

**Universidad de las Ciencias Informáticas**

**Facultad 3**



**Módulo de Biopsias para el sistema informático de registro y control en Anatomía Patológica.**

Trabajo de Diploma para optar por el título de Ingeniera en Ciencias Informáticas

**Autor:**

Rachel Avila Milord.

**Tutores:**

Ing. Pedro Arango Astorga

MSc. Mailen Edith Escobar Pompa

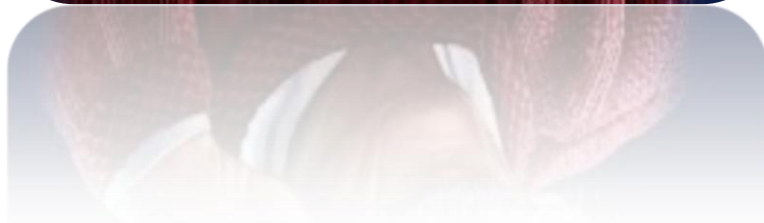
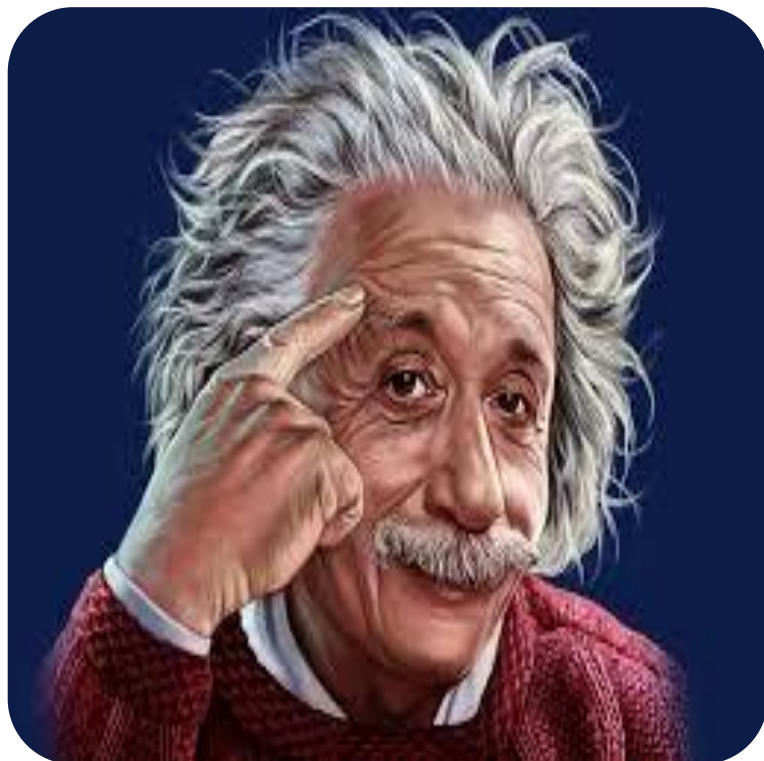
**Asesor:**

Dr. Cs. José Hurtado De Mendoza Amat

La Habana, agosto de 2020

“Año 62 de la Revolución”

[Type here]



*"No necesito saberlo todo, tan sólo necesito saber donde encontrar aquello que me hace falta, cuando lo necesite."*

*Albert Einstein.*

## **DECLARACIÓN DE AUTORÍA**

Declaro ser la única autora del trabajo de diploma con título “Módulo de Biopsias para el sistema informático de registro y control en Anatomía Patológica”y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación con carácter exclusivo.

---

Firma del Autor

**Rachel Avila Milord**

---

Firma del Tutor

**Ing. Pedro Arango Astorga**

---

Firma del Tutor

**MSc. Mailen Edith Escobar Pompa**

## **Agradecimientos**

**Al concluir una etapa maravillosa de mi vida quiero extender un profundo agradecimiento, a quienes hicieron posible este sueño, aquellos que junto a mi caminaron en todo momento y siempre fueron una inspiración, apoyo y fortaleza. Quisiera empezar agradeciendo a la persona más importante en mi vida mi motor a seguir, mi madre, eres una mujer que simplemente me hace llenar de orgullo, te amo y no va haber manera de devolverte tanto que me has ofrecido desde que incluso antes de haber nacido. Está tesis es un logro más que llevo a cabo, y sin lugar a dudas ha sido en gran parte gracias a ti; no sé en donde me encontraría de no ser por tus ayudas, tu compañía y amor.**

**A mi padre por siempre desear y anhelar lo mejor para mi vida, por cada día confiar y creer en mí y en mis expectativas, gracias por sus consejos y por cada una de sus palabras que me guiaron toda mi vida.**

**La vida no me alcanzará para agradecerle todo lo que han hecho por mí, el amor recibido, la dedicación y paciencia con la que cada día se preocupaban por mi avance y desarrollo de esta tesis, es simplemente único y se refleja en la vida de un hijo.**

**A mi hermano por brindarme su apoyo incondicional y siempre estar presente en los momentos más importantes.**

**A mi abuela Beba por brindarme su apoyo absoluto, por tus enseñanzas, por los mensajes de aliento y tu excelente manera de instruirme para afrontar las verdades de esta vida. En este reto universitario fuiste igualmente concluyente. Te doy mis más sinceras gracias abuela te quiero mucho.**

**A mis abuelos, aunque no se encuentran en esta vida, sé que estarían muy orgullosos de haberme hecho una ingeniera.**

**A mis tías, tíos y primos por compartir momentos significativos conmigo y por siempre estar dispuestos a escucharme y ayudarme en cualquier momento.**

**A mis amigas y amigos Beatriz, Arletys, Sindia, Yi, Raydel, Leo, Orly, Kkchi por brindarme su apoyo incondicional y convertirse en una gran familia a lo largo de esta vida universitaria. A los que han estado a lo largo de estos años brindándome lo mejor de ellos y me han demostrado que puedo contar con ellos cuando más lo he necesitado.**

**A mis tutores por su paciencia y dedicación, gran parte de este logro se los debo a ellos.**

**A todos los profesores que a lo largo de la carrera contribuyeron en mi formación personal y profesional.**

## **DEDICATORIA**

**Dedico esta tesis a mis padres, por haberme forjado como la persona que soy en la actualidad y que siempre me apoyaron incondicionalmente en la parte moral y económica para poder ser una profesional.**

**A mi hermano que lo quiero con todo mi corazón, por brindarme su tiempo y su apoyo absoluto.**

**A mis abuelas, tías, tíos y primos y demás familia en general por el apoyo que siempre me brindaron día a día en el transcurso de cada año de mi carrera universitaria.**

**A todos mis amigos por permitirme aprender más de la vida a su lado. Esto es posible gracias a ustedes y a todas aquellas personas que estuvieron apoyándome y aportando su granito de arena para que este gran día se hiciera realidad.**

## **RESUMEN**

Los resultados de la biopsia generan múltiples beneficios mayormente orientados a mejorar la calidad del trabajo médico. Su inmenso beneficio se sustenta en la gestión eficiente de la información generada, factible con la incorporación de las nuevas tecnologías de la información y las comunicaciones. En Cuba se realiza un número elevado de estudios biopsicos, sin embargo, el sistema informático usado en los departamentos de Anatomía Patológica no favorece la utilización efectiva de la información generada durante el estudio de las muestras histológicas. El objetivo de este trabajo es desarrollar un módulo de biopsia que permita la gestión de la información de las biopsias. Para una mejor especificación de los diagnósticos, el sistema promueve su codificación según los más recientes estándares de terminología la CIE-10 impulsados por la Organización Mundial de la Salud. Además, posibilita la configuración y aplicación de variadas reglas de control interno sobre los diagnósticos de la enfermedad, así como la creación de un amplio conjunto de reportes estadísticos sobre los datos, los cuales podrán ser empleados como soporte para la toma de decisiones. La solución desarrollada, facilita el proceso de evaluación de la calidad en la atención médica, a partir del análisis y clasificación de la correlación clinicopatológica, así como la generación de un informe de biopsia de mayor calidad y precisión.

### **Palabras claves**

Anatomía Patológica, Biopsia, Morbilidad.

# ÍNDICE

INTRODUCCIÓN .....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN .....	6
1.1. DESCRIPCIÓN GENERAL DEL OBJETO DE ESTUDIO .....	6
1.1.1 SITUACIÓN ACTUAL Y PERSPECTIVAS DE LA BIOPSIA EN CUBA Y EL MUNDO .....	7
1.1.2 LA GESTIÓN DE LA INFORMACIÓN EN BIOPSIAS.....	9
1.2 ANÁLISIS DE LAS SOLUCIONES EXISTENTES.....	12
1.2.1 SARCAP.....	13
1.2.2 AvanPat.....	14
1.2.3 ALAS HIS .....	15
1.2.4 SIRCAP .....	16
1.3 METODOLOGÍA, LENGUAJES Y HERRAMIENTAS DE DESARROLLO .....	17
1.3.1 METODOLOGÍA DE DESARROLLO.....	17
1.3.2 LENGUAJE DE MODELADO .....	17
1.3.3 INGENIERÍA DE SOFTWARE ASISTIDA POR COMPUTADORA (CASE) .....	18
1.3.4 LENGUAJE DE PROGRAMACIÓN .....	18
1.3.5 ENTORNO DE DESARROLLO INTEGRADO .....	19
1.3.6 CONTROL DE VERSIONES .....	19
1.3.7 PROTOTIPOS DE INTERFACES DE USUARIO.....	19
1.3.8 HERRAMIENTA PARA LA GENERACIÓN DE REPORTE.....	20
1.3.9 SISTEMA GESTOR DE BASES DE DATOS .....	20
1.3.10 MAPEO OBJETO-RELACIONAL .....	20
1.4 PATRONES PARA EL DESARROLLO DE SOFTWARE .....	21
1.5 ESTRATEGIA DE PRUEBAS .....	22
1.6 CONCLUSIONES PARCIALES.....	24
CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN.....	25
2.1 . MODELO CONCEPTUAL .....	25
2.2 . INGENIERÍA DE REQUISITOS DEL SISTEMA .....	26
2.2.1. OBTENCIÓN DE REQUISITOS.....	26
2.2.2. REQUISITOS FUNCIONALES DEL SISTEMA.....	28



2.2.3. REQUISITOS NO FUNCIONALES DEL SISTEMA.....	29
2.2.4. ESPECIFICACIÓN DE REQUISITOS FUNCIONALES .....	30
2.2.5. VALIDACIÓN DE REQUISITOS FUNCIONALES .....	33
2.3 . PLAN DE ITERACIONES.....	34
2.4 . ARQUITECTURA DE LA SOLUCIÓN .....	35
2.5. MODELO DE DISEÑO.....	36
2.6. PATRONES DE DISEÑO .....	37
2.6.1. PATRÓN DAO .....	37
2.6.2. PATRONES GOF .....	37
2.6.3. PATRÓN GRASP.....	38
2.7. MODELO DE DATOS .....	39
2.8. CONCLUSIONES PARCIALES.....	42
<b>CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS.....</b>	<b>43</b>
3.1. ESTÁNDARES DE CODIFICACIÓN.....	43
3.2. PRUEBAS DE UNIDAD.....	45
3.3. CONCLUSIONES PARCIALES.....	50
<b>CONCLUSIONES .....</b>	<b>51</b>
<b>RECOMENDACIONES .....</b>	<b>52</b>
<b>ANEXOS.....</b>	<b>57</b>

## ÍNDICE DE TABLAS

<b>Tabla 1:</b> Características de los sistemas más conocidos con soporte para la gestión de Anatomía Patológica Internacionales. ....	12
<b>Tabla 2:</b> Características de los sistemas más conocidos con soporte para la gestión de Anatomía Patológica Cubanos.....	16
<b>Tabla 3:</b> Descripción de los Conceptos del Modelo Conceptual.....	26
<b>Tabla 4:</b> Historia de usuario: Registrar biopsia.....	31
<b>Tabla 5:</b> Historia de usuario: Buscar paciente. ....	32
<b>Tabla 6:</b> Registrar paciente. ....	33
<b>Tabla 7:</b> Esfuerzo estimado por iteraciones.....	34
<b>Tabla 8:</b> Tarjeta CRC: BuscarPersonaController.java. ....	36
<b>Tabla 9:</b> Tarjeta CRC: ProtocoloController.java. ....	36
<b>Tabla 10:</b> Tarjeta CRC: RegistrarPacienteController.java.....	37
<b>Tabla 11:</b> Caso de prueba de aceptación: HU Adicionar biopsia. ....	47
<b>Tabla 12:</b> Juego de datos de prueba: HU Registrar autopsia (V: válido, I: inválido).....	49

## ÍNDICE DE FIGURAS

<b>Figura 1:</b> Modelo Conceptual.....	25
<b>Figura 2:</b> Modelo Vista Controlador. Fuente: (Arango 2014). ....	35
<b>Figura 3:</b> Patrón árbol fuertemente codificado. ....	40
<b>Figura 4:</b> Patrón árbol simple.....	40
<b>Figura 5:</b> Patrón llave subrogada. ....	41
<b>Figura 6:</b> Patrón control de acceso basado en roles. ....	41
<b>Figura 7:</b> Cabecera de archivos. ....	43
<b>Figura 8:</b> Definición de clases.....	44
<b>Figura 9:</b> Definición de variables y constantes.....	44
<b>Figura 10:</b> Definición de funciones. ....	44
<b>Figura 11:</b> Uso de llaves en bloques de instrucciones. ....	44
<b>Figura 12:</b> Posición de las llaves en bloque de instrucciones. ....	45
<b>Figura 13:</b> Espacio entre operadores.....	45
<b>Figura 14:</b> Resultados de las pruebas de unidad aplicadas a la clase ControlCalidadServicios.java. ....	46

## INTRODUCCIÓN

El objetivo final de un sistema de salud es potenciar y mejorar en calidad la vida del hombre (Amat, 2014). La atención primaria de salud tiene como principal función conocer el estado de salud o enfermedad de un individuo o una colectividad. Dentro de la atención primaria los exámenes auxiliares o complementarios tienen un papel muy importante y muchas veces decisivo en el diagnóstico de las enfermedades. Estos proporcionan datos clínicos sobre las condiciones internas del paciente, condiciones que de otro modo son inobservables. El gran valor de los exámenes complementarios está dado, entre otras cosas, por su utilidad para confirmar diagnósticos y rechazar otros. Por medio de estos procedimientos se puede observar y conocer más directamente acerca del estado anatómico, metabólico y funcional de los órganos y tejidos (Rodríguez Moreno, 2014).

La biopsia, (del griego bios vida, opsia observar), es un proceso mediante el cual se extrae un fragmento de un ser vivo, con el objetivo de realizar un estudio morfológico y estructural, llegando a conclusiones precisas de un diagnóstico (Cirión Martínez & Herrera Pérez, 2005), (Hidalgo Ríos & Ochoa Asse, 2014). Es un tipo de análisis de la atención primaria que se realiza para profundizar en diferentes parámetros ya que permite establecer, confirmar y modificar los diagnósticos clínicos o quirúrgicos, resultando imprescindible la utilización de la misma en los servicios médicos que se brindan a los pacientes (Amat, 2014). La biopsia constituye un paso básico en el diagnóstico y tratamiento de muchas enfermedades y es indispensable en las afecciones tumorales. Aunque la toma de material para la biopsia suele considerarse como un procedimiento de cirugía menor, y en realidad lo es en muchas ocasiones, la importancia de este paso supera su magnitud como procedimiento quirúrgico, pues de la precisión de la biopsia puede depender todo el tratamiento subsiguiente y aun la vida del paciente (Hardy, Griffin & Rodríguez, 1961), (Hidalgo Ríos & Ochoa Asse, 2014).

