controversia, 2015)

Esta 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 con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Agile Processes in Software Engineering and Extreme Programming: 20th International Conference, 2019) (Extreme programming implementation in academia for software engineering sustainability. In: 2018 Advances in Science and Engineering Technology International Conferences (ASET), 2018)

Características de la metodología XP

- XP es una metodología "liviana" que no tiene en cuenta la utilización de casos de uso y la generación de una extensa documentación.
- XP tiene asociado un ciclo de vida y es considerado a su vez un proceso.
- La tendencia de entregar software en espacios de tiempo cada vez más pequeños con exigencias de costos reducidos y altos estándares de calidad.
- XP define Historias de Usuario (HU) como base del software a desarrollar, estas historias las escribe el cliente y describen escenarios sobre el funcionamiento del programa. A partir de las HU y de la arquitectura perseguida se crea un plan de liberaciones entre el equipo de desarrollo y el cliente. (Villegas, 2009)

Fases de la metodología XP (Desarrollo ágil de software aplicando programación extrema, 2014)

- Planificación: durante esta etapa se lleva a cabo el proceso de identificación y confección de las HU.
- Diseño: durante esta etapa se crea un diseño evolutivo que va mejorando incrementalmente y que permite hacer entregas pequeñas y frecuentes de valor para el cliente, basado principalmente en el desarrollo de las tarjetas Clase-Responsabilidad-Colaboración (CRC).
- Desarrollo: en esta fase se realiza la implementación de las HU que fueron seleccionadas por cada iteración. Al inicio se lleva a cabo un chequeo del plan de iteraciones por si es necesario realizar modificaciones. Como parte de este plan se crean tareas de ingeniería para ayudar a organizar la implementación exitosa de las HU.
- Pruebas: esta fase permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones. XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñadas por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñada por el cliente final.



Figura 2 Proceso XP. Fuente: (Pressman, 2007)

El ciclo de desarrollo consiste en los pasos siguientes:

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

Ventajas de la metodología XP(Pressman, 2007)

- Apropiado para entornos volátiles.
- Estar preparados para el cambio, significa reducir su coste.
- Planificación más transparente para los clientes, conocen las fechas de entrega de funcionalidades. Vital para su negocio.
- Permitirá definir en cada iteración cuáles son los objetivos de la siguiente.

Valores de la metodología XP

Para todo trabajo que se realiza con la metodología XP se definen cinco valores, ellos son: comunicación, simplicidad, retroalimentación, valentía y respeto. Cada uno de estos valores se usa como un motor para actividades, acciones y tareas específicas de XP. (Pressman, 2007)

Esta metodología define, además, un conjunto de prácticas que son esenciales para el éxito de los proyectos y no deben pasarse por alto en el proceso de desarrollo, estas son (Joskowicz, 2008):

- **Recodificación:** consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de hacerlo más simple, conciso y/o entendible.
- **Pruebas unitarias:** las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben pasar las pruebas unitarias antes de ser liberados o publicados.
- **Pruebas de aceptación:** las pruebas de aceptación son creadas en base a las HU, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Las pruebas de aceptación son consideradas como “pruebas de caja negra”. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos.
- **Reuniones diarias de seguimiento:** el objetivo de tener reuniones diarias es mantener la comunicación entre el equipo, y compartir problemas y soluciones.

Para desarrollar la presente investigación fue decidido utilizar como metodología de desarrollo XP, ya que esta es la que más se adecúa a las características del equipo de trabajo, entorno, la duración del proyecto y la relación con el cliente. La duración del proyecto es relativamente corta, el equipo de trabajo está formado por una programadora. Además, se preocupa más en el avance exitoso del producto que en la generación de una documentación detallada del mismo, siendo capaz de adaptarse a los cambios de requisitos en cualquier punto del ciclo de vida del proyecto.

1.6 Herramientas para el modelado y representación de procesos de negocio

A continuación, se describen las principales herramientas y tecnologías seleccionadas para ser utilizadas en el proceso de desarrollo del software, las cuales son de software libre.

Ambiente de desarrollo

Para el desarrollo de la herramienta fueron seleccionados un conjunto de tecnologías, lenguajes y herramientas, que se describen a continuación.

1.6.1 Herramientas

Symfony

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja.

Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. (Solucionex, 2009)

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de Propel, permiten cambiar con facilidad de Sistema de Gestión de Bases de Datos (SGBD) en cualquier fase del proyecto.
- Utiliza programación orientada a objetos, de ahí que sea imprescindible PHP 5.
- Sencillo de usar en la mayoría de casos, está más (Solucionex, 2009) indicado para grandes aplicaciones web que para pequeños proyectos.

Aunque utiliza MVC (Modelo vista controlador), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.

Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.

Sigue la mayoría de mejores prácticas y patrones de diseño para la web. Preparado para aplicación empresarial y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo. Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo. Fácil de extender, lo que permite su integración con las librerías de otros fabricantes (Solucionex, 2009).

Herramienta CASE: Visual Paradigm 8.0

Visual Paradigm es una herramienta CASE que propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Es una herramienta con disponibilidad en múltiples plataformas y soporta varios Sistemas Gestores de Bases de Datos como Oracle, Microsoft SQL Server, MySQL PostgreSQL y SQLite. También permite el uso de las distintas metodologías propias de la Ingeniería de Software. (Pressman, 2007)

1.6.2 Lenguaje

PHP

Conocido como un lenguaje basado en servidores. Esto es porque el PHP no se ejecuta en la computadora del cliente, sino en la computadora que visita la página. Los resultados son brindados hacia el cliente y mostrados en su navegador (Ángel Cobo, 2005).

Ventajas de php:

PHP es compatible con (casi) cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de Unix, Windows (95,98, NT, ME,2000, XP, hasta las últimas versiones) y Macs. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al OS.

La sintaxis de PHP es similar a la del C, por esto cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP. Entre los lenguajes del tipo C se incluye Java y JavaScript, de hecho, mucha de la funcionalidad del PHP se debe a C en funciones como fread () o strlen (), esto provoca que los desarrolladores se sientan más cómodos.

PHP es completamente expandible. Está compuesto de un sistema principal (escrito por Zend), un conjunto de módulos y una variedad de extensiones de código. (Gonzalez, 2016)

Muchas interfaces distintas para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTPD. Otra alternativa es configurarlo como módulo CGI. Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos. Siempre podrás disponer de ODBC para situaciones que lo requieran.

Una gran variedad de módulos cuando un programador PHP necesite una interface para una librería en particular, fácilmente podrá crear una API para esta. Algunas de las que ya vienen implementadas

permiten manejo de gráficos, archivos PDF, Flash, Cybercash, calendarios, XML, IMAP, POP, entre otras.

PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz, así que se ejecuta rápidamente utilizando poca memoria.

PHP es Open Source, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no estás forzado a pagar actualizaciones anuales para tener una versión que funcione. (Gonzalez, 2016)

Además de PHP se encuentran otros lenguajes de programación cliente/servidor como es el caso de Perl que permite su uso tanto en desarrollo web como en otros entornos, posee una buena integración con la mayoría de los sistemas, pero puede ser lento ya que este compila al inicio de su ejecución y al no disponer de control de excepciones se vuelve complicado encontrar algunos errores. También se encuentra Rust que es un lenguaje rápido y seguro (gestión automática de guardado y el manejo de la memoria), está basado en expresiones, pero no se puede utilizar un valor que no haya sido inicializado previamente y es más complejo de aprender que otros lenguajes ya que algunas de sus reglas son diferentes a las habituales. (González, 2018)

En cuanto a la definición general, MySQL es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS, por sus siglas en inglés) con un modelo cliente-servidor. RDBMS es un software o servicio utilizado para crear y administrar bases de datos basadas en un modelo relacional. (Fernandez, 2015).

Lenguaje de modelado: UML

UML son las siglas de “lenguaje unificado de modelado”. Se trata de un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software.

1.6.3 Sistema gestor de base de datos

Un SGBD es un conjunto de programas que permiten la creación de base de datos y proporciona herramientas para añadir, borrar, modificar, y eliminar datos, además de mantener la integridad de estos.