El objetivo fundamental de la biopsia es llegar al diagnóstico cierto del proceso o enfermedad que se estudia, determinando la respuesta o evolución de una enfermedad. Es indicada siempre que existe una lesión o enfermedad que necesite de un diagnóstico definitivo, para poder tomar una decisión adecuada, es decir, si se trata de una lesión de tipo inflamatoria, neoplásica benigna o maligna entre otras, lo que facilitará en muchos casos la conducta clínica o quirúrgica a seguir (Hardy, Griffin & Rodríguez, 1961).

La biopsia es de gran valor, al permitir llegar a un diagnóstico en el mayor porcentaje de los casos, en el transcurso de la enfermedad, determinando con gran exactitud las causas que la producen. Facilita conocer si existe extensión tumoral a determinado límite de un tejido supuestamente sano. Resulta uno de los métodos más rigurosos y confiables, constituyendo un arma poderosa para beneficio del enfermo,

siempre que se cumpla con la calidad y representatividad de la toma de la muestra, permitiendo tomar una conducta correcta desde el punto de vista terapéutico(Hardy, Griffin & Rodriguez, 1961).

Pero tantas oportunidades y beneficios se sustentan en un eficiente aprovechamiento de la información generada por la biopsia, la cual muchas veces se ve afectada por la insuficiente estructuración de los datos acopiados, incluso su olvido en archivos físicos no siempre bien conservados. En este sentido, la utilización ordenada y masiva de las tecnologías de la información y las comunicaciones (TIC) juega un papel esencial en la búsqueda del máximo aprovechamiento de los estudios biópsico.

Las últimas décadas se han visto marcadas por una tendencia creciente al desarrollo de soluciones informáticas para la gestión cada vez más eficiente de la información, en respuesta a la necesidad de conocimiento útil como soporte a la toma de decisiones. Hoy, la informática está presente en casi todos los sectores de la sociedad y la medicina no es la excepción. En este contexto, es posible hablar de Registros Informatizados de Salud, Sistemas de Información Hospitalarios e incluso soluciones departamentales específicas.

La gestión de la información generada en los departamentos de Anatomía Patológica ha motivado la aparición de diferentes sistemas de información en todo el mundo, resaltando países como España y Estados Unidos por el nivel de especialización alcanzada. Sin embargo, dada la naturaleza del entorno en que han sido creados, estos sistemas en su mayoría poseen características que los hacen inapropiados en las condiciones cubanas, evidenciando la necesidad de una alternativa de origen nacional.

En Cuba en el Hospital Militar Central Dr. Luis Díaz Soto se usa el Sistema Automatizado de Registro y Control en Anatomía Patológica (SARCAP), el cual fue desarrollado y empleado en Cuba desde el 1985. Este sistema presenta módulos para autopsia y biopsia, almacenando información de más de 100 000 biopsias realizadas en varios hospitales(Amat, 2019).

Sin embargo, dadas sus características, el SARCAP no ha podido ser actualizado con la última revisión de la Clasificación Internacional de Enfermedades (CIE-10) de la Organización Mundial de la Salud (OMS), sobre la cual se dan las estadísticas nacionales de las biopsias. La codificación actual de los diagnósticos resulta redundante en algunos casos e insuficiente en otros para lograr una especificación detallada de la enfermedad, especialmente cuando resulta conveniente complementarla con los ejes morfológico y topográfico, así como el uso de comodines. Además, no posibilita la codificación de procedimientos médicos por sí mismos(Amat, 2014).

Está desarrollado sobre un ambiente MS-DOS, que muchas veces causa predisposición en las personas y ha llegado a ser un obstáculo para la colaboración internacional. Además, presenta un conjunto de limitaciones reales asociadas a la plataforma en que fue desarrollada que se considera necesario resolver, como son:

- Dependiente de la plataforma.
- Interfaz poco amigable.
- Insuficiente validación de la entrada de datos.
- Necesidad de memorizar los códigos de los diagnósticos.
- Incapacidad para transmitir los datos de las biopsias realizadas en cada uno de los hospitales a la base de datos nacional.
- Limitados recursos de visualización de la información y las estadísticas generadas.

En la búsqueda de una solución factible, se formula el siguiente **problema a resolver**: El sistema actual de registro y control en Anatomía Patológica afecta el análisis de la información generada por el procesamiento histológico<sup>1</sup> de las muestras en la red hospitalaria de Cuba.

En concordancia con lo anterior, se toma como **objeto de estudio** la gestión de la información en los departamentos de Anatomía Patológica, centrando el **campo de acción** en el proceso de registro y control de biopsias en la red hospitalaria de Cuba.

En respuesta al problema, se define como **objetivo general** desarrollar un módulo para el sistema informático para el registro y control en Anatomía Patológica que favorezca el análisis de la información generada durante el procesamiento histológico de muestras. Para darle cumplimiento al objetivo general se definen los siguientes objetivos específicos:

1. Formalizar el marco teórico conceptual de la investigación a partir de los referentes teóricos que garanticen el máximo aprovechamiento de los estudios patológicos.
2. Obtener el modelo de diseño del módulo de biopsia para el sistema informático de registro y control en Anatomía Patológica.
3. Implementar el módulo de biopsias para el sistema informático de registro y control en Anatomía Patológica.
4. Obtener la validación de la solución propuesta mediante pruebas de calidad de software y pruebas de aceptación a expertos en Anatomía Patológica.

---

<sup>1</sup>Denominamos proceso histológico a una serie de métodos y técnicas utilizados para poder estudiar las características morfológicas y moleculares de los tejidos.

Para dar cumplimiento al objetivo, se deberán desarrollar las siguientes **tareas de investigación**:

1. Revisión de la bibliografía para la actualización de tendencias actuales en estudios patológicos referidos a las biopsias.
2. Determinación de la metodología, herramientas y tecnologías a utilizar para el desarrollo del sistema.
3. Definición de las principales funcionalidades del sistema.
4. Obtención del modelo de diseño del sistema.
5. Obtención del modelo de datos del sistema.
6. Implementación de las funcionalidades del sistema.
7. Validación de la efectividad de la solución propuesta a partir de pruebas de aceptación con expertos en Anatomía Patológica.

Para el desarrollo de la investigación se harán uso de los siguientes **métodos científicos**:

**Métodos Teóricos:**

- **Método de Análisis histórico-lógico:** para el estudio de la evolución y situación actual de la práctica de la biopsia a nivel mundial, y en Cuba de forma particular; así como la creciente necesidad de una gestión cada vez más eficiente de la información generada y las soluciones encontradas a este problema.
- **Analítico-sintético:** se utilizará en el estudio de los fundamentos teóricos, en el procesamiento de la información y en la descomposición de cada uno de los requerimientos del sistema informáticos para la gestión y procesamiento de la información en Anatomía Patológica, lo cual servirá como punto de partida para la solución propuesta.
- **Método de la Modelación:** para un mejor entendimiento de los procesos a informatizar. Se va a hacer mediante la confección de los artefactos que propone la metodología de software a utilizar.

**Método Empírico:**

- **La entrevista** juega un papel esencial en el desarrollo del presente Trabajo de Diploma, específicamente la entrevista no estructurada, dada su flexibilidad y capacidad de obtener conocimiento de un tema directamente de los especialistas. El empleo de la entrevista a especialistas en anatomía patológica permitirá identificar las necesidades reales de informatización en los departamentos de Anatomía Patológica de Cuba, así como el entendimiento de los procesos a informatizar.

Para un mejor entendimiento de este documento, a continuación, se describe brevemente cada uno de los capítulos en que se encuentra estructurado.

**CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN:** describe los principales conceptos asociados al dominio del problema, así como las relaciones que se establecen entre ellos. Expone un análisis de las soluciones en el mercado, resaltando las tendencias y las tecnologías más comunes en su desarrollo. Finalmente, se argumentan las bases del desarrollo del módulo en cuestión.

**CAPÍTULO 2: NEGOCIO, REQUISITOS, ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN:** refleja cada uno de los pilares del desarrollo del módulo propuesto, la especificación de sus funcionalidades, la arquitectura, los modelos de diseño y de datos y el uso de patrones.

**CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS:** refiere los estándares de codificación, la estrategia de pruebas definida a partir de la metodología de desarrollo, con el objetivo de garantizar la validación de la solución propuesta. Se analizan los niveles de pruebas ejecutados y los resultados alcanzados. Finalmente, se presentan evidencias de cómo la solución propuesta pueda contribuir al análisis de biopsias.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

El presente capítulo recoge los elementos teóricos que fundamentan el desarrollo de un Módulo de Biopsias para el sistema informático de registro y control en Anatomía Patológica. Se realiza un análisis de la situación actual y perspectivas de la biopsia en Cuba y el mundo, así como la importancia de una eficiente gestión de la información que esta genera. Además, se reflejan las características esenciales de las diferentes soluciones informáticas existentes en el mercado; dejando claro, de manera general, las tendencias actuales en el desarrollo de este tipo de sistemas y las tecnologías más comúnmente utilizadas. Finalmente, se exponen los elementos más relevantes de la metodología, lenguajes y herramientas definidas para el desarrollo; reflejándose el uso de patrones y pruebas para garantizar la efectividad del proceso de desarrollo y calidad del producto final.

### 1.1. DESCRIPCIÓN GENERAL DEL OBJETO DE ESTUDIO

La Patología es la rama de la ciencias naturales que se dedica al estudio de las enfermedades, a su vez, la Anatomía Patológica es la rama de la Patología que estudia las alteraciones morfológicas, macro y microscópicas, que producen las enfermedades en las células, tejidos u órganos (Amat, 2014).

Entre los métodos de estudio de Anatomía Patológica se encuentra la biopsia, la cual es uno de los métodos diagnósticos más riguroso y confiables. Sobre ella descansa la terapéutica a aplicar en los pacientes y sirve también a su vez para evaluar los resultados(Hidalgo Ríos & Ochoa Asse, 2014).

La biopsia, según(Pérez, 1958), es la operación exploratoria que consiste en separar de un ser vivo una muestra cualquiera de tejido u órgano, tanto en forma de porción orgánica como de elementos disgregados, para su examen macro y microscópico, con el propósito de determinar su naturaleza mediante una cita o citodiagnosisis(Amat, 2014).

Los tipos de biopsias se distinguen, fundamentalmente, por el modo en que se obtiene el fragmento de tejido:A continuación se describe los tipos de biopsias que existen según (Amat, 2014):

- 1 **Incisional.** Consiste en la extirpación de un fragmento de la lesión.
- 2 **Por aspiración.** Es la obtención de un cilindro de tejido por medio de un trócar que se introduce en el órgano afectado. Este tipo de biopsia es útil en órganos profundos o no accesibles, como el riñón, el hígado, el pulmón, la próstata, etc. Tiene riesgos en cuando a sangramientos y a veces no es representativa o suficiente para llegar al diagnóstico.
- 3 Una modalidad de esta, que hoy tiene gran importancia y utilidad es la **biopsia por, punción aspirativa con aguja fina (BAAF)**. Esta variante evita riesgos y molestias al paciente y puede realizarse ambulatoriamente. Es la más rápida y económica, y a pesar de ser un estudio citológico



su desarrollo ha permitido diagnósticos de gran precisión, por ello su realización se hace cada vez más extendida y necesaria.

- 4 **Excisional.** Es la extirpación de toda la lesión junto con un margen adecuado de tejidos periféricos sanos. Es el tipo de biopsia más recomendable, sobre todo en lesiones pequeñas y accesibles, las cuales puede ser estudiadas íntegramente y establecer sus relaciones con los tejidos vecinos.
- 5 **Transoperatoria o por congelación.** Es la biopsia que se realiza durante el acto quirúrgico mediante la congelación del tejido objeto de estudio, de modo que es posible llegar a un diagnóstico rápido para tomar decisión sobre el tratamiento que se ha de seguir. Este tipo de biopsia tiene una gran trascendencia e implica gran responsabilidad para el patólogo. Ante un resultado dudoso en este estudio, se deben diferir las conclusiones para el examen detallado con la técnica de inclusión en parafina, que tarda unos días más, pero da detalles más completos.
- 6 **Postoperatoria.** Son las biopsias de las piezas u órganos que se extirpan, con el objetivo de precisar el diagnóstico, determinar con certeza la extensión del proceso y si la operación fue suficiente, insuficiente o excesiva.
- 7 **Extendidos citológicos.** Comúnmente llamada "citología", es una modalidad de biopsia que consiste en el estudio solamente de muestras citológicas de órganos o tejidos. Por tanto, también incluyen la BAFF.

### 1.1.1 SITUACIÓN ACTUAL Y PERSPECTIVAS DE LA BIOPSIA EN CUBA Y EL MUNDO

En Cuba hay 100 departamentos de anatomía patológica en igual número de hospitales que abarcan todas las provincias, incluyendo el municipio especial de Isla de la Juventud, y alrededor de 400 patólogos. Anualmente se realizan en el país más de 300 mil biopsias y 700 mil exámenes citológicos. Estos últimos incluyen los estudios cérvico vaginales para el diagnóstico temprano del cáncer de cuello y las punciones aspirativa con agujas finas para analizar lesiones de ganglios linfáticos, tiroides, mama y partes blandas (Borrajero Martínez, 2018).

El cáncer de mama es el cáncer más frecuente en mujeres de países industrializados: cerca de un 30 por ciento de los casos. El fundamento de este análisis de sangre es un método llamado "biopsia líquida", no del todo una novedad. Un grupo de investigadores de Gran Bretaña, Dinamarca, Polonia, Holanda y España publicó ya en noviembre de 2018 un estudio en el que presentaron las múltiples posibilidades de la detección del cáncer a través de muestras de sangre. Con la biopsia líquida fueron capaces de identificar hasta ocho tipos diferentes de cáncer, si bien el porcentaje de aciertos aún era bastante bajo: solo se pudo identificar uno de cada tres casos de cáncer de mama. Ese resultado, en comparación con el de una mamografía, es insuficiente, ya que esta llega a un porcentaje de aciertos de cerca del 78 por ciento (Schmidt, 2019).

El cáncer de mama es la forma de cáncer más común a nivel mundial. En Latinoamérica alrededor de 114,900 mujeres son diagnosticada cada año y 37,000 mueren por esta enfermedad. Se estima que la mortalidad ajustada a la edad, en países como Colombia, Ecuador, México y Perú es aproximadamente 10 por cada 100,000 mujeres. La biopsia de mama con guía radiología es uno de los pilares del diagnóstico y toma de decisiones clínico quirúrgicos en malignidad mamaria. Su impacto en el diagnóstico temprano y manejo de esta patología ha sido significativo en las últimas décadas (Mena, Benavides & Freire Álvaro, 2017).

El cáncer de próstata viene mostrando un aumento sostenido en su incidencia en los países desarrollados, llegando el 2018 a ser el segundo cáncer más frecuente en los hombres luego del cáncer pulmonar en el mundo, ocupando el quinto lugar en mortalidad. El diagnóstico de la biopsia ha mostrado tener muchos errores ya que sobre-diagnostica cánceres no significativos y sub-diagnostica cánceres clínicamente significativos (Guzmán & Messina, 2019).

En los 10 años de estudio se hallaron 139 reportes de biopsias renales en el Hospital Nacional Dos de Mayor, Lima, Perú, siendo la glomerulonefritis focal y segmentaria<sup>2</sup> la causa más frecuente de glomerulopatía primaria<sup>3</sup> encontrada. Este diagnóstico se ordena con estudios de este país como lo publicado por Castillo-Zegarra, donde se encontró que fue el 33,7%, en el estudio realizado por (Valdez Yáñez, Hernández-Pacheco & Arellán-Bravo, 2018) el porcentaje alcanzó el 36%.

El cáncer como problema de salud mundial es, entre las principales enfermedades mortales, una de las pocas que presenta una tendencia ascendente. Según la Organización Mundial de la Salud (OMS), la incidencia del carcinoma de piel se ha triplicado en las últimas dos décadas. Es el más frecuente en el ser humano y además el responsable de la mitad de todas las neoplasias malignas en Estados Unidos.

Australia, con una incidencia de 726 casos por 100 000 habitantes es el país que presenta la mayor cantidad de enfermos a nivel mundial. La incidencia del Colegio Canadiense de Medicina Naturopática (CCNM) se incrementa del 3 % al 6 % cada año, mientras que en los Estados Unidos aumentó en un 200 % en los pasados treinta años y en la actualidad corresponde a 150 casos por 100 000 habitantes (Fernández Martori, 2018).

---

<sup>2</sup>La **glomeruloesclerosis focal y segmentaria** (GEFS) provoca cicatrización (esclerosis) en diferentes lugares a través del riñón. Por lo general, solo afecta a una parte de un glomérulo y solo a un pequeño número de glomérulos dentro de los lugares cicatrizados. El término "focal" significa que algunos de los glomérulos resultan cicatrizados, mientras que otros permanecen normales. El término "segmentaria" significa que solo parte de un glomérulo individual resulta dañado.

<sup>3</sup>Se define como **glomerulopatías** o enfermedad glomerular a desordenes que afectan la estructura y la función glomerular. Su espectro es muy amplio, y puede ser desde asintomática hasta enfermedades que ponen en riesgo la vida. Acorde a su etiología, las podemos clasificar en **primarias** cuando el daño inicial ocurre en el riñón y no se atribuye a una enfermedad sistémica.

Cuba también está en alerta, siendo esta localización la primera causa de morbilidad en el registro nacional con un reporte de 9785 casos en total. El estudio histopatológico de las muestras cutáneas es una de las herramientas diagnósticas más importante con las que cuenta el dermatólogo para confirmar la etiología neoplásica. El valor del resultado de la biopsia también permite decidir una conducta posterior. El informe debe contener datos que ayuden al especialista a definir el pronóstico. En el país la información registrada transita desde que se recoge el resultado del departamento de anatomía patológica y se procede a reportar el caso por parte del médico de asistencia hasta que el sistema estadístico procesa este dato que llega al registro de cáncer nacional. Cada Dirección Municipal de Salud debe manejar la estadística de sus unidades que luego ingresan a las Direcciones Provinciales y de ahí al Registro Nacional (Fernández Martori, 2018).

Todos estos estudios muestran la importancia que reviste la biopsia para la identificación de enfermedades, sobre todo cáncer, de ahí la importancia de una correcta gestión de la información que genera.

### 1.1.2 LA GESTIÓN DE LA INFORMACIÓN EN BIOPSIAS

Los datos recogidos durante la biopsia quedan plasmados en un documento conocido como **Modelo de Solicitud de Biopsia (citología) (Anexo 1) y (Anexo 2)**, el cual incluye datos demográficos y hospitalarios, datos clínicos (incluye aspectos quirúrgicos y biopsias anteriores), descripción macroscópica y microscópica, diagnóstico anatomopatológicos<sup>4</sup>, así como los diagnósticos complementarios relevantes de laboratorios y complementarios de imagenología, haciendo una impresión diagnóstica clínica, observaciones y la pieza enviada a anatomía patológica (tipo de estudio citológico realizado).

El patólogo responsable de la biopsia debe velar que se diagnostiquen correctamente los resultados del estudio de una muestra de tejido u órgano del paciente para saber cuál es su enfermedad, si es benigna o maligna. Estos se recogen en el modelo y se conservan en historias clínicas para futuras evaluaciones. Adicionalmente, realizará un control de calidad a los datos antes de firmar el informe anatomopatológico correspondiente.

El diagnóstico anatomopatológico es el producto generado en los servicios de Anatomía Patológica. Constituye una opinión médica experta, basada en el estudio interpretativo de las muestras del paciente integrado con los datos procedentes de la historia clínica, y en la que el peso principal descansa en un conocimiento científico actualizado y en la experiencia del patólogo (Castaño 2013). Será la principal herramienta para profundizar en la correlación clínico-patológica y obtener conocimiento que permita

---

<sup>4</sup>El **diagnóstico anatomopatológico** continúa siendo un arma eficaz para la confirmación del diagnóstico clínico. Es una herramienta que contribuye a una mejor docencia médica, conocimiento de las patologías y valoración de la calidad en el diagnóstico y terapéutica clínica

mejorar la calidad del trabajo médico y el tratamiento de las enfermedades (Arango Astorga & Cabrera Nicolau, 2014).

Los informes deben ser portadores de información consistente, elaborados con claridad y en un formato que permita fácilmente la recuperación de la información. El informe anatomopatológico ideal mezcla elementos sinópticos con descripciones narrativas que permiten encontrar rápidamente los elementos más significativos y también se puede analizar la opinión del patólogo. Cualquier forma de estandarización del informe debe ser flexible a incorporar las notas que el patólogo estime convenientes(Arango Astorga & Cabrera Nicolau, 2014).

La estructuración de la información es el mejor modo de asegurar su disponibilidad y explotación, sea con fines docentes o investigativos; pero no es lo único importante. El uso de estándares de terminología para la codificación de los diagnósticos juega un papel esencial. Una terminología clínica estándar hace una contribución significativa a la mejora de la calidad y seguridad de la asistencia sanitaria, permitiendo la representación coherente de la información y su posterior recuperación(Arango Astorga & Cabrera Nicolau, 2014).

Existen diversas formas de codificar, desde sistemas de nomenclaturas hasta calificaciones de las enfermedades propias de países o de instituciones, o incluso personales. Los dos sistemas utilizados internacionalmente para la codificación de los diagnósticos individuales son la Clasificación Internacional de Enfermedades (CIE) de la OMS y la Nomenclatura Sistematizada de Medicina “Systematized Nomenclature of Medicine” (SNOMED) de colegio de Patólogos Norteamericanos “College of American Pathologists” (CAP)(Amat, 2014).

El estándar de terminología clínica “*SNOMED ClinicalTerms*” (SNOMED CT), acrónimo de SystematizedNomenclatureof Medicine ClinicalTerms es una terminología médica integral, controlada, que ofrece contenido clínico y expresividad para la documentación y la comunicación médica. Se trata de una terminología constituida por conceptos, términos y relaciones cuyo fin es representar con precisión toda la información médica. El contenido se divide en jerarquías que abarcan, por tanto, todos los aspectos de la información del paciente, desde hallazgos clínicos, procedimientos como intervenciones, tratamientos, cirugías, hasta información sobre el contexto social y demográfico del paciente(Pablo, 2018). Es una terminología de referencia que permite a los profesionales de la salud de todo el mundo representar la información clínica de forma precisa e inequívoca, el sitio oficial de la (IHTSDO) la reconoce como la terminología clínica multilingüe más completa en el mundo(Arango Astorga, Nicolau Cabrera & Amat, 2019).

Según el sitio oficial SNOMED International Release Management Home, SNOMED CT ofrece una amplia gama de opciones diferentes para la recuperación inmediata y posterior reutilización para abordar de inmediato y a más largo plazo los requerimientos médicos y los de otros usuarios. La naturaleza de las jerarquías de SNOMED CT permiten recuperar y reutilizar selectivamente la información para cumplir con diferentes requerimientos en diversos niveles de generalización (Confluence, 2019).

Snomed CT es apoyado por los países miembros de la IHTSDO, entre ellos: Australia, Bélgica, Canadá, Chile, Dinamarca, España, Estados Unidos, Islandia, Israel, Malasia, Nueva Zelanda, Países Bajos, Polonia, Singapur, Suecia, República de Checa y Reino Unido. Su utilización requiere la compra de una Licencia de Afiliados por los no miembros de la IHTSDO (Arango Astorga & Cabrera Nicolau, 2014).

Por otro lado, el sitio oficial de la Oficina Regional para las Américas de la Organización Mundial de Salud (OPS/OMS), la Clasificación Estadística Internacional de Enfermedades y Problemas Relacionados con la Salud, Décima Revisión (CIE-10) es la última en una serie que tiene sus orígenes en el año 1950. La OMS asumió la responsabilidad de la Clasificación Internacional de Enfermedades (CIE) a su creación en 1948 cuando se publicó la Sexta Revisión, desde entonces su propósito ha sido permitir el registro sistemático, el análisis, la interpretación y la comparación de los datos de mortalidad y morbilidad recolectados en diferentes países o áreas, en diferentes momentos (Arango Astorga, Nicolau Cabrera & Amat, 2019)

La CIE permite la conversión de los términos diagnósticos y de otros problemas de salud, del lenguaje común a códigos alfanuméricos que faciliten su almacenamiento y posterior recuperación para el análisis de la información. Su más reciente revisión (CIE-10) constituye uno de los estándares internacionales más usados para la elaboración de las estadísticas de morbilidad y mortalidad en el mundo (Arango Astorga, Nicolau Cabrera & Amat, 2019).

En Cuba la elección de la CIE es casi obligada al estar imposibilitados de obtener el SNOMED completo y actualizado. Además, este último puede resultar poco práctico como resultado de su excesivo nivel de detalles; en tanto la CIE, de ser conveniente una especificación más detallada de la enfermedad, puede ser complementada con los ejes morfológico y topográfico. Siendo el eje morfológico un complemento para la codificación de los tumores y el eje topográfico para una mejor localización del sitio de las enfermedades (Arango Astorga, Nicolau Cabrera & Amat, 2019).

El sistema de codificación es la base de un sistema de información eficiente para los servicios de Anatomía Patológica, el cual sirve de apoyo para las funciones docentes, investigativas y de aseguramiento de la calidad de los servicios. Posibilita la recuperación y procesamiento rápido y eficiente de los resultados de la biopsia, que, analizado y empleado adecuadamente, contribuyen a la disminución de la morbilidad y mortalidad.

## 1.2 ANÁLISIS DE LAS SOLUCIONES EXISTENTES

En la actualidad, existen una gran cifra de empresas que se dedican al desarrollo de sistemas informáticos para la gestión de información hospitalarias (HIS, por sus siglas en inglés). Diversas soluciones están disponibles en el mercado, unas más generales y otras para departamentos específicos. Los sistemas de información son herramientas básicas que dan respuesta a la labor asistencial de los servicios de anatomía patológica, que va dirigida a la elaboración de diagnósticos, evaluaciones pronosticadas y selección de dianas terapéuticas. Además, sirven de apoyo para las funciones docentes, investigadoras y de calidad de los servicios de anatomía patológica.

Los dos países que resaltan en el desarrollo y comercialización de sistemas de información en Anatomía Patológica son: España y Estados Unidos. Una encuesta referida en el *“Libro Blanco de Anatomía Patológica en España 2019”* revela que en este país el 98,7% de los centros tienen un programa informático específico para Anatomía Patológica. El más ampliamente implantado es Patwin, en sus diferentes versiones, seguido de Vitropath y Gespath y el grado de satisfacción mayor es para (Patwin 4.9), seguido de Gespath y Novopath (Guerra, 2019b). Por otro lado, indica *“CAP Today”* que en Estados Unidos las empresas que han distribuido más números de contratos para sitios que operan los sistemas de anatomía patológica se encuentran las siguientes: ApEasy (596), Cerner Corp. (177), Computer Trust Corp. (106), Epic (123), Novopath (245), Psyche Systems Corp. (107), SCC Soft Computer (139), Sunquest (357), Technidata (150<sup>+</sup>), VitalAxis (112) (Cap, 2019). **La Tabla 1** resume las características más importantes de los sistemas más conocidos con soporte para la gestión de Anatomía Patológica.

**Tabla 1:** Características de los sistemas más conocidos con soporte para la gestión de Anatomía Patológica Internacionales.

Nombre	País	Plataforma	Tipo de aplicación	Sistema de codificación	Software propietario
Patwin	España	Windows	Cliente - Servidor	SNOMED	Propietario
Vitropath	España	Multiplataforma	Cliente - Servidor	Configurable	Propietario
Novopath	España	Windows	Web	Configurable	Propietario
Gespath	España	Windows	Cliente - Servidor	SNOMED	Propietario
Cerner CoPathPlus	EE.UU	Windows	Escritorio	SNOMED	Propietario
AP Easy	EE.UU	Windows	Web	Configurable	Propietario
WinSURGE	EE.UU	Windows	Web	SNOMED	Propietario
EpicBeaker	EE.UU	Windows	Web	SNOMED	Propietario
NovoPathAnatomicPathology†	EE.UU	Windows	Web	SNOMED	Propietario
WindopathE.essential	EE.UU	Windows	Web	SNOMED	Propietario
SoftPathDx	EE.UU	Multiplataforma	Web	SNOMED	Propietario

Nombre	País	Plataforma	Tipo de aplicación	Sistema de codificación	Software propietario
SunquestCoPathPlus	EE.UU	Windows	Cliente - Servidor	SNOMED	Propietario
SunquestPowerPath	EE.UU	Windows	Cliente - Servidor	SNOMED	Propietario
TD-Histo/Cyto	EE.UU	Windows	Web	SNOMED	Propietario
VitalDx LIS	EE. UU	Windows	Web	Configurable	Propietario

De manera general, los datos evidencian un predominio de software privativo en el mercado internacional, el uso de SNOMED CT como sistema de codificación, arquitectura cliente servidor y dependencia de la plataforma Windows. Lo anterior supone el pago de costosas licencias para su adquisición y actualización, sin contar las limitaciones de Cuba para el uso del SNOMED y la contrariedad con las políticas cubanas de migración a software libre que representa el propio sistema y su dependencia del sistema operativo Windows.

Por lo regular, las empresas no venden programas informáticos, los “licencian”. El comprador de un producto adquiere el derecho de utilizarlo como guste, por el contrario, al adquirir una licencia solo se está pagando por utilizarlo según las condiciones fijadas por el productor. Es común entonces que la solución a los problemas sea actualizar el software a una nueva versión, teniendo que pagar por ello; e incluso, la interrupción en la asistencia técnica a productos “antiguos” según su proveedor.

En las condiciones actuales, la soberanía tecnológica constituye un elemento de seguridad nacional para cualquier país, y especialmente para Cuba, que ha sido víctima por más de 50 años de un férreo bloqueo económico, comercial y financiero impuesto por el gobierno de los Estados Unidos sobre la nación. Los departamentos de Anatomía Patológica almacenan información considerablemente sensible para un país, la vida privada, la salud y las libertades individuales; por tanto, se debe garantizar la seguridad y la confiabilidad de su gestión, de lo cual no se podrá estar absolutamente seguro con el empleo de software propietario, por no conocer el funcionamiento interno del producto.

Es posible establecer entonces que la alternativa foránea resulta inapropiada en las condiciones cubanas, corroborando que la solución debe provenir de la creatividad nacional. Especial atención merecen entonces los desarrollos de origen cubano. En el siguiente epígrafe se realiza un análisis general de las tendencias actuales y tecnologías utilizadas en el desarrollo de los sistemas de información en Anatomía Patológica cubanos.