MySQL

Presenta varias ventajas para los desarrolladores como son:

- Verificación basada en host.
- Puede usarse, aunque no haya ninguna red disponible.
- Sistema flexible de privilegios y contraseñas.
- Cifrado de seguridad de todo el tráfico de contraseñas
- La biblioteca se puede integrar en aplicaciones independientes.
- Ofrece el servidor por separado para entornos cliente/servidor. (Cabrera, 2019)

Existen además de MySQL otros RDBMS como por ejemplo MariaDB que, a pesar de tener una alta escalabilidad, fácil integración y acceso en tiempo real como desventaja presenta la carencia de interfaz en caché de memoria y la carencia de rastreador de optimización. Otro ejemplo sería MongoDB que posee validación de documentos y motor de almacenamiento cifrado este RNBMS no es adecuada para aplicaciones que requieren transacciones complejas, como tampoco es adecuada para la sustitución directa de aplicaciones heredadas y es usado con mayor frecuencia en aplicaciones móviles y catálogos de producto. (Cabrera, 2019)

1.7 Conclusiones del capítulo

El análisis de la situación toxicológica global y nacional permitió la comprensión de la necesidad de una herramienta informática de interfaz amigable que brinde rapidez en el tiempo de respuesta en casos de intoxicaciones para la prevención y control de eventos.

El análisis de las herramientas y sistemas existentes para la representación de la información toxicológica mostró deficiencias para su empleo nacional, por lo que se decide el desarrollo de una que permita cubrir las limitantes encontradas.

El análisis de las tecnologías utilizadas para el desarrollo de herramientas de estas características, propició recopilar los basamentos teóricos para su selección.

Capítulo 2. Propuesta de solución

La propuesta de solución que se caracteriza en este capítulo consta de un sistema de gestión que trata la información referente a los casos de intoxicación en el CENATOX. Se realiza la descripción de los procesos del negocio, definiendo los trabajadores. Se especifican los requisitos funcionales y no funcionales, se obtienen los artefactos correspondientes a las fases de planificación, diseño e implementación de la metodología seleccionada. Es definida la arquitectura y los principales patrones de diseño utilizados en el desarrollo de la solución.

2.1 Proceso de negocio a informatizar

El proceso del negocio describe cada proceso del negocio, especificando sus actividades, datos, roles y reglas del mismo. Facilita un entendimiento entre cliente y desarrollador, además se enfoca en comprender los problemas actuales de la organización e identifica mejoras potenciales.

El proceso de gestión de intoxicaciones inicia una vez que el especialista obtiene la información de que se añaden nuevas sustancias, se modifican o son eliminadas del mercado. En el caso de eliminación/modificación retiran las Fichas Manuales anteriores y las remplazan en caso necesario. En caso de una adición, el especialista agrega la Ficha Manual proporcionada por el proveedor de la nueva intoxicación al registro del Centro. Los datos se almacenan en hojas de papel que son archivadas posteriormente en los archivos que se destinan al almacenamiento de las Fichas Manuales.

El proceso de ayuda a pacientes empieza cuando un solicitante llama por teléfono al CENATOX para reportar un caso de intoxicación. El especialista del centro registra los datos generales del paciente y del solicitante, los datos del agente casual, la vía de exposición, los datos de la conducta y evolución del paciente, la identificación de los peligros, las propiedades físicas y químicas de las sustancias de manera manual. Una vez que se registran los datos se hace una profunda búsqueda en las Fichas Manuales de los tipos de intoxicaciones, identificando así las coincidencias entre las sesiones de identificación de los peligros, propiedades físicas y químicas, entre otras informaciones que coincidan con los datos registrados a partir de la llamada. De esta manera los especialistas del centro, apoyándose en la sección de primeros auxilios que emite la Ficha Manual, evalúan un posible diagnóstico.

Personas que intervienen en el negocio

Una vez que se identifican los procesos de negocio, es necesario encontrar los involucrados en su realización. A continuación, se muestran las personas que se identificaron en la solución propuesta:

Tabla 4 Personas que intervienen en el negocio. Fuente: (Elaboración propia)

Nombre	Descripción
Solicitante	Persona que describe los datos del nuevo caso de intoxicación.
Especialista del CENATOX	Persona encargada de gestionar un caso de intoxicación y emitir un diagnóstico

2.2 Análisis de la solución

Para brindar solución a la problemática planteada se propone el desarrollo de una aplicación web que contará con cuatro interfaces ofreciendo una herramienta capaz de garantizar la gestión de intoxicaciones con la ayuda de las Fichas Manuales de cada reactor tóxico, ofreciendo así la posibilidad de filtrar búsquedas con los datos recopilados otorgados por la persona que realiza la llamada al centro, siendo capaz así de confeccionar un posible diagnóstico.

La herramienta será capaz de gestionar las Fichas Manuales de los reactores tóxicos que estén presente o relacionados con la intoxicación del paciente, la información será analizada por el especialista para contiguamente emitir un diagnóstico.

Descripción de la herramienta

Para resolver la problemática propuesta se analizan las características que debe tener el software para que sean cumplidos los objetivos planteados. Para ello son identificadas las funcionalidades y los requisitos no funcionales de la herramienta.

Funcionalidades de la herramienta

Los requisitos funcionales (RF) para un sistema describen lo que este debe hacer. Los RF son declaraciones de servicios que el sistema debe proporcionar, cómo debe reaccionar el sistema entradas particulares y cómo debe comportarse el sistema en situaciones particulares. En algunos

casos, los requisitos funcionales también pueden indicar explícitamente lo que el sistema no debe hacer. (Sommerville, 2015)

A continuación, son mostradas las funcionalidades que a partir de los procesos de negocio estudiados y las tareas realizadas en el CENATOX fueron identificadas:

Tabla 5 Descripción de los requisitos funcionales. Fuente: (Elaboración propia)

Requisito funcional	Descripción
RF1 Gestionar usuario	Esta funcionalidad permite gestionar un usuario.
RF2 Adicionar usuario	Esta funcionalidad permite adicionar un usuario.
RF3 Eliminar usuario	Esta funcionalidad permite eliminar un usuario.
RF4 Modificar usuario	Esta funcionalidad permite modificar un usuario.
RF5 Autenticar usuario	Esta funcionalidad permite autenticar los usuarios que harán uso del sistema, especificando dos roles que pueden ser asignados (Administrador y Especialista).
RF6 Autocompletar consulta	Esta funcionalidad permite que la herramienta mientras que el especialista está escribiendo el nombre genérico o los otros nombres del reactor tóxico esta le autocomplete la posible búsqueda.
RF7 Consultar Ficha Manual	Esta funcionalidad permite consultar las fichas manuales que contienen la información de los reactores tóxicos.

RF8 Visualizar Ficha Manual	Esta funcionalidad permite visualizar las fichas con las que se trabaja en el CENATOX.
RF9 Exportar a PDF	Esta funcionalidad permite exportar un documento en formato PDF que contiene la Ficha Manual.

Requisitos no funcionales de la herramienta

Los requisitos no funcionales (RNF) son restricciones en los servicios o funciones que ofrece el sistema. Incluyen restricciones de tiempo, restricciones en el proceso de desarrollo y restricciones impuestas por los estándares. Los requisitos no funcionales a menudo se aplican al sistema como un todo en lugar de las características o servicios individuales del sistema. (Sommerville, 2015)

Las cualidades que debe tener para un correcto funcionamiento un software son tenidas en cuenta para realización correcta del producto final. A continuación, se listan los requisitos no funcionales definidos previamente con el cliente:

- **Usabilidad**

RNF 1: Los grupos de botones y vínculos se deben organizar por funcionalidad. (Esto con el objetivo de facilitar al usuario la interacción con el software)

RNF 2: Los mensajes para interactuar con los usuarios y los de errores deben tener la suficiente información, en idioma español y no deben revelar información interna. (Por ejemplo, cuando el usuario no haya puesto la contraseña y seleccione aceptar debe mostrar el mensaje de error siguiente: Debe llenar los campos correspondientes)

- **Software**

Características del software de los ordenadores donde se desee instalar el sistema:

- ✓ RNF 3: Debe tener como mínimo Sistema operativo: Windows 7 Ultimate 64 bits.

- **Hardware**

Características del hardware de los ordenadores donde se desee instalar el sistema:

- ✓ RNF 4: Debe tener como mínimo una RAM: 2 GB.
- ✓ RNF 5: Debe tener como mínimo un procesador: Intel (R) Dual Core CPU @ 2.50 GHz.
- ✓ RNF 6: Debe tener como mínimo un espacio en disco disponible de 500 GB.

- **Interfaz**

RNF 7: El software debe tener una interfaz sencilla con una navegación intuitiva para el usuario.

- **Seguridad**

RNF 8: La herramienta debe requerir la autenticación como primera acción, con un nombre de usuario y una contraseña, que debe ser de conocimiento exclusivo de las personas que se pueden autenticar. Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.

RNF 9: La información solo puede ser modificada por aquellos usuarios que son autorizados.