### 1.2.1 SARCAP

El Sistema Automatizado de Registro y Control en Anatomía Patológica (**SARCAP**) fue desarrollado en Cuba, en el hospital “Dr. Luis Díaz Soto” en el año 1985. Posteriormente aplicado en otros departamentos

de Anatomía Patológica del país, con el objetivo de registrar y controlar de forma automatizada la información obtenida de las biopsias y autopsias, para facilitar su procesamiento y recuperación con fines docentes, científicos, asistenciales y administrativos (Amat, 2014).

El SARCAP está compuesto por 2 subsistemas (autopsias y biopsias), los cuales presentan una estructura modular y utilizan un diccionario común con los códigos de las enfermedades y sus correspondientes descriptores. Posee funcionalidades para la introducción y modificación de los datos de los estudios realizados, así como su recuperación a partir de diferentes criterios en dependencia de las variables objeto de estudio. Permite realizar diferentes controles de calidad a los datos, imprimir informes y obtener una buena cantidad de reportes estadísticos. Estando desarrollado sobre un ambiente MS-DOS y desatendido desde el año 2002, el sistema posee un conjunto de limitaciones que se considera necesario resolver, para lo cual es preciso rehacerlo completamente (Arango Astorga & Cabrera Nicolau, 2014).

Dadas sus características, el SARCAP no ha podido ser actualizado a la última revisión de la Clasificación Internacional de Enfermedades (CIE-10) de la Organización Mundial de la Salud (OMS). La codificación actual de los diagnósticos resulta redundante en algunos casos e insuficiente en otros para lograr una especificación detallada de la enfermedad, especialmente cuando resulta conveniente complementarla con los ejes morfológicos y topográficos, así como el uso de comodines. (Amat, 2014).

Es dependiente del sistema operativo Windows. Posee una interfaz poco amigable y limitados recursos de visualización de la información recuperada y las estadísticas generadas. La validación de la entrada de los datos es insuficiente y precisa la memorización de los códigos de los diagnósticos. Finalmente, funciona de manera aislada en cada uno de los hospitales y no posibilita la transmisión de los datos de las autopsias y de las biopsias a la base de datos nacionales.

### 1.2.2 AvanPat

En el 2004 se dio a conocer AvanPat 1.1 en el VI Congreso Virtual Hispanoamericano de Anatomía Patológica. Surge como una respuesta a la necesidad que el desarrollo creciente de la medicina cubana, en áreas de investigación como la anatomía patológica, demanda en cuanto a la fluidez de la información, asesoramiento, procesamiento e intercambio de datos, incluyendo imágenes. Las dispersiones a lo largo del país de centros hospitalarios con posibilidad de aplicación de tecnologías de avanzadas en áreas del diagnóstico abren una nueva etapa donde el trabajo en redes, la aplicación de la telepatología y otras técnicas informáticas se convierten en una realidad potencial (Medell, 2004).

**AvanPates** un sistema que almacena, organiza y gestiona la información relacionada con los estudios que se realizan en los departamentos de Anatomía Patológica. Cuenta con un conjunto de plantillas prediseñadas para estudios (citologías de líquidos, vaginales y punción aspirativa con aguja fina, biopsia y



necropsias) con más de 350 campos incluyendo imágenes y video. Permite al patólogo diseñar nuevos estudios o agregar campos a los prediseñados. Genera de forma automática los informes administrativos que deben entregar los departamentos de Anatomía Patológica acerca del trabajo realizado en el periodo de tiempo seleccionado. Posee módulo para estadísticas con fines investigativos. Incorpora un registro de pacientes que permite almacenar la información de forma personalizada para su análisis evolutivo, aunque es posible integrarse con uno existente (Medell, 2004).

Posterior a su publicación, no existen referencias de la utilización del sistema. No se le dio seguimiento y no se tienen noticias de su empleo en centros de salud cubanos para la introducción de los datos de estudios anatomopatológicos realizados. En consecuencia, no ha recibido la validación de los especialistas en la práctica; lo cual se pudo corroborar en entrevista realizada al Dr. Cs. José de Mendoza Amat, responsable del tema en el Grupo Nacional de Anatomía (Arango Astorga & Cabrera Nicolau, 2014).

### 1.2.3 ALAS HIS

Finalmente, y de muy reciente creación, el Sistema de Información Hospitalaria Alas HIS, desarrollado en la Universidad de las Ciencias Informáticas, está concebido como una aplicación web a ser desplegada en cada centro de salud, con una base de datos local encargada del almacenamiento y reporte de los datos generados por los distintos módulos instalados. A su vez, todos los sistemas realizarán réplica de la información de la Historia Clínica Electrónica Única hacia un Centro de Datos Central, garantizando la seguridad mediante la implementación de control de accesos basado en roles. Está codificado en Java y es multiplataforma, utiliza PostgreSQL como gestor de bases de datos y JBoss Application Server como servidor de aplicaciones (Arango Astorga & Cabrera Nicolau, 2014).

El sistema Alas HIS ha venido ganando un prestigio internacional en los últimos tiempos, logrando ser desplegado en diferentes centros hospitalarios venezolanos. Sin embargo, la concepción integral del sistema y sus características particulares, hacen pensar que el mismo fue concebido para su exportación y no para ser desplegado en Cuba, cuyas deficiencias en la conectividad interdepartamental de los centros de salud y el alto grado de obsolescencia tecnológica hacen la solución inviable en la mayoría de las instituciones hospitalarias de la Isla. Hasta el momento, el sistema no ha podido ser desplegado en ningún centro de salud cubano (Arango Astorga, Nicolau Cabrera & Amat, 2019).

Por otro lado, el módulo de Anatomía Patológica se queda en la gestión de solicitudes, muestras e información macro y microscópica de autopsias, biopsias y citologías, así como la generación de reportes e informes. No se realiza un apropiado control de calidad sobre los datos ni se facilita la identificación y clasificación de discrepancias diagnósticas, aspectos de suma importancia para la evaluación de la calidad asistencial (Arango Astorga, Nicolau Cabrera & Amat, 2019).

### 1.2.4 SIRCAP

El análisis de las tecnologías de la información y las comunicaciones aplicadas al área de Anatomía Patológica, demostró la necesidad de un nuevo sistema informático en la búsqueda del mayor aprovechamiento de los estudios patológicos.

Esta nueva versión basada en los mismos fundamentos del actual SARCAP, posee una plataforma e interfaz más moderna y permite la conversión de las más de 140 mil autopsias contenidas en la base de datos existentes en este momento una vez que estén creadas las condiciones para ampliar su explotación. Según sus siglas, SARCAP pasa de ser automatizado a informatizado, SIRCAP, dada la posibilidad de aplicar minería de datos (Arango Astorga, Nicolau Cabrera & Amat, 2019).

SIRCAP admite la introducción de nuevos casos, así como la codificación de los diagnósticos de causa de muerte según los estándares establecidos por la OMS, permitiendo que la misma sea complementada con los ejes morfológico, topográfico y comodines, lográndose una mayor precisión del diagnóstico realizado. Además, posibilita la codificación de procedimientos médicos realizados al caso durante su estancia en los centros hospitalarios (Arango Astorga, Nicolau Cabrera & Amat, 2019).

De igual forma podrá realizarse un control de calidad sobre los datos introducidos con el objetivo de garantizar el cumplimiento de la metodología establecida y la calidad de la información. El sistema a partir de las reglas configuradas permitirá realizar correcciones automáticas a la información almacenada o mostrará los casos que deben ser revisados por los especialistas (Arango Astorga, Nicolau Cabrera & Amat, 2019).

Para un soporte completo a la gestión de la información en los departamentos de Anatomía Patológica, sería de utilidad el desarrollo de una especialización dentro del sistema para el procesamiento de autopsias perinatales, así como su extensión a biopsias y citologías (Arango Astorga, Nicolau Cabrera & Amat, 2019).

**Tabla 2:** Características de los sistemas más conocidos con soporte para la gestión de Anatomía Patológica Cubanos.

Nombre	Plataforma	Tipo de aplicación	Sistema de codificación	Software propietario
<b>SARCAP</b>	<i>Windows</i>	<i>Escritorio</i>	<i>CIE</i>	<i>Propietario</i>
<b>AvanPat</b>	<i>Windows</i>	<i>Cliente -Servidor</i>	<i>Configurable</i>	<i>Propietario</i>
<b>Alas HIS(Módulo AP)</b>	<i>Multiplataforma</i>	<i>Web</i>	<i>CIE</i>	<i>Propietario</i>
<b>SIRCAP</b>	<i>Multiplataforma</i>	<i>Escritorio</i>	<i>CIE</i>	<i>Libre</i>

Después de realizar el análisis de las soluciones informáticas existentes en el área de Anatomía Patológica surgió la necesidad de hacer una nueva herramienta ya que ninguno de los sistemas anteriores le da solución a este problema.

### **1.3 METODOLOGÍA, LENGUAJES Y HERRAMIENTAS DE DESARROLLO**

Para el desarrollo del Sistema de Información para Anatomía Patológica se utilizaron herramientas, tecnologías, lenguajes, arquitectura y metodología que guían el desarrollo de *software* para la gestión de la información de las biopsias. Las decisiones de *software* empleadas en el desarrollo del módulo de biopsia derivan del sistema general, es decir, del Sistema Informático de registro y control en Anatomía Patológica. De esta forma se mantiene la integridad tecnológica en la solución general. En este epígrafe se describe cada una de ellas y se indica la función que realizan dentro del sistema.

#### **1.3.1 METODOLOGÍA DE DESARROLLO**

Se adoptó como metodología Programación Extrema (eXtreme Programming, XP). 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. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y facilidad para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (Letelier & Pénade 2019). Los programadores trabajan en parejas y se desarrollan pruebas para cada una de las tareas. Todas las pruebas deben ser ejecutadas satisfactoriamente cuando el nuevo código es integrado al sistema (Sommerville, 2007).

#### **1.3.2 LENGUAJE DE MODELADO**

El Lenguaje Unificado de Modelado (UML por sus siglas en inglés) es un lenguaje de representación visual que permite combinar diversos elementos gráficos y crear diagramas. Se usa para modelar sistemas y usa tecnología orientada a objetos. Describe lo que hará un sistema, pero no dice cómo implementarlo. Su objetivo es visualizar, especificar, construir y documentar los artefactos que se crean durante el proceso de desarrollo. Involucra todo el ciclo de vida del proyecto y está pensado para varios lenguajes y plataformas. Es aplicable en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso utilizar (Group, 2007).

### 1.3.3 INGENIERÍA DE SOFTWARE ASISTIDA POR COMPUTADORA (CASE)

Para (Sommerville, 2007) CASE(ComputerAided Software Engineering) es el nombre con el que se identifica la herramienta utilizada para apoyar las actividades del proceso de software. Según (Kendall & Kendall, 2005) estas herramientas facilitan la interacción entre los miembros de un equipo al hacer que la diagramación sea un proceso iterativo y dinámico. Los analistas de sistemas se apoyan en estas herramientas desde el principio hasta el fin del ciclo de vida, para incrementar la productividad, comunicarse de manera eficiente con los usuarios y desarrolladores, e integrar el trabajo que desempeñan en el sistema.

**Visual Paradigm**, versión 8.0, es una herramienta multiplataforma de diseño y fácil de usar para sistemas de tecnología informática. CASE es una herramienta aplicable en todo el ciclo de vida del desarrollo de *software*. Soporta UML, BPMN (*Business Process Modeling Notation*, por sus siglas en inglés), entre otras tecnologías. Permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación. La herramienta proporciona abundantes tutoriales UML, demostraciones interactivas de UML y proyectos UML. Es fácil de instalar y actualizar y compatible entre ediciones (Paradigm, 2019).

### 1.3.4 LENGUAJE DE PROGRAMACIÓN

Las afirmaciones de (ORACLE, 2019b) apuntan que **JavaFX** es una tecnología que ofrece un modelo de desarrollo y despliegue unificado para la creación de Aplicaciones de Internet Enriquecidas (Rich Internet Applications, RIA), que no son más que aplicaciones web con las características y capacidades de aplicaciones de escritorio.

Las aplicaciones JavaFX son completamente desarrolladas en Java, una de las tecnologías más ampliamente desplegadas con una de las más grandes comunidades de desarrolladores en el mundo. Se vale de beneficios de la plataforma tales como la programación orientada a objetos, herencia, polimorfismo y manejo de excepciones; al tiempo que aprovecha la potencia de las prácticas de programación basadas en patrones de diseño. JavaFX cuenta con su propio lenguaje declarativo JavaFX Script, al igual que un conjunto de herramientas de diseño y desarrollo, lo cual permite crear contenido expresivo, dinámico y funcional, combinando las mejores capacidades de la plataforma Java con funcionalidades de multimedia interactivas (texto, imagen, audio y video). Es una plataforma de código abierto, de libre distribución y fácil acceso, integrada en las versiones más recientes del popular IDE NetBeans (Arango Astorga & Cabrera Nicolau, 2014),(ORACLE, 2019b).

### 1.3.5 ENTORNO DE DESARROLLO INTEGRADO

Las observaciones de (Bell & Parr, 2003); (Matellán Olivera, 2004);(Rouse, 2007) señalan que un IDE (*IntegratedDevelopmentEnvironment*) es un programa compuesto por un conjunto de herramientas que proveen facilidades a los programadores para agilizar el proceso de desarrollo de *software*. Consta de un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

De acuerdo con (ORACLE, 2019b) **NetBeans IDE** es un proyecto exitoso de código abierto con una gran cantidad de usuarios, una comunidad en constante crecimiento. Proporciona un amplio soporte para las últimas tecnologías de Java, incluyendo JavaFX. Soporta integración con diferentes herramientas para el control de versiones y Sistemas Gestores de Bases de Datos (SGBD). Es un producto libre, gratuito y sin restricciones de uso. Su empleo está determinado por la experiencia del equipo de desarrollo en su utilización, en virtud de minimizar el tiempo necesario para la familiarización con nuevos lenguajes y herramientas de desarrollo.

### 1.3.6 CONTROL DE VERSIONES

**Git** es un sistema de control de versiones distribuido, gratuito y de código abierto diseñado para manejar, desde proyectos pequeños hasta muy grandes, con velocidad y eficiencia. es un sistema de control de versiones gratuito de código abierto. Posee un repositorio que almacena información en una jerarquía típica de archivos y directorios, manejándolos en el tiempo y recordando todos los cambios realizados sobre los mismos, incluso los cambios en el propio árbol de directorios, lo cual permite recuperar versiones antiguas de los archivos o examinar el historial de modificaciones realizadas. Supera a las herramientas de Gestión de configuración de software (SCM por sus siglas en inglés) como Subversión, el Sistema de Control (CVS por sus siglas en inglés), Perforce y ClearCase con características como ramificación local, áreas de preparación convenientes y múltiples flujos de trabajo (Git, 2020).

### 1.3.7 PROTOTIPOS DE INTERFACES DE USUARIO

JavaFXSceneBuilder es una herramienta de diseño visual que permite a los usuarios diseñar rápidamente interfaces de usuario de aplicaciones JavaFX, sin codificación. Los usuarios pueden arrastrar y soltar componentes de la interfaz de usuario a un área de trabajo, modificar sus propiedades, aplicar hojas de estilo y el código FXML para el diseño que están creando se genera automáticamente en segundo plano. El resultado es un archivo FXML que luego se puede combinar con un proyecto Java al vincular la interfaz de usuario a la lógica de la aplicación (Oracle, 2019a).

### 1.3.8 HERRAMIENTA PARA LA GENERACIÓN DE REPORTE

La generación de reportes constituye un componente esencial en todo sistema informático de gestión, especialmente en Sistemas de Información para Anatomía Patológica, en virtud de facilitar el análisis de la información recogida como apoyo a la toma de decisiones (Arango Astorga & Cabrera Nicolau, 2014).

Para la generación de reportes en la aplicación, se hará uso del conjunto de bibliotecas de código abierto **JasperReports**, por constituir el motor de reportes más popular para aplicaciones Java. Permite la generación de informes con un alto grado de profesionalidad, a partir de la incorporación de imágenes, gráficos y tablas; facilitando su exportación en una amplia variedad de formatos que incluyen HTML, PDF, Word y Excel (Heffelfinger, 2006).

### 1.3.9 SISTEMA GESTOR DE BASES DE DATOS

Las valoraciones de (Hansen & Hansen, 1997); (Date, 2001); (Sánchez Asenjo, 2013) apuntan que un SGBD (Data Base Management System, DBMS) permite a los usuarios procesar, describir, administrar y recuperar de forma rápida y eficiente los datos almacenados en una base de datos; garantizando la seguridad de los mismos mediante el acceso limitando al personal autorizado. Su objetivo fundamental es suministrar al usuario las herramientas que le permitan manipular, entérminos abstractos, los datos; de forma que no sea necesario conocer cómo están estos almacenados en la computadora, o el método de acceso empleado (Mato García, 2005).

Según la Comunidad Cubana de PostgreSQL, el mismo es un potente sistema gestor de bases de datos multiplataforma bajo licencia BSD (Berkeley Software Distribution), menos restrictiva que otras como GPL (General Public License) y muy cercana al dominio público (PostgreSQL, 2020a). En sus últimas versiones, incorpora características que habitualmente solo se ven en productos comerciales de alto calibre. Garantiza a sus usuarios atomicidad, consistencia, aislamiento y durabilidad. La propia referencia, lo identifica como el SGBD de código abierto más avanzado del mundo, principal argumento considerado para su elección, en correspondencia con las políticas cubanas de migración a software libre (PostgreSQL, 2020a), (PostgreSQL, 2020b).

### 1.3.10 MAPEO OBJETO-RELACIONAL

Hibernate, versión 4.0, es una herramienta ORM (*Object Relational Mapping*, por sus siglas en inglés) para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos XML que permiten establecer estas relaciones. Es de libre distribución, siendo de las más completas. Actualmente su uso está muy extendido y está siendo desarrollada de forma muy activa. Hibernate para Java puede ser utilizado en aplicaciones

Java independientes o en aplicaciones Java EE, mediante el componente *HibernateAnnotations* que implementa el estándar JPA, que es parte de esta plataforma (Ottinger, Linwood & Minter, 2016).

#### 1.4 PATRONES PARA EL DESARROLLO DE SOFTWARE

El patrón es una descripción de un problema y su solución que recibe un nombre y que es aplicable en otros contextos, con una sugerencia sobre la manera de utilizarlo en circunstancias diversas (Larman, 1999a). Los patrones tienen como objetivo capturar y establecer explícitamente conocimiento abstracto de resolución de problemas, que suele ser implícito y se adquiere solo a través de la experiencia (Winn & Calder, 2002).

Un patrón se caracteriza por cuatro elementos esenciales, un nombre corto (describe el problema de diseño), el problema potencial (describe cuándo aplicar el patrón), la solución (describe los elementos que componen el diseño, sus relaciones, responsabilidades y colaboración) y las consecuencias o resultados de su aplicación, importantes para la evaluación de diseños alternativos a partir de los costos y beneficios de su aplicación (Gammay otros, 2004).

La literatura consultada hace referencia a tres categorías de patrones:

- **Patrones de Arquitectura:** los patrones de arquitectura resuelven los problemas relacionados con el estilo arquitectónico, representan la relación entre los componentes de todo el sistema y cada uno de ellos está compuesto por pequeños módulos. Proveen un conjunto de subsistemas predefinidos donde se especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos. Los patrones arquitectónicos tienen un gran impacto en la base del código, con mayor frecuencia afectando toda la aplicación (Global), ya sea horizontal (es decir, cómo estructurar el código dentro de una capa) o verticalmente (es decir, cómo se procesa una solicitud desde las capas externas hacia las capas internas), ya que expresan un esquema organizativo estructural para sistemas de software (Advance, 2019).
- **Patrones de Diseño:** los patrones de diseño definen esquemas para refinar los subsistemas o componentes de sistema software o sus relaciones. Describen una estructura recurrente y común que resuelve un problema de diseño dentro de un contexto, como los descritos por (Gammay otros, 2004).
- **Patrones de Implementación:** son patrones de bajo nivel específicos de un lenguaje de programación. Describen cómo implementar aspectos particulares de componentes o sus relaciones, utilizando las características de un lenguaje determinado. Representan el nivel más bajo de patrones, manejan aspectos tanto de diseño como de implementación y son mayormente utilizados durante la fase de desarrollo del sistema (Marquina & Parra, 2008).

La utilización de patrones surte considerables beneficios al desarrollo de sistemas. Los patrones ofrecen soluciones probadas que pueden ser aplicadas una y otra vez en problemas similares, en diferentes momentos y proyectos; al tiempo que proveen un poderoso mecanismo de reutilización. Contribuyen a la transmisión de ideas, haciendo uso de un vocabulario y formato común, lo cual propicia una comunicación efectiva pues solo de mencionar el nombre de un patrón se evita realizar largas descripciones sujetas a ser incomprendidas por el interlocutor (Alur, Crupi & Malks, 2003).

## 1.5 ESTRATEGIA DE PRUEBAS

La gestión de calidad, es un conjunto de acciones y herramientas que tienen como objetivo evitar posibles errores o desviaciones en el proceso de producción y en los productos o servicios obtenidos mediante el mismo (ISO, 2018). Además, es una actividad que se aplica a lo largo del proceso de software. El cumplimiento de la especificación del programa y la satisfacción de las expectativas del cliente deben estar en la mira del equipo de desarrollo a cada momento, a lo que usualmente suele llamarse verificación y validación (Arango Astorga & Cabrera Nicolau, 2014).

Las pruebas de software son el proceso de verificación y validación del servicio o aplicación de software al verificar si cumple con los requisitos del usuario y qué se implementa según las características. Además, juegan un papel vital en el proceso de desarrollo de un software de alta calidad (Uddin Azeem, 2019). Constituyen el último bastión en el aseguramiento de la calidad. Son un conjunto de actividades planeadas con anticipación y realizadas de forma sistemática, con el fin de descubrir resultados distintos al esperado; de ahí la necesidad de contar con una estrategia de pruebas desde los primeros momentos del proceso de software (Arango Astorga & Cabrera Nicolau, 2014).

Reconocidos autores (Pressman, 2005); (Sommerville, 2007) tienden a coincidir en la definición de los siguientes niveles de pruebas, adaptables a las condiciones del desarrollo en cuestión:

- **Pruebas unitarias:** se concentran en el diseño y comportamiento de cada componente del software, desde la perspectiva de su implementación.
- **Pruebas de integración:** prueban la correcta relación entre los componentes del software.
- **Pruebas de aceptación:** evalúan el cumplimiento de los requisitos pactados con el cliente.
- **Pruebas de sistema:** se ejecutan en un ambiente similar al ambiente operacional real, probando el software como un todo de conjunto con el resto de los elementos del sistema.

En la metodología XP las pruebas son tan importantes como la programación, fortalecen la confianza del cliente en el equipo de desarrollo y dentro de este. Los máximos exponentes de esta metodología (Beck &



Andres, 2004), proponen dos principios para aumentar la rentabilidad de las pruebas: el chequeo doble y el incremento del costo de los defectos.

El chequeo doble consiste en la aplicación de dos conjuntos de pruebas: uno escrito desde la perspectiva de los programadores y orientado a probar exhaustivamente cada uno de los componentes del sistema, y el otro escrito desde la perspectiva de los clientes o usuarios finales, para probar el funcionamiento del sistema como un todo. Lo cual es coherente con los niveles de pruebas de unidad y aceptación respectivamente.

Por otro lado, el costo de los defectos se incrementa con el tiempo, de ahí la máxima de corregirlos tan pronto como sean descubiertos, lo cual implica ejecutar pruebas a cada uno de los componentes una vez que están listos y luego con relativa frecuencia. De lo anterior se infiere que las mismas personas que cometen los errores (los programadores), deben escribir las pruebas de unidad, a su vez deben ser automatizadas y siempre que sea posible, escritas antes de la implementación, evitando así que se vean influenciadas por la misma.

Para la validación del cumplimiento de las expectativas del cliente, así como el descubrimiento de errores que solo los usuarios finales son capaces de detectar, resultan de vital importancia las pruebas de aceptación, usualmente pruebas alfa y beta.

Las **pruebas alfa** las realiza el usuario en un entorno controlado por el equipo de desarrollo, mientras los desarrolladores registran los errores y problemas detectados (Pressman, 2005).

Las **pruebas beta** se aplican en el lugar y condiciones reales en que será ejecutado el software, en ausencia de los desarrolladores. El usuario final registra los problemas detectados (reales o imaginarios) y los informa regularmente a los desarrolladores (Pressman, 2005).

## 1.6 CONCLUSIONES PARCIALES

- La descripción de los conceptos asociados al objeto de estudio permitió contextualizar los principales términos abordados en el capítulo y la investigación en general.
- EL análisis de los sistemas informáticos estudiados constituyó un punto de referencia para el desarrollo de las funcionalidades de la especialidad de anatomía patológica, como parte del módulo de Biopsia del Sistema de Registro y Control para Anatomía Patológica.
- Para mantener la integridad del sistema SIRCAP, fueron seleccionados la arquitectura, metodologías y tecnologías que permiten mantener la línea base de desarrollo de este sistema.

## CAPÍTULO 2:ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

Como resultado de la investigación desarrollada en el capítulo anterior y con el objetivo de adicionar un nuevo módulo de Biopsia para el sistema de registro y control para el departamento de Anatomía Patológica,el presente capítulo recoge las principales características de la propuesta de solución. Se define el modelo de dominio de la aplicación, los principales conceptos y sus relaciones. Se analizan lastécnicas de captura de requisitos, las historias de usuario y prototipos empleados en la validación de los requisitos. Resultado de la planificación, se presenta el plan de iteraciones para el desarrollo del sistema. Se describe la arquitectura, los modelos de diseño y de datos y los patrones aplicados.

### 2.1 . MODELO CONCEPTUAL

El modelo conceptual es el análisis orientado a objetos que tiene por finalidad estipular una especificación del dominio del problema y los requerimientos desde la perspectiva de la clasificación por objetos y desde el punto de vista de entender los términos empleados en el dominio. Para descomponer el dominio del problema hay que identificar los conceptos, los atributos y las asociaciones del dominio que se juzgan importantes(Larman, 1999b). Es un artefacto útil que permite obtener conocimiento acerca de cómo se desenvuelve el problema en el contexto real y considerado un diccionario visual de las abstracciones relevantes, vocabulario e informaciones del dominio. Constituye una primera aproximación al diseño definitivo.

La Figura 1 refleja los principales conceptos del dominio y las relaciones entre ellos.

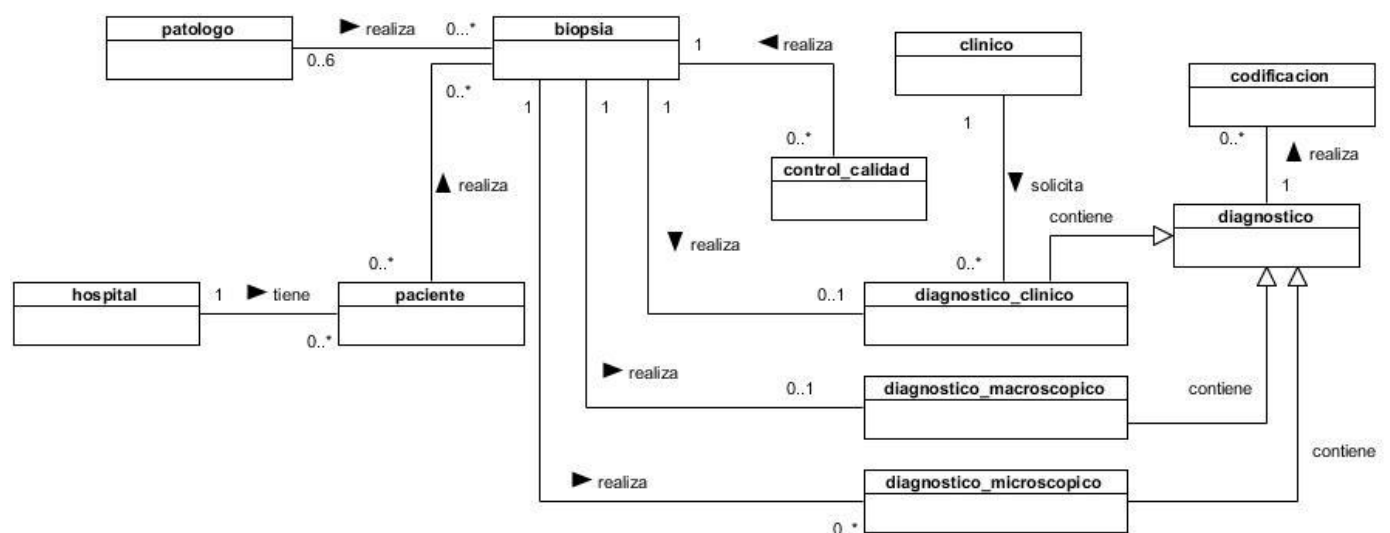


Figura 1:Modelo Conceptual

**Tabla 3:** Descripción de los Conceptos del Modelo Conceptual

<b>Conceptos</b>	<b>Descripción</b>
<b>Patólogo</b>	Es el encargado de realizar las biopsias.
<b>Biopsia</b>	Se realiza diagnóstico clínico, macroscópico y microscópico para llegar a una conclusión precisa del estudio.
<b>Hospital</b>	Tiene pacientes destinados a la atención y asistencia de sus enfermedades.
<b>Paciente</b>	Se realiza la biopsia para examinar la presencia de alguna enfermedad.
<b>Control de calidad</b>	Se realiza a los datos de las biopsias almacenadas permitiendo la uniformidad en la información, criterios diagnósticos consensuados, profundización en la correlación clínico-patológica, además de la posibilidad de evaluar la calidad del diagnóstico.
<b>Clínico</b>	Es el encargado de solicitar el diagnóstico clínico al paciente para tener un diagnóstico más concluyente.
<b>Codificación</b>	Se les realiza a los diagnósticos de las enfermedades según la CIE.
<b>Diagnóstico</b>	Se evidencian y se identifica la enfermedad o cualquier estado de salud.
<b>Diagnóstico clínico</b>	Contiene el diagnóstico ya que el mismo describe el diagnóstico clínico del paciente.
<b>Diagnóstico macroscópico</b>	Contiene el diagnóstico que describe el estado de cada uno de los componentes internos de los órganos y sistemas del interior de la persona
<b>Diagnóstico microscópico</b>	Contiene el diagnóstico donde se definen la enfermedad del paciente y se explican los procesos que la describen, mediante un análisis del diagnóstico macroscópico y la historia clínica.