RNF 10: La herramienta debe estar disponible en todo momento para las personas autorizadas que necesiten acceder y manejar la información.

Estas características mencionadas anteriormente ponen en práctica los principios de la seguridad: integridad, confidencialidad y disponibilidad.

El centro debe garantizar la seguridad de los datos almacenados en la computadora donde se utiliza la herramienta, esto se debe a que la base de datos con la que trabaja la herramienta estará en la propia computadora.

2.3 Fase de Planificación

La planeación es la etapa inicial que plantea la metodología XP. En este punto se comienza a interactuar con el cliente y el resto del grupo de desarrollo para así descubrir los requerimientos del sistema, es llevado a cabo el proceso de identificación y confección de las historias de usuario, así como la familiarización del equipo de trabajo con las tecnologías y herramientas seleccionadas para el desarrollo del software. 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 (Extreme programming implementation in academia for software engineering sustainability. In: 2018 Advances in Science and Engineering Technology International Conferences (ASET), 2018). Se identifican el número y tamaño de las iteraciones al igual que se plantean ajustes necesarios a la metodología según las características del proyecto. El resultado de la fase es un plan de entregas donde se realiza una estimación de las versiones que tendrá el producto en su realización, de manera tal que guíe su desarrollo. (Agile Processes in Software Engineering and Extreme Programming: 20th International Conference, 2019)

2.3.1 Historias de Usuarios

Las HU constituyen la técnica utilizada en XP para especificar los requisitos del software; en ellas el cliente describe brevemente las características que el sistema debe poseer, y se realiza una por cada característica principal del sistema. La redacción de las mismas se realiza bajo la terminología del cliente, no del desarrollador, de forma que sea clara y sencilla, sin profundizar en detalles. El tratamiento de las HU es muy dinámico y flexible, en cualquier momento pueden reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada HU es lo suficientemente comprensible y delimitada para que los programadores puedan implementarlas en unas semanas. (Extreme programming implementation in academia for software engineering sustainability. In: 2018 Advances in Science and Engineering Technology International Conferences (ASET), 2018)

Las HU son equivalentes a los casos de uso en otras metodologías. Deben poder ser programadas en un tiempo entre una y tres semanas. Si la estimación resulta superior a tres semanas, debe ser dividida en dos o más historias. Si es en menos de una semana, se debe combinar con otra historia. (Letelier, et al., 2006)

Las HU correspondientes al software fueron creadas especificando la prioridad de cada una de ellas. Se representan a través de tablas con los elementos siguientes:

- **Número:** número de la HU, incremental en el tiempo.
- **Nombre:** nombre que identifica la HU.
- **Referencia:** es el conjunto de HU de las que depende la actual.
- **Prioridad:** esta característica se obtiene del cliente con los valores: alta, media o baja, en dependencia de la importancia y orden de la implementación.
 - ✓ Alta: se le otorga a las HU que resultan funcionalidades fundamentales en el desarrollo de la herramienta, las que el cliente define como principales para el control de la misma.
 - ✓ Media: se le otorga a las HU que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.
 - ✓ Baja: se le otorga a las HU que constituyen funcionalidades que ayudan al control de elementos asociados al equipo de desarrollo, a la estructura, y no tienen que ver directamente con el sistema de desarrollo.
- **Iteración asignada:** número de la iteración en la cual se desarrollará la HU.
- **Puntos estimados:** tiempo estimado en semanas que será asignado.
- **Descripción:** breve descripción del proceso que define la HU.
- **Observaciones:** alguna acotación importante que se desee señalar sobre la HU.
- **Prototipo de interfaz:** prototipo de interfaz, si aplica.

Fueron definidas 6 HU, de ellas 3 se clasificaron con prioridad alta y las otras 3 con prioridad media. A continuación, se representan 2 de las HU clasificadas con prioridad alta.

Tabla 6 Descripción de la HU 4 Consultar Ficha Manual. Fuente: (Elaboración propia)

Historia de Usuario	
Número: 4	Nombre de HU: Consultar Ficha Manual
Referencia: Autocompletar consulta	Prioridad: alta
Iteración Asignada: 1	Puntos Estimados: 1.5
Descripción: El software debe permitir consultar la Ficha Manual correspondiente a cada tipo de reactor tóxico por los especialistas del centro.	
Observaciones: Los reactores tóxicos son almacenados en las Fichas Manuales indicando: Nombre genérico, Clasificación, Otros nombres, Cuadro Clínico, Código, Versión, Fecha, Tratamiento e Información adicional.	

Tabla 7 Descripción de la HU 5 Visualizar Ficha Manual. Fuente: (Elaboración propia)

Historia de Usuario	
Número: 5	Nombre de HU: Visualizar Ficha Manual
Referencia: Consultar Ficha Manual	Prioridad: alta
Iteración Asignada: 2	Puntos Estimados: 1
Descripción: El software debe permitir visualizar los datos necesarios del caso del tipo de intoxicación buscada en las Fichas Manuales.	
Observaciones: Se visualizan los campos correspondientes en la Ficha Manual con la información que considere necesaria el especialista mientras se la otorga al consultante, teniendo la posibilidad de exportarla a pdf.	

2.3.2 Estimación del esfuerzo por HU

Las diferentes estimaciones de esfuerzo asociadas a la implementación de las historias las establecen los programadores utilizando como medida el punto (Letelier, et al., 2006), esto se conoce como estimación de esfuerzo por puntos. Para ello el programador define una puntuación para cada HU de acuerdo a la complejidad de implementación de la misma para ellos, tomando en cuenta una medida de tiempo por cada punto de estimación. A continuación, es mostrada la estimación de esfuerzo por cada HU:

Tabla 8 Estimación del esfuerzo por HU. Fuente: (Elaboración propia)

No.	Nombre de Historia de Usuario	Puntos de estimación (semanas)
1	Gestionar usuario	1.5
2	Autenticar usuario	1
3	Autocompletar consulta	1.5
4	Consultar Ficha Manual	1.5
5	Visualizar Ficha Manual	1
6	Exportar a PDF	1.5

Con la realización de la estimación de esfuerzo por puntos, el tiempo de desarrollo obtenido fue de 8 semanas. Los tiempos estimados de las HU variaron de 1 a 1.5 semanas.

2.3.3 Plan de iteraciones

Después de estimar el tiempo de desarrollo de las HU, como parte de la metodología XP, se procede a realizar el plan de iteraciones para así implementar por un orden lógico cada funcionalidad propuesta. A continuación el plan de iteraciones:

- Iteración 1: en esta iteración es confeccionada la arquitectura del software, son definidas las clases del modelo, se realiza la conexión a la base de datos y son diseñadas las interfaces. Se desarrollan las HU uno y dos correspondientes al trabajo con usuarios que utilizan el software, la número tres y la cuatro correspondientes a Autocompletar consulta y Consultar Ficha Manual respectivamente. Como resultado será obtenida la herramienta en un 50 % de su implementación.
- Iteración 2: se desarrolla la HU cinco que es la encargada de Visualizar Ficha Manual. Como resultado se obtendrá la herramienta en un 85 % de su implementación.
- Iteración 3: se le da solución a la HU seis que es la encargada de exportar a PDF. Al culminar se desarrollan pruebas de aceptación y unitarias obteniendo así un 100 % de la herramienta, lista para realizar la entrega final de la misma.

Tabla 9 Plan de duración de las iteraciones. Fuente: (Elaboración propia)

Iteraciones	Historias de usuario a implementar	Duración de iteración (semanas)
Iteración 1	Gestionar usuario	5.5
	Autenticar usuario	
	Autocompletar consulta	
	Consultar Ficha Manual	
Iteración 2	Visualizar Ficha Manual	1
Iteración 3	Exportar a PDF	1.5
Total		8

2.3.4 Plan de entregas

El plan de entregas establece qué HU serán agrupadas para conformar una entrega, y el orden de implementación. En este plan son concentradas las funcionalidades referentes a un mismo tema en módulos, esto provee de un mayor entendimiento en la fase de implementación. Posee como objetivo definir el número de liberaciones que se realizarán en el transcurso del proyecto y las iteraciones que se requieren para desarrollar cada una.

Teniendo en cuenta que una semana de trabajo consta de 5 días laborables que tienen 8 horas de trabajo dedicadas en cada uno, se realizó el cálculo del tiempo. La siguiente tabla muestra el plan de duración de las entregas:

Tabla 10 Plan de Entrega. Fuente: (Elaboración propia)

	Final de la primera iteración	Final de la segunda iteración	Final de la tercera iteración
Herramienta	2da semana de abril	3ra semana de abril	1ra semana de mayo
Herramienta de gestión de información toxicológica	v1.0	v1.1	Finalizado

2.4 Fase de diseño

La metodología de desarrollo XP plantea prácticas especializadas que trabajan de forma directa en la realización del diseño para lograr un sistema robusto y reutilizable. En todo momento se trata de mantener la simplicidad para crear un diseño evolutivo que va mejorando incrementalmente.

El principio mantenerlo sencillo (MS) es seguido rigurosamente por el diseño de XP. Un diseño sencillo siempre es preferido sobre una representación más compleja. Además, el diseño guía la implementación de una historia conforme se escribe: nada más y nada menos. (Pressman, 2007)

En esta fase son confeccionadas las tarjetas CRC para así guiar el proceso de implementación de la solución, son descritos también los patrones de diseño utilizados. El diseño aporta una representación del software mediante el establecimiento de la arquitectura y el modelo de datos.

2.4.1 Arquitectura de Software

El diseño arquitectónico se preocupa por comprender cómo debe organizarse un sistema de software y diseñar la estructura general de ese sistema. En el modelo del proceso de desarrollo de software el diseño arquitectónico es la primera etapa del proceso de diseño de software. Es el vínculo crítico entre

el diseño y la ingeniería de requisitos, ya que identifica los principales componentes estructurales de un sistema y las relaciones entre ellos. El resultado del proceso de diseño arquitectónico es un modelo arquitectónico que describe cómo se organiza el sistema como un conjunto de componentes comunicantes. (Sommerville, 2015)

Modelo Vista Controlador

El patrón Modelo Vista Controlador (Model View Controller, MVC por sus siglas en inglés) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la interfaz de usuario y el código es el que provee de datos dinámicos a la página; el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista. (Pressman, 2007)(Sommerville, 2015)

Modelo: representación específica de la información con la que el sistema opera. Maneja los datos y controla sus transformaciones. No tiene conocimiento específico de los controladores y las vistas, ni siquiera contiene referencias a ellos. Es el sistema en sí el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas y notificar a las vistas cuando cambia el modelo.

Vista: maneja la representación visual de los datos representados por el modelo, el cual es presentado en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: responde a eventos, usualmente acciones del usuario, e invoca cambios en el modelo y probablemente en la vista.

La arquitectura de la solución propuesta está basada en el patrón arquitectónico Modelo Vista Controlador (MVC). Este patrón garantiza la reducción del esfuerzo de programación, la cual es necesaria en la implementación de sistemas múltiples (Patrón modelo-vista-controlador, 2014). Permite la separación de cada una de las capas del sistema: el modelo, donde se encuentran los datos y las reglas del negocio; la vista, que muestra la información del modelo al usuario y el controlador, que gestiona las entradas del usuario.

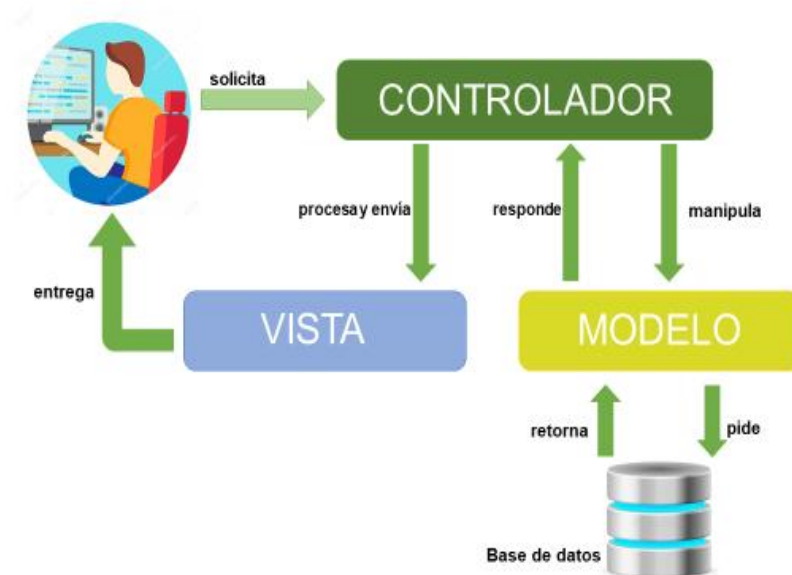


Figura 3 Arquitectura MVC. Fuente: (Elaboración propia)

Se utiliza el patrón anteriormente descrito, este separa la lógica de negocio de la interfaz de usuario en tres capas diferentes: clases controladoras, clases interfaz y la clase de acceso a datos; cada una con funcionalidades bien definidas, aminorando esfuerzo en la implementación de la herramienta y garantizando así una mejor organización del trabajo.

2.4.2 Tarjetas Clase-Responsabilidad-Colaboración

La técnica Clase-Responsabilidad-Colaboración (CRC) sugiere una forma de trabajo, preferentemente grupal, para hallar los objetos del dominio de la aplicación, sus responsabilidades y cómo colaboran con otros para realizar tareas. Esta técnica emplea las llamadas tarjetas CRC. Las tarjetas CRC son una técnica de diseño orientado a objetos. Identifican las clases y asociaciones que participan en el diseño del sistema, las responsabilidades que debe cumplir cada clase y establecen cómo una clase colabora con otras clases para cumplir con sus responsabilidades. Las tarjetas permiten a todos los miembros del proyecto apoyar con ideas y seleccionar las mejores dentro del diseño.

Las tarjetas CRC brindan una manera sencilla de identificación y organización de las clases que resultan relevantes para los requerimientos de un sistema o producto. Son confeccionadas con el objetivo de desarrollar una representación organizada de las clases. (Pressman, 2007)

Las tarjetas son divididas en tres sesiones:

- Clase: nombre de la clase que se corresponde con la tarjeta.
- Responsabilidad: describe las funcionalidades que contiene la clase.
- Colaboración: muestra las clases que guardan relación con la que se describe en la tarjeta.

A continuación, se muestran en las tablas 11 y 12 las tarjetas CRC que corresponden a clases del sistema.

Tabla 11 Tarjeta CRC clase Datos. Fuente: (Elaboración propia)

Tarjeta CRC	
Clase: Datos	
Responsabilidad	Colaboración
Es la clase encargada de: <ul style="list-style-type: none"> • Realizar la conexión a la base de datos. • Cerrar la conexión con la base de datos. • Hacer las consultas necesarias a la base de datos. 	Todas las clases por las que está conformada la aplicación.

Tabla 12 Tarjeta CRC clase PrincipalController. Fuente: (Elaboración propia)

Tarjeta CRC	
Clase: PrincipalController	
Responsabilidad	Colaboración
Es la clase encargada de: <ul style="list-style-type: none"> • Insertar datos. • Modificar datos. • Exportar a PDF. 	Generate_PDFFileiText Datos PrincipalController Usuario Reactor Tóxico

2.4.3 Patrones de Diseño

Un patrón de diseño se caracteriza como "una regla de tres partes que expresa una relación entre cierto contexto, un problema y una solución" (Alexander 1979). Para el diseño de software, el contexto permite al lector entender el ambiente en el que reside el problema y qué solución sería apropiada en dicho ambiente. Un conjunto de requerimientos, incluidas limitaciones y restricciones, actúan como sistema de fuerzas que influyen en la manera en la que puede interpretarse el problema en este contexto y en cómo podría aplicarse con eficacia la solución. (Pressman, et al., 2015)

Los patrones son técnicas para resolver problemas comunes en el desarrollo del software y otros ámbitos referentes al diseño de iteración o interfaces. Cada patrón describe un problema que ocurre en nuestro entorno y también, el núcleo de la solución al problema, de forma que puede utilizarse varias veces. (Tabares, 2014)

Los patrones de diseño contribuyen a reutilizar diseño gráfico, identificando aspectos claves de la estructura de un diseño que puede ser aplicado en una gran cantidad de situaciones. Estos proveen una estructura conocida por todos los programadores, de manera que la forma de trabajo no resulta diferente entre los mismos.

Resultan útiles en el desarrollo del software pues ahorran tiempo, ayudan a estar seguros de la validez del código y establecen un lenguaje común.