## 2.2 . INGENIERÍA DE REQUISITOS DEL SISTEMA

La ingeniería de requisitos del sistema especifica qué es lo que el sistema debe hacer, sus funciones y sus propiedades esenciales y deseables. Como en el análisis de requerimientos del software, crear definiciones de requerimientos del sistema requiere consultar con los clientes del sistema y con los usuarios finales (Ian, 2011).

### 2.2.1. OBTENCIÓN DE REQUISITOS

La obtención de requisitos determina el dominio de la aplicación, se especifican los servicios que debe proveer el sistema, la funcionalidad requerida del sistema y las restricciones de hardware y software. Es indispensable la participación de los usuarios y clientes para la identificación de los requerimientos del sistema (Mejía Alvarez, 2009).

La captura de requisitos es la etapa de mayor interacción con el usuario, en ella el equipo de desarrollo busca comprender las necesidades que debe cubrir el sistema, apoyándose en toda fuente de información disponible (Báez, 2001). A continuación, se describen las técnicas empleadas en la captura de requisitos.

La **entrevista** es una técnica muy utilizada, y requiere una mayor preparación y experiencia por parte del analista. La entrevista se puede definir como un “intento sistemático de recoger información de otra persona” a través de una comunicación interpersonal que se lleva a cabo por medio de una conversación estructurada. Debe quedar claro que no basta con hacer preguntas para obtener toda la información necesaria (Guerra, 2019a), es muy importante la forma en que se plantea la conversación y la relación que se establece en la entrevista. Su aplicación permite obtener una amplia visión de las necesidades del usuario, proporcionando una mejor comprensión de los objetivos de la solución buscada.

Se realizaron entrevistas a especialistas de los departamentos de Anatomía Patológica, en especial a: Dr. Cs. José Hurtado De Mendoza Amat profesor consultante de Anatomía patológica, Dra. Cs. Teresita de Jesús Montero González Especialista de segundo del Instituto Superior de Medicina Militar “Dr. Luis Díaz Soto y la Dra. Janet Lamadrid García Especialista de Anatomía Patológica del Instituto Nacional de Oncología y Radiobiología. Inicialmente se realizaron entrevistas no estructuradas, para un conocimiento general del dominio del problema, seguidamente se hizo uso de una combinación de entrevistas estructuradas y no estructuradas con el fin de profundizar en los conocimientos adquiridos.

Las entrevistas realizadas arrojaron información valiosa acerca de las necesidades y perspectivas de utilización del sistema, resaltando la importancia de desarrollar una aplicación sencilla, amigable y fácil de utilizar, con un modo de operación similar al sistema en explotación. Se puso en evidencia la necesidad de restringir el acceso a la información, de brindar un informe de biopsia de mayor calidad y permitir la codificación de un número ilimitado de diagnósticos.

Los especialistas plantearon la necesidad de que el módulo de biopsia se implementen las principales funcionalidades del sistema en explotación, entre ellas: la gestión de los datos de las biopsias, la especificación de criterios para el muestreo de los datos (con el objetivo de realizar estudios sobre los mismos), la emisión de listados configurables de biopsias y el desglose de valores por atributos almacenados, así como la realización del control de calidad interno y la clasificación de la correlación clínico-patológica.

Por otro lado, **la tormenta de ideas** es un proceso interactivo no estructurado de grupo, donde las opiniones individuales se analizan y estudian en colectivo. Esta técnica se puede utilizar para identificar un primer conjunto de requisitos en aquellos casos donde no están muy claras las necesidades que hay

que cubrir, o cuando se está creando un sistema que habilitará un servicio nuevo para la organización (Guerra, 2019a).

En varias sesiones, se desarrollaron tormentas de ideas con la participación del equipo de desarrollo usuarios potenciales del sistema. Como resultado, se logró generar una amplia gama de opiniones sobre los requisitos a satisfacer por el software, así como las limitaciones de la solución existente. Entre ellas:

- Ampliar la codificación de los diagnósticos con la utilización de los ejes morfológico y topográfico, para mayor especificación de la Clasificación Internacional de Enfermedades, o alternativamente.
- Posibilitar el uso de comodines (prefijos y sufijos) para la codificación de los diagnósticos.
- Permitir la introducción de descripciones macroscópicas del paciente por aparato o sistema.
- Facilitar la búsqueda del paciente por nombres parciales o parecidos.
- Enriquecer los reportes realizados con el empleo de recursos gráficos que propicien una mejor visualización y análisis de los resultados.
- Permitir la modificación de forma controlada de los diccionarios utilizados para la codificación de los diagnósticos.
- Permitir la configuración de las reglas a verificar como parte del control interno de calidad y las acciones del sistema ante el incumplimiento de las mismas.

Estas técnicas se complementaron con el análisis de tendencias y tecnologías más comunes en el desarrollo de este tipo de sistemas, así como el estudio de la documentación referente al tema.

### 2.2.2. REQUISITOS FUNCIONALES DEL SISTEMA

A partir del análisis de los procesos de negocio surgen las funcionalidades que el sistema debe incorporar para dar solución al problema identificado. Estas funcionalidades son especificadas mediante los requisitos funcionales del sistema que son capacidades o condiciones que el sistema debe cumplir y que definen el comportamiento interno del software. Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, la forma en que debe reaccionar ante ciertas entradas y cómo se debe comportar en situaciones particulares (Ian, 2011). Además, aporta una visión más detallada de lo que se va a implementar.

A continuación, se enuncian los requisitos funcionales identificados en la propuesta de solución:

- **RF#1** Adicionar paciente.
- **RF#2** Buscar paciente.
- **RF#3** Visualizar paciente.
- **RF#4** Modificar paciente.

- **RF#5** Adicionar biopsia.
- **RF#6** Buscar biopsia.
- **RF#7** Visualizar biopsia.
- **RF#8** Modificar biopsia.
- **RF#9** Adicionar impresión diagnóstica clínica.
- **RF#10** Visualizar impresión diagnóstica clínica.
- **RF#11** Modificar impresión diagnóstica clínica.
- **RF#12** Adicionar diagnóstico anatomopatológico.
- **RF#13** Modificar diagnóstico anatomopatológico.
- **RF#14** Visualizar diagnóstico anatomopatológico.
- **RF#15** Filtrar biopsias según criterios especificados.
- **RF#16** Exportar datos de biopsias.
- **RF#17** Importar datos de biopsias.
- **RF#18** Emitir listado configurable de biopsia.
- **RF#19** Adicionar reporte dinámico.
- **RF#20** Modificar reporte dinámico.
- **RF#21** Visualizar reporte dinámico.
- **RF#22** Exportar reporte dinámico.

### 2.2.3. REQUISITOS NO FUNCIONALES DEL SISTEMA

Los requisitos no funcionales son limitaciones sobre servicios o funciones que ofrece el sistema. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema (Ian, 2011). Los requisitos no funcionales van a ser adoptados de los requisitos definidos para el proyecto Desarrollo del SIRCAP en la documentación del producto (Arango Astorga & Cabrera Nicolau, 2014). A continuación, se enuncian los requisitos no funcionales identificados en la propuesta de solución:

#### Requisitos de hardware:

- **RNF-01** Recursos tecnológicos.

El análisis de la base tecnológica presente en los departamentos de Anatomía Patológica del país, arrojó que el nuevo sistema deberá operar con prestaciones mínimas de hardware:

1. Procesador: 1.6 GHz. o superior.
2. Memoria RAM: 512 MB o superior.
3. Disco duro: 40 GB o superior.

### Requisitos de software:

- **RNF-02** Entorno de ejecución.

Para la ejecución de la aplicación, con independencia del sistema operativo, constituyen premisa indispensable las siguientes instalaciones:

1. Entorno de ejecución de Java (JRE), versión 1.8 o superior.
2. Sistema gestor de bases de datos PostgreSQL, versión 9.2 o superior.

### Requisitos de interfaz:

- **RNF-03** Interfaz de usuario.

El sistema debe ofrecer una interfaz amigable, intuitiva y fácil de operar, con un modo de operación lo más cercano posible al del software en explotación. Además, debe mantener la línea de diseño establecida para alcanzar la uniformidad en la solución final.

### Requisitos de seguridad:

- **RNF-04** La seguridad estará regida por un mecanismo de control de acceso basado en roles, convenientemente asignados a los usuarios del sistema al momento de su creación.
- **RNF-05** Se establecerá un mecanismo de autenticación para el acceso al sistema.
- **RNF-06** Las contraseñas no podrán ser almacenadas en texto plano, o en espacios que permitan su acceso o modificación por personas no autorizadas.
- **RNF-07** El sistema debe preservar la seguridad de la información, garantizando en todo momento su confidencialidad, integridad y disponibilidad.

## 2.2.4. ESPECIFICACIÓN DE REQUISITOS FUNCIONALES

La especificación de requisitos establece la base para el acuerdo entre usuarios y desarrolladores de software, quedando definido el comportamiento deseado del producto (IEEE, 2004). En el entorno de XP, las historias de usuario constituyen el artefacto utilizado para describir las funcionalidades del sistema. Contienen una breve descripción del comportamiento del sistema desde la perspectiva del usuario y representan un medio de comunicación entre el mismo y el equipo de desarrollo. A continuación, se presentan las historias de usuario correspondientes a tres de los requisitos funcionales identificados.



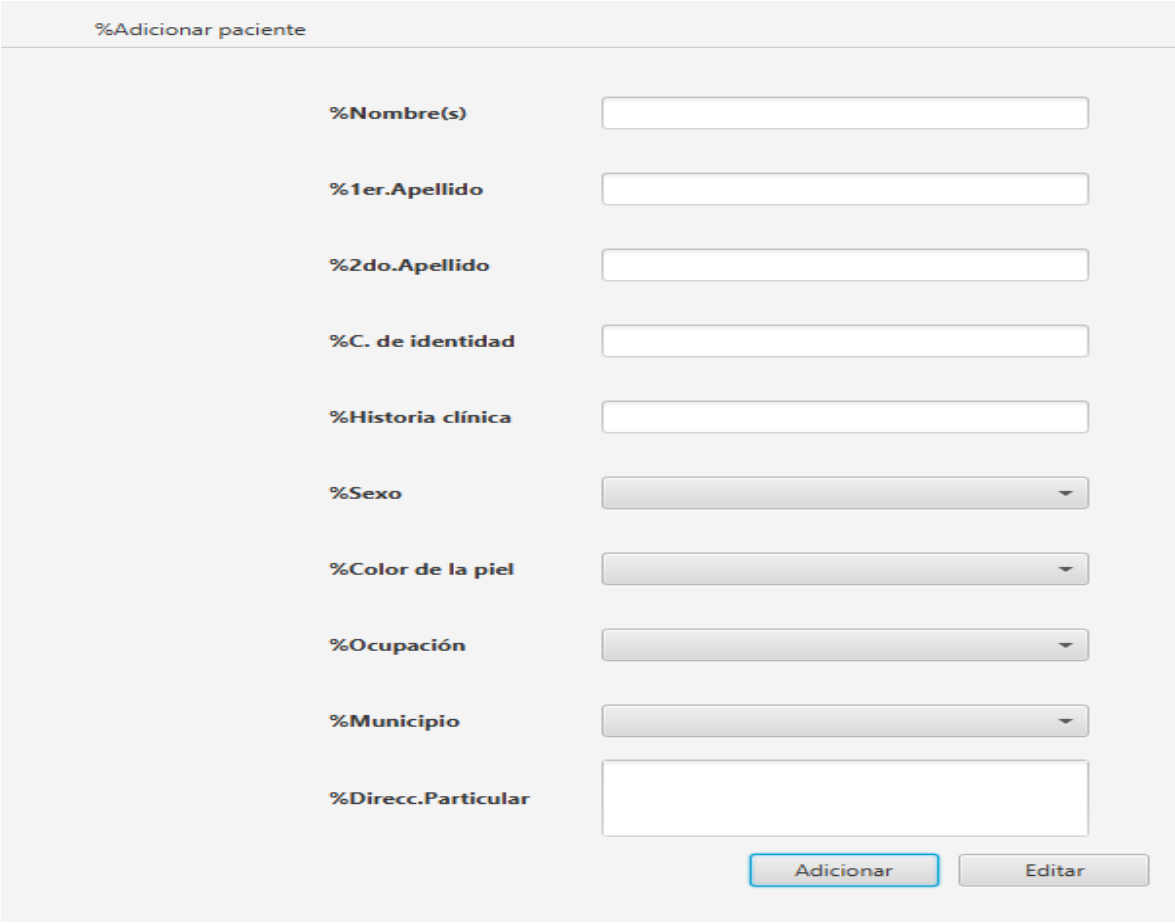
**Tabla 4:**Historia de usuario: Registrar biopsia

HISTORIA DE USUARIO	
<b>Número:</b> HU-5	<b>Usuario:</b> Todos
<b>Nombre de historia:</b> Adicionar biopsia.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Rachel Avila Milord.	
<b>Descripción:</b> Permite registrar los datos generales asociados a la biopsia.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz de usuario:</b>	

**Tabla 5:** Historia de usuario: Buscar paciente.

HISTORIA DE USUARIO	
<b>Número:</b> HU-6	<b>Usuario:</b> Todos
<b>Nombre de historia:</b> Buscar biopsia.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Rachel Avila Milord.	
<b>Descripción:</b> Muestra la o las biopsias registrada en el sistema que cumplen criterios de búsqueda previamente establecidos, permitiendo modificar los datos si fuera necesario.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz de usuario:</b>	

Tabla 6: Registrar paciente.

HISTORIA DE USUARIO	
Número: HU-6	Usuario: Todos
Nombre de historia: Adicionar paciente.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Rachel Avila Milord.	
Descripción: Permite registrar los datos generales de los pacientes.	
Observaciones:	
Prototipo de interfaz de usuario:	
 <p>The screenshot shows a web form titled "%Adicionar paciente". It contains the following fields and controls:</p> <ul style="list-style-type: none"> <li>%Nombre(s): Text input field</li> <li>%1er.Apellido: Text input field</li> <li>%2do.Apellido: Text input field</li> <li>%C. de identidad: Text input field</li> <li>%Historia clínica: Text input field</li> <li>%Sexo: Dropdown menu</li> <li>%Color de la piel: Dropdown menu</li> <li>%Ocupación: Dropdown menu</li> <li>%Municipio: Dropdown menu</li> <li>%Direcc.Particular: Text input field</li> <li>Buttons: "Adicionar" (highlighted in blue) and "Editar"</li> </ul>	

### 2.2.5. VALIDACIÓN DE REQUISITOS FUNCIONALES

La validación de requisitos es un proceso más general. Se debe asegurar que el software cumple las expectativas del cliente y los usuarios finales, representa un paso de gran importancia para la obtención

de un sistema que verdaderamente satisfaga las necesidades planteadas. Los prototipos pueden ser costosos, pero evitan el despilfarro de recursos al intentar satisfacer requisitos equivocados (IEEE, 2004). Fue desarrollado un prototipo por cada una de las historias de usuario, encontrándose los mismos dentro de la descripción de cada una.

### 2.3 . PLAN DE ITERACIONES

Una vez identificados los requisitos y descritas las historias de usuario correspondientes, el equipo de desarrollo realiza una estimación del esfuerzo que supone requerirá la implementación de cada una, mientras el cliente define la prioridad en función del valor para el negocio. Finalmente, se llega a un acuerdo sobre el orden de implementación y el contenido de las entregas, apostando por enfrentar las historias de más valor y riesgo lo antes posible (Letelier, 2003).