Patrones (GRASP) utilizados

Los Patrones Generales de Software para Asignar Responsabilidades (GRASP, por sus siglas en inglés), son parejas de problema y solución, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (2017). A continuación, se mencionan los patrones GRASP utilizados:

- **Bajo acoplamiento:** este patrón se evidencia en cada una de las clases diseñadas, lo que posibilita que, en el caso de que se produzca una modificación en alguna de ellas, se tenga la menor consecuencia en el resto de las clases.
- **Creador:** este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos. Es empleado al momento de definir la creación de objetos, se garantiza que si una clase puede crear objetos de otra es porque guardan determinada relación, ya sea que una contiene los datos que necesita la segunda o que una contiene a la otra.
- **Patrón alta cohesión:** cada elemento del diseño debe realizar una labor única dentro del software, no desempeñada por el resto de los elementos y auto-identificable, lo que hace más fácil la organización del proyecto y que las clases no se sobrecarguen de funcionalidades ajenas.

2.5 Fase de Implementación

Después de haber definido los elementos necesarios en las etapas de planificación y diseño se transita a la de codificación o implementación de la aplicación, donde se da cumplimiento al plan de iteraciones. En esta fase se realiza la implementación de las HU que fueron seleccionadas por cada iteración, también se crean las tareas de ingeniería para ayudar a organizar la implementación de las HU de forma exitosa. (Pressman, et al., 2015)(Agile Processes in Software Engineering and Extreme Programming: 20th International Conference, 2019)

2.5.1 Tareas de ingeniería

La metodología de software XP plantea que la implementación de un software se hace de manera iterativa. Durante cada iteración son desarrolladas un conjunto de HU que son definidas por el cliente y descritas por el equipo de desarrollo. En esta fase de implementación las HU son divididas en tareas de ingeniería, las que son asignadas a programadores para ser implementadas durante la iteración correspondiente. (Pressman, 2007)

Las tareas de ingeniería serán representadas mediante tablas divididas por las siguientes secciones (Gutiérrez, et al., 2006):

- **Número de tarea:** los números de cada tarea deben ser consecutivos.
- **Nombre de la tarea:** nombre que identifica a la tarea.
- **Número de historia de usuario:** número de la historia de usuario a la que pertenece la tarea.
- **Tipo de tarea:** las tareas pueden ser de Desarrollo, Corrección y Mejora.
- **Puntos estimados:** tiempo estimado en días que se le asignará a cada tarea.
- **Fecha inicio:** fecha en la que se comienza la tarea.
- **Fecha fin:** fecha en la que es concluida la tarea.
- **Programador responsable:** nombre y apellidos del programador que va a realizar la tarea.
- **Descripción:** descripción de la tarea.

Las HU están compuestas por una o varias tareas de ingeniería, con el objetivo de especificar las tareas llevadas a cabo por el programador responsable de ella o de ellas.

Tabla 13 Distribución de tareas de Ingeniería por HU. Fuente: (Elaboración propia)

HU	Tareas de Ingeniería por HU
Consultar Ficha Manual	Consultar Ficha Manual correspondiente a la

	búsqueda
Visualizar Ficha Manual	Visualizar Ficha Manual consultada

A continuación, en las tablas 14 y 15 son mostradas algunas de las tareas correspondientes a las HU 4 y 5:

Tabla 14Tarea de ingeniería Consultar Ficha Manual correspondiente a la búsqueda. Fuente: (Elaboración propia)

Tareas de ingeniería	
Número de tarea: 4	Número de HU: 4
Nombre de tarea: Consultar Ficha Manual correspondiente a la búsqueda.	
Tipo de tarea: Desarrollo	Puntos estimados: 10
Fecha de inicio: 3/4/2020	Fecha de fin: 12/4/2020
Programador responsable: Susel Hernández Trejo	
Descripción:Esta tarea permite que la herramienta realice la consulta de la Ficha Manual a la base de datos para así poder interactuar con la próxima interface en la ya se encuentra con la misma (Ficha Manual).	

Tabla 15Tarea de ingeniería Visualizar Ficha Manual consultada. Fuente: (Elaboración propia)

Tareas de ingeniería	
Número de tarea: 5	Número de HU:5
Nombre de tarea: Visualizar Ficha Manual consultada.	
Tipo de tarea: Desarrollo	Puntos estimados: 7
Fecha de inicio: 13/4/2020	Fecha de fin: 19/4/2020
Programador responsable: Susel Hernández Trejo	
Descripción:Esta tarea permite que la herramienta realice la visualización de la Ficha Manual consultada en la base de datos.	

2.5.2 Estándares de codificación

Entre los elementos más importantes que destaca XP se encuentran los estándares de codificación. XP resalta que la comunicación de los programadores es a través del código, por lo cual se hace necesario que sigan ciertos estándares de programación para lograr un entendimiento entre dichos programadores, de forma tal, que cualquier persona del equipo de desarrollo pueda modificar el

código. Además, se hace preciso que el código sea entendible para que posteriormente otros programadores puedan apoyarse en este trabajo y desarrollen otras soluciones. (Extreme programming implementation in academia for software engineering sustainability. In: 2018 Advances in Science and Engineering Technology International Conferences (ASET), 2018)

En el caso de la herramienta que se desarrolla en la presente investigación, el estándar que utiliza es:

Máxima longitud de las líneas

- Todas las líneas se limitan a un máximo de 79 caracteres.

Importaciones

- Las importaciones se encuentran en líneas separadas.

Comentarios

- Se utilizan comentarios de una línea para hacer más entendible el código.
- Comentarios de una línea: comentario pequeño que solo abarca una línea y describe el código que le sigue.

Estilo de los nombres

- Clases e Interfaces: los nombres de las clases presentan la primera letra en mayúscula, en caso de ser un nombre compuesto, la inicial de cada palabra se representa en mayúscula. Se utilizan nombres simples y de alguna manera que describan el contenido, se usan palabras completas, a no ser que la abreviatura sea muy conocida.
- Métodos y variables: los nombres de cada método se representan en minúscula, en caso de ser un nombre compuesto, la inicial de la primera palabra se simboliza en minúscula, y la de las otras palabras que lo componen en mayúscula. Los nombres de las variables son cortos, pero con significados lógicos, capaces de permitir a un observador identificar su función.

2.6 Conclusiones del capítulo

La definición de los requisitos funcionales permitió el diseño adecuado de la propuesta de solución en nueve funcionalidades a partir de lo que se necesitaba para dar solución a la problemática.

La descripción de las HU divididas por iteraciones y la planificación del esfuerzo dedicado al desarrollo en cada una de ellas, logró una mejor organización del trabajo y el establecimiento de fechas para la culminación por cada una de las iteraciones.

El empleo de patrones de diseño hizo posible la escalabilidad en el componente desarrollado teniendo en cuenta su capacidad para reaccionar y adaptarse a cambios futuros.

Las tareas de ingeniería correspondientes a cada HU permitieron la organización del trabajo en una secuencia lógica de pasos.

El estándar de codificación utilizado proporcionó un buen entendimiento y una mejor estructuración del código.

Capítulo 3. Validación de la propuesta de solución

Partiendo de lo diseñado en el capítulo 2, se llega a la codificación del sistema. Para la realización de la validación del software con el objetivo de comprobar que funciona según lo diseñado y que las funcionalidades se implementaron de forma adecuada.

3.1 Fase de pruebas

El ciclo de vida de desarrollo de software describe las fases para el desarrollo de software, desde su fase inicial hasta su implementación y mantenimiento. De forma similar, el proceso de pruebas de software describe la manera de cómo va a ser probado el sistema de software. (Chillán Zulca, et al., 2017) Las pruebas de software son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto. (Pressman, et al., 2015)

Llevar a cabo un adecuado proceso de pruebas de software permite identificar y corregir los defectos encontrados a lo largo del proceso de desarrollo de software, garantizando un alto grado de la calidad del producto. (Chillán Zulca, et al., 2017) Cuando se realiza el proceso de pruebas el programa es ejecutado con datos de ejemplo cumpliendo con los tipos de datos de cada campo. Es necesario verificar los resultados de la prueba que se ejecuta para buscar errores, anomalías o información de atributos no funcionales del programa. El proceso de prueba tiene dos metas distintas: demostrar al desarrollador y al cliente que el software cumple con los requerimientos y encontrar situaciones donde el comportamiento del software sea incorrecto, indeseable o no esté de acuerdo con su especificación (Sommerville, 2011). XP divide las pruebas en dos grupos: pruebas de aceptación, o pruebas funcionales diseñadas por el cliente final, destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida y pruebas unitarias, encargadas de verificar el código, diseñadas por los programadores. (Analysis of linear sequential and extreme programming development methodology for a gaming application., 2016) (Integrating design thinking into extreme programming, 2019)

3.2 Niveles de prueba

El proceso de realización de pruebas está compuesto por una serie de niveles entre los que se pueden encontrar: el nivel de pruebas unitarias, el nivel de pruebas de integración, el nivel de pruebas de sistema y nivel de pruebas de aceptación. (Pressman, 2007)

Uno de los pilares de la metodología XP es el proceso de pruebas. XP anima a probar constantemente tanto y como sea posible. Esto posibilita aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. (Gutiérrez, et al., 2006) Terminada la implementación el sistema fue sometido a los niveles de prueba que son detallados a continuación, los que propiciaron la detección de los errores existentes.

3.2.1 Pruebas de aceptación

Las pruebas de aceptación XP, también llamadas pruebas del cliente, son especificadas por este y se centran en las características y funcionalidades generales del sistema que son visibles y revisables por parte del cliente. Las pruebas de aceptación se derivan de las historias de los usuarios que se han implementado como parte de la liberación del software. (Pressman, 2007)

Las pruebas de aceptación son creadas en base a las HU, en los ciclos de la iteración del desarrollo y son la representación de la satisfacción del cliente con el producto que se le entrega y es este el principal encargado de verificar su cumplimiento. Una HU no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. A continuación, son mostrados los casos de pruebas correspondientes a Autenticar usuario y a Consultar Ficha Manual.

Tabla 16 Caso de prueba Autenticar usuario. Fuente: (Elaboración propia)

Escenario	Descripción	Variable usuario	Variable Contraseña	Respuesta del sistema	Flujo central
Autenticar usuario.	Autenticar los usuarios de sistema.	V Susel	V Susel123	Accede a la interfaz principal del sistema.	Acceder a la herramienta. Llenar los campos de texto
		V Susel	I Susel45	Muestra la notificación, usuario o contraseña incorrectos.	usuario y contraseña. Seleccionar el botón aceptar.
		I	V Susel123	Muestra la notificación, complete	

				los campos.	
		V Susel	I	Muestra la notificación, complete los campos	
		I	I	Muestra la notificación, complete los campos.	

Tabla 17 Caso de prueba Consultar Ficha Manual. Fuente: (Elaboración propia)

Escenario	Descripción	Variable Campo de búsqueda	Respuesta del sistema	Flujo central
Consultar Ficha Manual.	Consultar diferentes Fichas Manuales de intoxicaciones.	V paracetamol	Accede a la interfaz de la Ficha Manual correspondiente.	Accede a interfaz principal. Llena el campo de búsqueda y selecciona el botón Buscar.
		I	Muestra la notificación complete el campo.	
		I mesa	Muestra la notificación no existen coincidencias.	

3.2.2 Pruebas unitarias

Las pruebas unitarias son una de las piedras angulares de la metodología XP, son realizadas para controlar el funcionamiento de pequeñas porciones de código. (Malforá, et al., 2006) Es recomendado

que estas pruebas sean automatizadas, pues esto facilita al programador identificar funciones que no ofrecen una salida acorde con la lógica que es deseada implementar.

Métodos de pruebas

Las pruebas de caja negra se refieren a las pruebas que son llevadas a cabo en la interfaz del software. Una prueba de caja negra revisa algunos aspectos fundamentales de un sistema con poca preocupación por la estructura lógica interna del software. Las pruebas de caja blanca se basan en el examen cercano de los detalles del procedimiento. Las rutas lógicas a través del software y las colaboraciones entre componentes se ponen a prueba al revisar conjuntos específicos de condiciones y/o bucles. (Pressman, 2007)

Pruebas de caja negra

El método de pruebas de caja negra, también llamado pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. (Pressman, et al., 2015)

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes (Pressman, 2007):

- Funciones incorrectas o faltantes.
- Errores de interfaz.
- Errores en las estructuras de datos o en el acceso a las bases de datos externas.
- Errores de comportamiento o rendimiento.
- Errores de inicialización y terminación.

Pruebas de caja blanca

El método de prueba de caja blanca, es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. (Pressman, et al., 2015)

Con la utilización de los métodos de caja blanca se pueden derivar casos de prueba que (Pressman, 2007):

- Garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez.

- Revisen todas las decisiones lógicas en sus lados verdadero y falso.
- Ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas.
- Revisen estructuras de datos internas para garantizar su validez.

Resultados de pruebas de caja negra

Mediante el método de caja negra y con el diseño de los casos de prueba se realizaron cuatro iteraciones de pruebas que pertenecen al nivel de aceptación. Estas pruebas fueron realizadas con el objetivo de corregir errores que impidieran el correcto funcionamiento de la solución.

Para la evaluación de la solución fueron realizadas tres iteraciones donde se probó el software de manera íntegra, finalizando con una cuarta iteración donde fue comprobada la resolución de todas las no conformidades detectadas. Fueron encontradas en dichas pruebas en la primera iteración un total de seis no conformidades, de las que fueron todas resueltas; en la segunda iteración fueron detectadas cinco no conformidades, de las que nuevamente todas fueron resueltas; en la tercera iteración se encontraron dos no conformidades, de las que fueron todas resueltas y en la cuarta iteración no fueron detectadas no conformidades, de esta forma quedan resueltas todas las no conformidades encontradas durante las pruebas.

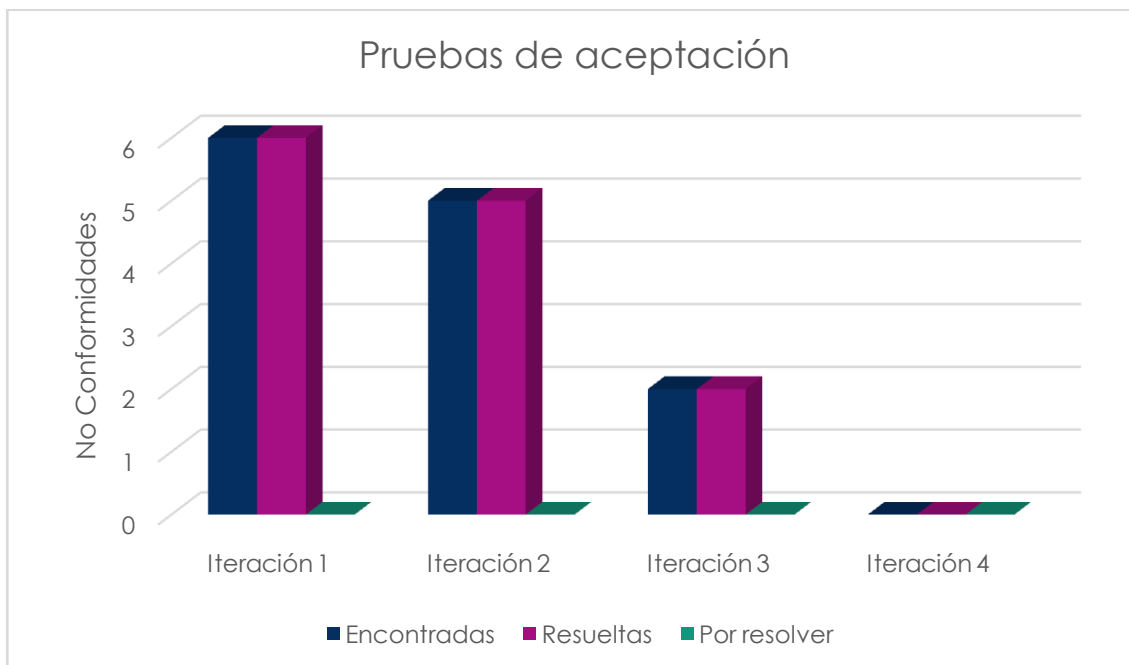


Figura 4 Pruebas de aceptación. Fuente: (Elaboración propia)

Resultados de pruebas de caja blanca

En total fueron implementadas ocho pruebas unitarias, de ellas tres presentaron no conformidades y al concluir la iteración se corrigieron. Los resultados se pueden apreciar en el siguiente gráfico.