El ciclo de desarrollo de software guiado por XP se caracteriza por ser iterativo e incremental. Los niveles de prioridad para el negocio y el esfuerzo estimado de implementación para cada una de las historias de usuario especificadas, hicieron posible la planificación de la implementación del sistema, quedando definidas cuáles historias de usuario serían implementadas en cada iteración.

**Tabla 7:** Esfuerzo estimado por iteraciones.

No. Iteración	Historia de usuario	Prioridad	Esfuerzo estimado (Horas)	
1	Adicionar paciente.	Alta	5	58
	Adicionar biopsia.	Alta	3	
	Adicionar impresión diagnóstica clínica.	Alta	8	
	Adicionar diagnóstico anatomopatológico.	Alta	8	
	Emitir listado configurable de biopsias.	Media	13	
	Filtrar biopsias según criterios especificados.	Alta	21	
2	Visualizar paciente.	Media	3	21
	Modificar biopsia.	Media	3	
	Visualizar impresión diagnóstica clínica.	Media	3	
	Visualizar diagnóstico anatomopatológico.	Media	3	
	Modificar diagnóstico anatomopatológico.	Media	3	
	Visualizar biopsia.	Media	3	
	Buscar biopsia.	Media	3	
3	Buscar paciente.	Media	3	

Exportar datos de biopsias.	Baja	3	40
Importar datos de biopsias.	Baja	3	
Adicionarreportedinámico.	Alta	13	
Modificar reporte dinámico.	Media	4	
Visualizar reporte dinámico	Media	13	
Exportar reporte dinámico	Media	4	

## 2.4 . ARQUITECTURA DE LA SOLUCIÓN

Para el desarrollo del Módulo de biopsia y teniendo en cuenta la tecnología propuesta, se define como parte de la línea base de la arquitectura la implementación del patrón de diseño de arquitectura de *software* Modelo-Vista-Controlador. Este patrón arquitectónico permite la separación de los datos de una aplicación, la interfaz de usuario y la lógica de control, en tres componentes distintos: el modelo, que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia; la vista, que compone la información que se envía al cliente y los mecanismos de interacción con éste; y el controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno(Alicante, 2014).

- El **Modelo** administra el comportamiento y los datos del dominio de aplicación.
- La **Vista** define las interfaces de usuario. Posibilita la interacción del usuario y muestra la información obtenida por el modelo del medio persistente.
- El **Controlador** responde a eventos que se traducen en solicitudes de servicio para el modelo o la vista, gestionando la interacción entre el usuario y los datos.

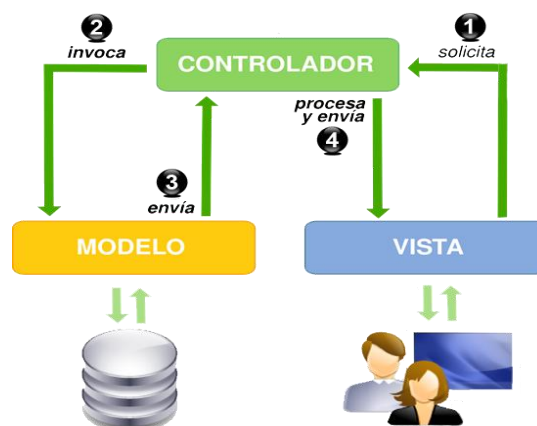


Figura 2:Modelo Vista Controlador. Fuente:(Arango Astorga & Cabrera Nicolau, 2014).

## 2.5.MODELO DE DISEÑO

La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo desarrollar(Valladarez Mellendez , Gailtan & Pérez Reyes, 2016).Uno de los principios fundamentales de la metodología XP es la simplicidad, por lo que en la etapa de diseño se propone el empleo de TarjetasClase-Responsabilidad-Colaboración(CRC), en lugar de diagramas de clases para la descripción del sistema en notación UML(Arango Astorga & Cabrera Nicolau, 2014).

Las Tarjetas CRC, permiten conocer que clases componen el sistema y cuales interactúan entre sí. Se dividen en tres secciones: Nombre de la Clase, Responsabilidades y Colaboradores(Valladarez Mellendez , Gailtan & Pérez Reyes, 2016).Su utilización en el diseño potencia el uso de patrones de asignación de responsabilidades(Arango Astorga & Cabrera Nicolau, 2014).A continuación, se muestran las tarjetas CRC correspondientes a tres de las clases controladoras presentes en la propuesta de solución (Tabla 8,9 y 10).

**Tabla 8:**Tarjeta CRC:BuscarPersonaController.java.

TARJETA CRC	
<b>Clase:</b> BuscarBiopsiaController.java	
<b>Súper Clase:</b> AnchorPane.java	
<b>Responsabilidades:</b>	<b>Colaboraciones:</b>
Recuperar las biopsias que cumplan los criterios de búsqueda establecidos.	BiopsiaPaciente.java ProtocoloController.java

**Tabla 9:** Tarjeta CRC:ProtocoloController.java.

TARJETA CRC	
<b>Clase:</b> ProtocoloController.java	
<b>Súper Clase:</b> AnchorPane.java	
<b>Responsabilidades:</b>	<b>Colaboraciones:</b>
Registrar los datos relacionados con la autopsia.	Biopsia.java DiagnosticoClinicoController.java
Modificar los datos relacionados con la autopsia.	DiagnosticosMacroscopicosController.java DiagnosticosAnatomopatologicosController.java

**Tabla 10:** Tarjeta CRC: RegistrarPacienteController.java.

TARJETA CRC	
<b>Clase:</b> AdicionarPacienteController.java	
<b>Súper Clase:</b> AnchorPane.java	
<b>Responsabilidades:</b>	<b>Colaboraciones:</b>
Registrar los datos relacionados con la persona.	RegistrarPaciente.java ProtocoloController.java

## 2.6. PATRONES DE DISEÑO

Según (Gammay otros, 2004) un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Los patrones de diseño constituyen soluciones genéricas probadas a problemas comunes del desarrollo de software, propician flexibilidad, elegancia y reutilización. Para un mejor diseño de la solución, se hace uso de los siguientes patrones:

### 2.6.1. PATRÓN DAO

EL patrón Objeto de Acceso a Datos (*Data Access Object, DAO*) es un objeto que encapsula y abstrae el acceso a datos de un almacén persistente o externo del sistema (Alur, Crupi & Malks, 2003). Se utilizó como interfaz para las operaciones de persistencia relacionadas con una entidad particular (métodos para la creación, lectura, modificación, eliminación y búsqueda).

Los objetos del negocio acceden a los DAO a través de sus interfaces, siendo independientes de cualquier implementación en específico. Posibilita la creación de una arquitectura desacoplada y un modelo escalable, facilitando el tratamiento centralizado de las transacciones (Arango Astorga & Cabrera Nicolau, 2014).

### 2.6.2. PATRONES GOF

De los patrones popularizados por la autoproclamada “Pandilla de los Cuatro” Gang of Four (GoF) (Gammay otros, 2004) fueron utilizados los siguientes:

- **Método de fabricación (*Factory Method*):** su propósito es definir una interfaz para crear objetos donde se delega la creación de las instancias a las subclases (Gammay otros, 2004). Se empleó

como complemento al patrón DAO, centralizando en una clase (DAOHibernateFactory.java) la responsabilidad de instanciar las implementaciones DAO particulares del ORM Hibernate, evitando la dependencia del mismo en las clases controladoras.

- **Instancia única (Singleton):** diseñado para que sólo se cree una instancia de la clase y provee un punto global de acceso a ésta. Este patrón es útil cuando se quiere tener solamente un objeto único instanciado (Gamma et al., 2004). Se utilizó para garantizar la existencia de una única instancia de la clase SessionFactory.java, canal único de comunicación entre la base de datos y la aplicación, garantizando un mecanismo de acceso global a dicha instancia.
- **Fachada (Facade):** provee una interfaz unificada de alto nivel orientada a facilitar el uso del sistema (Gamma et al., 2004). Empleado en la interfaz principal (SIRCAPController.java).
- **Visitante (Visitor):** es un patrón de comportamiento que permite definir una operación sobre objetos de una jerarquía de clases sin necesidad de modificarla (Gamma et al., 2004). Se empleó para separar los algoritmos del control interno de calidad de la estructura de las entidades persistentes involucradas, unificándolos en una sola clase creada a tales efectos.

### 2.6.3. PATRÓN GRASP

Como patrones generales de asignación de responsabilidades (*General Responsibility Assignment Software Patterns, GRASP*) fueron utilizados los siguientes:

- **Experto:** es el encargado de asignar responsabilidad al experto en información, la clase que tiene la información necesaria para realizar la responsabilidad (Larman, 1999a). El patrón resulta de utilidad en las clases del modelo, las cuales contienen toda la información relacionada con los objetos persistentes que representan.
- **Creador:** el patrón creador es una especialización del patrón experto. Este patrón ayuda a guiar la decisión de quién debe crear un objeto de otra clase (Larman, 1999a). En la solución propuesta el patrón creador sugiere encontrar clases de objetos que estén vinculadas (o que se conozcan) para hacerlas responsables de la creación de los objetos, manteniendo así un bajo acoplamiento. Se usa en la clase SARCAPController.java, la cual instancia las clases encargadas de cada una de las funcionalidades accesibles desde la interfaz principal.
- **Controlador:** propone asignar la responsabilidad de recibir o manejar un mensaje de evento del sistema a una clase (Larman, 1999a). En la arquitectura definida este patrón se evidencia en las clases controladoras, responsables de implementar todas las funcionalidades pertenecientes a una interfaz determinada.



- **Bajo acoplamiento:** este patrón debe tratar de mantener el nivel de acoplamiento bajo minimizando el conocimiento que unas clases deben tener de otras (Larman, 1999a).
- **Alta cohesión:** es el encargado de que la información que contiene una clase debe de ser coherente, altamente relacionada con dicha clase y que la misma no realice una excesiva cantidad de trabajo(Larman, 1999a).

La utilización de ambos patrones se ve reflejado en todo el diseño realizado, ayudado por la utilización del patrón de arquitectura de la solución. El acoplamiento mide la fuerza con que una clase está conectada a otra, por lo tanto, mantenerlo bajo garantiza una menor dependencia e impacto de cambio entre las clases. Por otro lado, la cohesión se refiere al grado de coherencia de la información de una clase, la cual debe estar altamente relacionada con ella, pero no debe cargarla de trabajo. El uso del patrón MVC va a permitir una organización dentro de las clases, teniendo las interfaces en la vista, las controladoras en el controlador y las clases persistentes en el modelo, de tal forma que, si se realiza un cambio en una de las capas, las otras se ven afectadas lo menos posible, lo que garantiza un bajo acoplamiento. Esta organización también permite que las clases con responsabilidades afines se encuentren dentro de la misma capa, estando definida con claridad la información que gestiona cada una. De este modo se garantiza una mejor claridad y facilidad de comprensión del diseño, simplificándose el mantenimiento y la introducción de mejoras en el sistema.

## 2.7. MODELO DE DATOS

Un modelo de datos es un conjunto de conceptos utilizados para organizar los datos de interés y describir su estructura en forma comprensible para un sistema informático. Constituye una definición lógica y abstracta de los objetos y operadores que en conjunto constituyen la máquina abstracta con la que interactúan los usuarios. Este modelo proporciona una representación visual y física de los datos persistentes del sistema, que en el futuro serán la base de datos. Los objetos permiten modelar la estructura de los datos y los operadores admiten modelar su comportamiento. Se obtiene a partir del diagrama de clases persistentes y su forma se expresa mediante un diagrama de UML, siendo sus elementos esenciales las entidades, atributos y relaciones entre entidades(Rodríguez Bacallao, 2019).El Modelo Entidad-Relación (MER) asociado a la Propuesta de solución está compuesto por 49 tablas. El mismo puede ser consultado en el **(Anexo 3)**.

Para la realización del MER se hizo uso de diferentes patrones de diseño de bases de datos; los cuales permitieron la creación de una base de datos fortalecida. Se utilizó el patrón Árbol fuertemente codificado para modelar la estructura de la división política administrativa del país, conformada por la jerarquía n país, n provincia, n municipio como se puede observar en la figura 3.



Figura 3: Patrón árbol fuertemente codificado.

Por otro lado, el patrón Árbol simple permitió la representación de la Clasificación Internacional de Enfermedades (CIE-10) en la entidad neje\_cie10, la cual almacena elementos de igual tipo entre los que existe una jerarquía padre-hijo como se puede observar en la Figura 4.



Figura 4: Patrón árbol simple.

Con el patrón Llave subrogada, se evitó el uso de identificadores del negocio en las entidades como se puede observar en la Figura 3.



Figura 5: Patrón llave subrogada.

Finalmente, se utilizó el patrón Control de acceso basado en roles (RBAC, por sus siglas en inglés) para administrar el acceso de los usuarios a las diferentes funcionalidades del sistema.

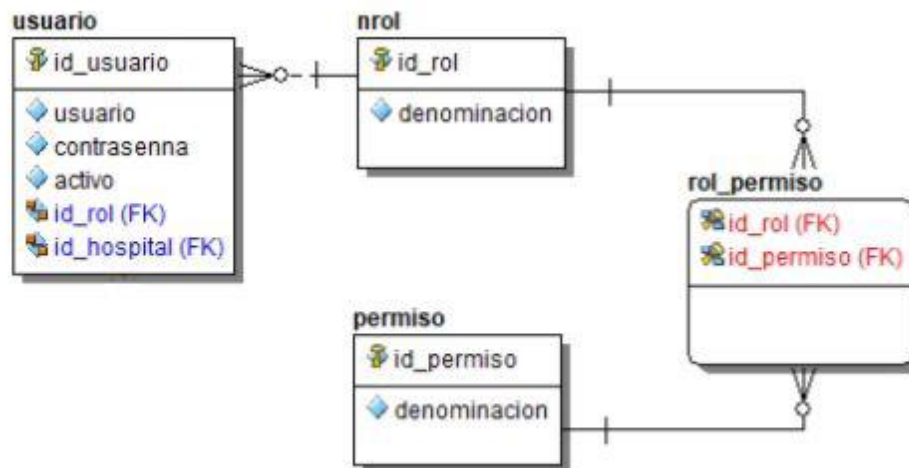


Figura 6: Patrón control de acceso basado en roles.

## 2.8. CONCLUSIONES PARCIALES

- El uso de la metodología XP permitió definir los elementos esenciales para la construcción de la propuesta de solución.
- El uso de un modelo conceptual permitió obtener una representación visual del problema que ayuda en la comprensión del mismo.
- Se especificaron y validaron los requisitos del sistema, realizando una cuidadosa planificación de su implementación en función del esfuerzo necesario y su prioridad para el negocio.
- Se fundamentó la elección del patrón Modelo Vista Controlador para la arquitectura del sistema permitiendo un diseño e implementación adecuado del Módulo de Biopsia.
- El uso de patrones de diseño contribuyó a la obtención de un producto final de mayor calidad.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

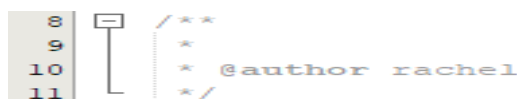
Una vez detallada la propuesta de solución se procedió a su ejecución. El presente capítulo precisacada uno de los pasos que desde ese momento tuvo lugar en el afán de continuar y reforzar elaseguramiento de la calidad, iniciado desde la misma planeación del proceso de software y se describe los estándares de codificación tomados en consideración. Se ofreceun análisis de cada uno de los niveles de pruebas ejecutados, los términos en que se realizaron laspruebas y los resultados alcanzados. Finalmente, se expone un conjunto de reportes generados porel sistema, evidenciando su utilidad para el análisis de morbimortalidad.