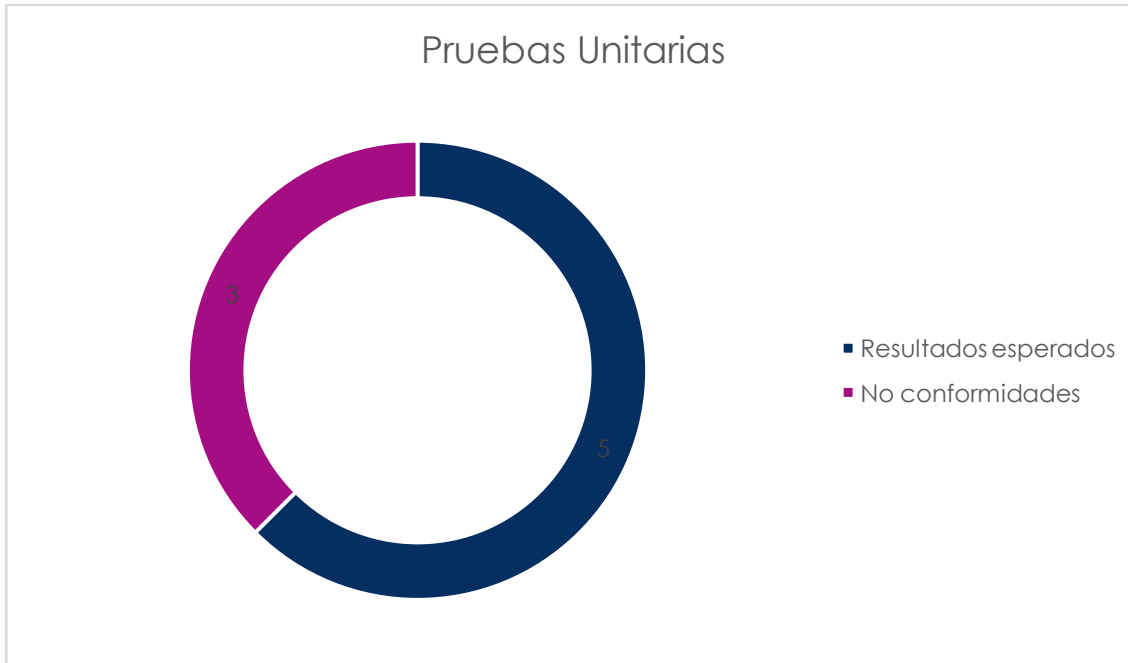


Figura 5 Pruebas unitarias. Fuente: (Elaboración propia)

3.3 Conclusiones del capítulo

Con la realización de las pruebas de aceptación y las pruebas unitarias se pudo detectar, documentar y corregir las no conformidades existentes en la herramienta; además permitieron verificar el correcto funcionamiento del sistema y el cumplimiento de los requisitos del cliente.

Conclusiones

El análisis de los softwares y herramientas identificados propició asegurar que éstos no cubren con las necesidades descritas en la problemática que se relacionan con el análisis de datos acerca de los casos de intoxicaciones.

El empleo de las metodologías, herramientas y tecnologías viabilizaron que la implementación de la herramienta se realizara exitosamente, cumpliendo así con objetivo general.

Mediante el diseño e implementación de las funcionalidades propuestas fue obtenida una herramienta informática que responde a las necesidades definidas del cliente durante la etapa de análisis.

La ejecución de las pruebas, permitió detectar y corregir deficiencias presentes en la herramienta, demostrando así que esta cumple con los requisitos definidos, garantizando su correcto funcionamiento.

Recomendaciones

Luego de haber analizado los resultados del presente trabajo de diploma, se recomienda:

- Incorporar funcionalidades que respondan a los demás procesos que se realizan en el CENATOX, como el proceso de obtención de datos del paciente y el solicitante.
- Actualizar periódicamente la información toxicológica que brinda el software por el personal del grupo de Respuesta Médica.

Bibliografía

Agile Processes in Software Engineering and Extreme Programming: 20th International Conference. **Kruchten, , Fraser, and Coallier, . 2019.** Montréal, QC, Canada : s.n., 2019. XP 2019.

Analysis of linear sequential and extreme programming development methodology for a gaming application. **Sharma, and Hasteer, . 2016.** 2016. 2016 Internacional Conference on Communication and Signal Processing (IPCCSP). pp. 1916-1920.

Ángel Cobo, Patricia Gómez, Daniel Pérez, Docío Rocha. 2005. *PHP y MySQL tecnologías para el desarrollo web.* Madrid : s.n., 2005. 84-7978-706-6.

2017. *ApunteRUP.doc.* 2017.

Cabrera, Jennifer Champagne y Marina. 2019. Capterra. [Online] 6 febrero 2019. [Cited: 10 enero 2020.] www.capterra.es.

2020. Cenatox. [Online] 20 agosto 2020. [Cited: 20 agosto 2020.] <http://www.cenatox.sld.cu>.

Chillán Zulca, Miriam Soraya and Lozano Pushug, Nelly Marisol. 2017. *Elaboración de una guía de procedimientos para la fase de pruebas en el desarrollo de software.* 2017.

Consideraciones generales sobre drogas de abuso. **Pascual Simón, J R and Fernández Rodríguez, B L. 2002.** 4, 2002, MEDISAN, Vol. 6.

2019. CropLife Latin America. [Online] 2019.

Desarrollo ágil de software aplicando programación extrema. **Gómez, Aveiro Rosado, Duarte, Alexander Quintero and Guevara, Cesar Daniel Meneses. 2014.** 1, 10 febrero 2014, Revista Ingenio Universidad Francisco de Paula Santander Ocaña, Vol. 5, pp. 24-29.

1998. Directrices para la lucha contra las intoxicaciones. Organización Mundial de la Salud. Ginebra: OMS : s.n., 1998. pp. 4-19.

Dr. C. My. Fleites Mestres, Pedro R. 2019. *Entrevista en el CENATOX.* La Habana: CENATOX, octubre 2019.

Dra García Rosales, . 2020. Centro Nacional de Información de Ciencias Médicas. [Online] 2020. [Cited: 26 agosto 2020.] Infomed.

Dra Rodríguez Rubinos, , et al. 2007. *Caracterización clínica epidemiológica de las intoxicaciones agudas en pediatría.* 2007.

1998. *El Programa Internacional de Seguridad de las Sustancias Químicas (IPCS).* Organización Mundial de la Salud. Ginebra: OMS : s.n., 1998. pp. 3-21.

Extreme programming implementation in academia for software engineering sustainability. In: 2018 Advances in Science and Engineering Technology International Conferences (ASET). **Sadath, , Karim, and Gill, . 2018.** 2018. pp. 1-6.

- Fernandez, . 2015.** *hostinger.es*. [Online] 2019 octubre 2015. [Cited: 3 febrero 2020.] <https://www.hostinger.es/tutoriales/que-es-mysql/>.
- Fraga Álvarez, and Pérez Ojeda, . 1996.** *Breve Reseña Histórica acerca del desarrollo de la Toxicología en Cuba*. La Habana: CENATOX : s.n., 1996.
- González, . 2018.** *dinahosting*. [Online] 17 agosto 2018. [Cited: 10 enero 2020.] www.dinahosting.com.
- Gonzalez, . 2016.** *Programacion.net*. [Online] 12 junio 2016. [Cited: 4 febrero 2020.] https://programacion.net/articulo/por_que_elegir_php_143.
- Gutiérrez, J J and al, . 2006.** *Pruebas del Sistema en Programación Extrema. Departamento de Lenguajes y Sistemas Informáticos*. Universidad de Sevilla : s.n., 2006.
- Informatización en el Sistema Nacional de Salud. Enfoques hacia la dirección en salud. Díaz, A R, González, G G and Aguiar, K B. 2013.* 16, 2013, Revista de Información para la Dirección en salud, Vol. 9.
- 2005.** *Ingeniería del software. Sistemas Informáticos. Nivel de madurez software. Informática Aplicada a la Gestión Pública*. Universidad de Murcia Rafael Barzanallana. 2005. Apunte.
- Integrating design thinking into extreme programming. Sohaib, , et al. 2019.* 6, 1 june 2019, Journal of Ambient Intelligence and Humanized Computing, Vol. 10, pp. 2485-2492.
- Jiménez Rangel, . 2006.** *Propuesta de diseño para la creación del servicio de toxicología Hospital "Pastor Oropeza Riera" IVSS*. Barquisimeto: Universidad Central de los Andes : s.n., 2006.
- Joskowicz, . 2008.** *Reglas y prácticas en eXtreme Programming*. Universidad de Vigo. 2008. vol. 22.
- La Biblioteca Virtual de Salud-Toxicología. D, González Machín. 1999.* Lima: CEPIS/OPS : s.n., 1999, pp. 2-10.
- La evolución de la Toxicología: de los venenos a la evaluación de riesgos. Vallverdú, . 2005.* 3, Pamplona, España : s.n., 27 abril 2005, Revista de Toxicología, Vol. 22, pp. 153-161. 0212-7113.
- La gestión de información como herramienta fundamental en el desarrollo de los centros tecnológicos. Capote Marrero, , González Machín, and Rodríguez Durán, . 2003.* 2, 2003, ACIMED, Vol. 11.
- Letelier, and Penadés, M C. 2006.** *Metodologías Ágiles para el desarrollo de software: eXtreme Programming (XP)*. 2006.
- Malforá, and al, . 2006.** *Testing en eXtreme Programing*. 2006.
- Mesa Cisneros, Víctor Alexis and Miranda Hernández, Manuel Alejandro. 2018.** *Plaguitox: Software para la gestión de información de los casos de intoxicaciones por plaguicidas en el Cenatox*. La Habana : s.n., 2018.
- Mortalidad por intoxicaciones agudas producidas con medicamentos. Cuba, 1995-1996. González, María Luisa Valiente, et al. 2000.* 1, 2000, Rev Cubana Farm, Vol. 34, pp. 25-33.
- Moya Díaz, and Villariel, . 2003.** *Síntesis del desarrollo histórico del Centro Nacional de Toxicología*. La Habana: CENATOX : s.n., 2003.

- Nuevas tecnologías de la información: problemas éticos fundamentales.* **2005.** 2005, Revista Acimed.
- Origen e historia de la Toxicología.* **Dra. Liudmila Pérez Barly, Dr. Jurek Guirola Fuentes, Dr. Pedro Fleites. 2014.** 4, Matanzas, Cuba : s.n., 19 junio 2014, Revista Cubana de Medicina Militar, Vol. 43, pp. 499-514.
- Patrón modelo-vista-controlador.* **Romero Fernández, and González Díaz, . 2014.** 2014, Revista Telem@tica, pp. 47-57.
- Pérez Barly, . 2013.** *Propuesta de un sistema automatizado de información toxicológica de urgencia.* La Habana : s.n., 2013.
- Pérez Pérez, O F, et al. 2010.** *Sinopsis histórica de la toxicología.* La Habana: CENATOX : s.n., 2010. Tesis de maestría.
- Pressman, . 2007.** *Ingeniería de Software. Un enfoque práctico.* 2007.
- Pressman, R S and Bruce, R Maxim. 2015.** *Software Engineering: A practitioner's approach 8th.* 8th. 2015. McGraw-Hill Education. ISBN 978-0-07-802212-8.
- Programación Extrema: Prácticas, Aceptación y Controversia.* **Campos, Saúl Gonzalez and Martínez, Luis Felipe Fernández. 2015.** 15, 16 junio 2015, Cultura Científica y Tecnológica, Vol. 0.
- Solucionex. 2009.** solucionex. [Online] 30 julio 2009. [Cited: 4 febrero 2020.] <https://www.solucionex.com/blog/por-que-utilizar-symfony>.
- Sommerville, . 2015.** *Software Engineering.* 10th. 2015. Pearson Education. ISBN 978-1-292-09613-1.
- Sommerville, . 2011.** *Software engineering 9th Edition.* 9th. 2011. Addison Wesley. ISBN 10 137035152.
- Suárez Escandón, . 2007.** *Valoración de le enseñanza de la Toxicología en los estudios de Medicina en Emergencias.* La Habana: Facultad de Ciencias Médicas "Miguel Enríquez" : s.n., 2007.
- Tabares, Ricardo Botero. 2014.** *Patrones Grasp y Anti-Patrones: un Enfoque Orientado a objetos desde Lógica de Programación.s.l: Entre Ciencia e Ingeniería.* 2014.
- Villegas, Adrian Anaya. 2009.** *Monografías.* 2009.

Anexos

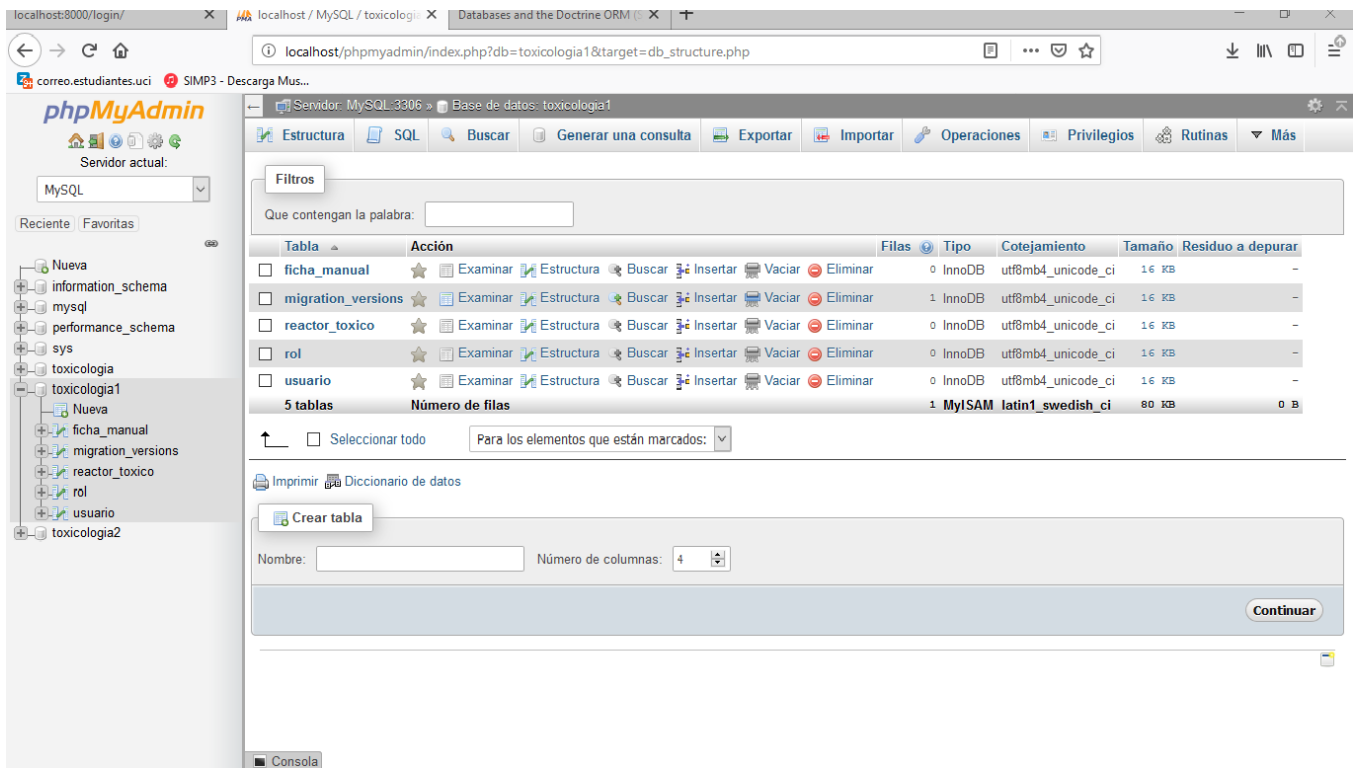


Figura 6 Sistema de tablas de la herramienta. Fuente: (Elaboración propia)

Tabla 18 Descripción de la HU 1 Gestionar usuario. Fuente: (Elaboración propia)

Historia de Usuario	
Número: 1	Nombre de HU: Gestionar usuario
Referencia: ninguna	Prioridad: media
Iteración Asignada: 1	Puntos Estimados: 1.5
Descripción: El software debe permitir gestionar los usuarios deseados.	
Observaciones: ninguna	

Tabla 19 Descripción de la HU 2 Autenticar usuario. Fuente: (Elaboración propia)

Historia de Usuario	
Número: 2	Nombre de HU: Autenticar usuario
Referencia: Gestionar usuario	Prioridad: media
Iteración Asignada: 1	Puntos Estimados: 1
Descripción: El software debe permitir autenticar usuario	
Observaciones: ninguna	

Tabla 20 Descripción de la HU 3 Autocompletar consulta. Fuente: (Elaboración propia)

Historia de Usuario	
Número: 3	Nombre de HU: Autocompletar consulta

Referencia: Autenticar usuario	Prioridad: media
Iteración Asignada: 1	Puntos Estimados: 1.5
Descripción: El software debe autocompletar la consulta de la Ficha Manual.	
Observaciones: La autocompletación se realiza con los nombres genéricos y los otros nombres de las Fichas Manuales.	

Tabla 21 Descripción de la HU 6 Exportar a PDF. Fuente: (Elaboración propia)

Historia de Usuario	
Número: 6	Nombre de HU: Exportar a PDF
Referencia: Visualizar Ficha Manual	Prioridad: alta
Iteración Asignada: 3	Puntos Estimados: 1.5
Descripción: El software debe permitir exportar a PDF la Ficha Manual consultada.	
Observaciones: ninguna	