### 3.1. ESTÁNDARES DE CODIFICACIÓN

Los estándares de codificación son reglas que se aplican para lograr uniformidad en el código producido por un grupo de desarrollo de un sistema. Estos reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores. Los estándares de codificación no destapan problemas existentes, evitan más bien que los errores ocurran, lo que permite obtener un código de alta calidad(Rodríguez Bacallao, 2019).En función de la homogeneidad en la implementación, se definió un conjunto de pautas a seguir durante la codificación.

- **Cabecera de archivos**

Todos los archivos deben iniciar con una cabecera que especifique el autor de los últimoscambios, adicionalmente se podrán incluir comentarios u otros datos de interés.En la figura 7 se puede observar el uso de este estándar.



```
8  
9  
10 /*  
11 */ @author rachel
```

Figura 7: Cabecera de archivos.

- **Definición de clases**

Las clases serán colocadas en archivos independientes que solo contendrán el código de lamisma. Se utilizará el estilo de codificación “Upper Camel Case”, el cual establece que losnombres iniciarán con letra mayúscula y de poseer más de una palabra, la primera letra decada una deberá ser también mayúscula. No se permiten letras mayúsculas sucesivas amenos que se trate de siglas conocidas en el dominio del sistema. En la figura 8 se puede observar el uso de este estándar.

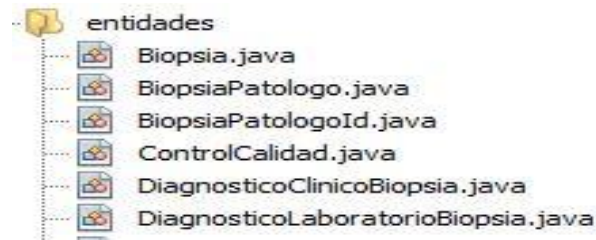


Figura 8: Definición de clases.

- **Definición de variables y constantes**

Los nombres de las variables deben ser descriptivos y concisos. No se usarán grandes frases ni pequeñas abreviaciones. Se utilizará el estilo de codificación “Lower Camel Case”, Los nombres iniciarán con letra minúscula y cada nueva palabra debe iniciar con mayúscula. Las constantes se escribirán en su totalidad con mayúscula, usando un guión bajo para la separación de palabras. En la figura 9 se puede observar el uso de este estándar. En la figura 9 se puede observar el uso de este estándar.

```
12     private String idBiopsia;  
13     private String numeroBiopsia;  
14     private DiagnosticoClinicoBiopsia diagnosticoClinicoBiopsia;
```

Figura 9: Definición de variables y constantes.

- **Definición de funciones**

Los nombres de las funciones deben dar una idea clara del objetivo con el que fueron concebidas. Al igual que para las variables se utilizará el estilo “Lower Camel Case”. En la figura 10 se puede observar el uso de este estándar.

```
34     public void setDiagnosticoClinicoBiopsia(DiagnosticoClinicoBiopsia diagnosticoClinicoBiopsia) {  
35         this.diagnosticoClinicoBiopsia = diagnosticoClinicoBiopsia;  
36     }
```

Figura 10: Definición de funciones.

- **Uso de llaves en bloques de instrucciones**

Todo bloque de instrucciones debe ir entre llaves, aun cuando conste de una sola línea. En la figura 11 se puede observar el uso de este estándar.

```
16     if (paciente.getEdad() != null) {  
17         edad.set(String.valueOf(paciente.getEdad()));  
18     }
```

Figura 11: Uso de llaves en bloques de instrucciones.

- **Posición de las llaves en bloques de instrucciones**

Las llaves de apertura se colocarán al final de la sentencia que delimita el bloque de instrucciones, las de cierre se alinean con el inicio de la sentencia en una nueva línea. En la figura 12 se puede observar el uso de este estándar.

```
23 | | | | if (flag) {  
24 | | | |     for (Object o : items) {  
25 | | | |         //instrucciones  
26 | | | |     }  
27 | | | | }
```

Figura 12: Posición de las llaves en bloque de instrucciones.

- **Espacios entre operadores**

Para mejor legibilidad del código, se colocarán espacios a ambos lados de los operadores. En la figura 13 se puede observar el uso de este estándar.

```
30 | | | | for (int i = 0; i < items.size(); i++) {  
31 | | | |     a = i;  
32 | | | | }
```

Figura 13: Espacio entre operadores.

## 3.2. PRUEBAS DE UNIDAD

Para las pruebas de unidad, se recurrió a la utilización del conjunto de bibliotecas JUnit, escritas en Java y desarrolladas por Kent Beck, principal exponente de la metodología XP; con el objetivo de facilitar la creación y ejecución de pruebas unitarias según las características propuestas por XP.

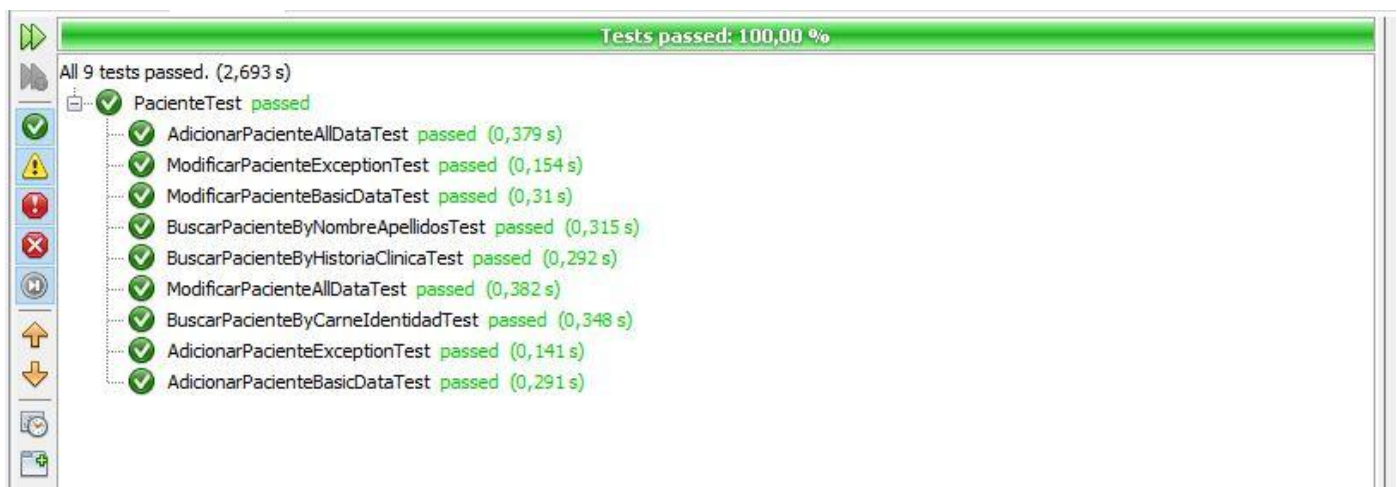
**JUnit** constituye un marco de pruebas que utiliza anotaciones para identificar métodos que especifican una prueba. Es un proyecto de código abierto alojado en **GitHub**, además de una instancia de la arquitectura **xUnit** para marcos de trabajo enfocados en pruebas de unidad (Lars, 2019). De conjunto con sus complementos para las principales herramientas de desarrollo (NetBeans entre ellas) facilita la codificación y ejecución de las pruebas con un esfuerzo mínimo (Arango Astorga & Cabrera Nicolau, 2014).

Este marco de trabajo, provee las plantillas necesarias para la creación de las pruebas de unidad, permitiendo al programador enfocarse en los parámetros de las funcionalidades objeto de prueba y la respuesta esperada. Al finalizar, se muestra un informe con las funcionalidades que superaron las pruebas y las que no, debiendo ser corregidas hasta lograr 100 % de efectividad.

En la solución particular que se propone, la realización de pruebas de unidad se vio dificultada por las características de la mayor parte de las funcionalidades de clase, al estar orientadas a operaciones sobre

la base de datos y no poseer valores de retorno (métodos void), siendo necesario en la mayoría de las clases probar sus funcionalidades de manera agrupada. Por otro lado, el encapsulamiento implícito en la programación orientada a objetos representó un obstáculo que debió ser saldado antes de poder codificar y ejecutar las pruebas.

Las dificultades antes descritas, impidieron la realización de pruebas a cada una de las clases, seleccionándose aquellas más relevantes para la solución, especialmente las asociadas a la gestión de los datos demográficos y hospitalarios, los diagnósticos clínicos, macroscópicos y anatomopatológicos; así como las clases involucradas en la realización del control interno de calidad y el análisis de la correlación diagnóstica. La figura 14 muestra los resultados de las pruebas de unidad aplicadas a la clase que agrupa los algoritmos asociados al control de calidad.



**Figura 14:**Resultados de las pruebas de unidad aplicadas a la clase ControlCalidadServicios.java.

La utilización de JUnit hizo posible la ejecución controlada de las principales funcionalidades por clase, permitiendo su evaluación acorde al comportamiento esperado. Para lograr un 100 % de efectividad en cada prueba, todos los errores se fueron corrigiendo en la medida en que se fueron revelando. La automatización de las pruebas facilitó su repetición luego de cada cambio importante en el código fuente, garantizando la efectividad de la integración entre los componentes.

## PRUEBAS DE ACEPTACIÓN

El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento (Cerillo, 2020). En consecuencia, y a diferencia de las pruebas de unidad, en XP las pruebas de aceptación no tienen por qué ser 100 % efectivas luego de cada iteración, las



correcciones o mejoras en las historias de usuario asociadas a pruebas de aceptación no superadas, forman parte de la planeación de la siguiente iteración (Arango Astorga & Cabrera Nicolau, 2014).

Las pruebas de aceptación permiten al cliente saber cuándo el sistema funciona y verifica que cumpla con sus expectativas, y que los programadores conozcan que es lo que resta por hacer (Cillero, 2020). Como fue explicado en el Capítulo 1 de la presente investigación para las pruebas de aceptación la metodología XP propone dos niveles de prueba, prueba alfa y prueba beta.

Las pruebas alfa se realizarán en un entorno controlado por el equipo de desarrollo, comprobando la correspondencia entre las funcionalidades del sistema y el comportamiento esperado. Finalizada cada iteración de desarrollo, se realizaron pruebas a las funcionalidades implementadas, validando y identificando no conformidades en otras. En las sucesivas iteraciones, además de las pruebas a las nuevas funcionalidades, se repitieron las pruebas anteriores aumentando su criticidad incorporando nuevos casos.

Se especificaron casos de prueba para todas las historias de usuario. A continuación, se presenta el diseño de un caso de prueba y un juego de datos empleado en la validación de la historia de usuario Adicionar biopsia requisito correspondiente a la primera iteración de desarrollo.

Tabla 11: Caso de prueba de aceptación: HU Adicionar biopsia.

CASO DE PRUEBA DE ACEPTACIÓN			
<b>Historia de usuario:</b>		HU-5: Adicionar biopsia.	
<b>Descripción:</b>		Permite validar el registro de los datos generales de la biopsia, correspondientes a la primera página del modelo de solicitud de biopsia.	
<b>Condiciones de ejecución:</b>		El usuario se autentica en el sistema.  En la interfaz principal, se selecciona el menú <b>Inicio</b> y seguidamente  <b>Introducir biopsia.</b>	
Escenarios de prueba:		Flujo del escenario:	Resultados esperados:
EP1	Registrar biopsia introduciendo datos	Se introducen datos válidos para una biopsia.	Se registran los datos.  El modelo de solicitud de

	válidos.	Se intenta pasar a la siguiente página del modelo de solicitud de biopsia.	biopsia pasa de página.
EP2	Registrar biopsia introduciendo datos no válidos.	Se introducen datos no válidos para una biopsia. Se intenta pasar a la siguiente página del modelo de solicitud de biopsia.	El sistema muestra una alerta sobre los datos no válidos. Se resaltan los campos no válidos. El modelo de solicitud de biopsia no pasa de página.
EP3	Registrar biopsia dejando campos obligatorios vacíos.	Se introducen datos dejando campos obligatorios vacíos. Se intenta pasar a la siguiente página del modelo de solicitud de biopsia.	El sistema muestra una alerta sobre los campos obligatorios que permanecen vacíos. Se resaltan los campos vacíos. El modelo de solicitud de biopsia no pasa de página.
EP4	Registrar biopsia previamente registrada en el sistema.	Se introducen los datos de una biopsia previamente registrada. Se intenta pasar a la siguiente página del modelo de solicitud de biopsia.	El sistema muestra una alerta sobre la biopsia previamente registrada. Se resalta el número de biopsia. El modelo de solicitud de biopsia no pasa de página.
EP5	Cancelar registrar biopsia.	Se introducen algunos datos de la biopsia. Se presiona el botón	El sistema muestra una alerta, solicitando confirmación para continuar. Se cierra el modelo de

	<b>Cancelar.</b>	solicitud de biopsia.
--	------------------	-----------------------

Tabla 12: Juego de datos de prueba: HU Registrar autopsia (V: válido, I: inválido).

de	JUEGO DE DATOS DE PRUEBA								
	Escenarios prueba	Biopsia	Fecha de realización	Ejecutado por	Fecha de recepción	Ejecutado por	Fecha de estudio macroscópico	Ejecutado por	Fecha de diagnóstico
EP1	<b>B20200001</b>	<b>10/05/2020</b>	<b>Juan</b>	<b>19/05/2020</b>	<b>Miguel</b>	<b>22/05/2020</b>	<b>Miguel</b>	<b>22/05/2020</b>	<b>Miguel</b>
	(V)	(V)	(V)	(V)	(V)	(V)	(V)	(V)	(V)
	<b>B20200002</b>	<b>11/05/2020</b>							
	(V)	(V)	(V)	(V)	(V)	(V)	(V)	(V)	(V)
EP2	<b>B2020</b>	<b>10/05/2020</b>							
	(I)	(V)	(V)	(V)	(V)	(V)	(V)	(V)	(V)
EP3		<b>10/05/2020</b>				<b>22/05/2020</b>	<b>Miguel</b>		
	(I)	(V)	(V)	(V)	(V)	(V)	(V)	(V)	(V)
	<b>B20200003</b>			<b>19/05/2020</b>	<b>Juan</b>				
	(V)	(I)	(V)	(V)	(V)	(V)	(V)	(V)	(V)
EP4	<b>B20200001</b>	<b>10/05/2020</b>	<b>Juan</b>	<b>19/05/2020</b>	<b>Miguel</b>	<b>22/05/2020</b>	<b>Miguel</b>	<b>22/05/2020</b>	<b>Miguel</b>
	(V)	(V)	(V)	(V)	(V)	(V)	(V)	(V)	(V)
EP5	<b>B20200010</b>	<b>10/05/2020</b>	<b>Juan</b>					<b>22/05/2020</b>	<b>Miguel</b>
	(V)	(V)	(V)	(V)	(V)	(V)	(V)	(V)	(V)

### 3.3. CONCLUSIONES PARCIALES

- La utilización de los estándares de codificación constituirá una guía para alcanzar un producto no solo funcional, sino de código legible y escalable.
- La realización de un adecuado proceso de pruebas durante el desarrollo de la aplicación, permitirá alcanzar un software de elevada calidad y comprobada satisfacción de los requisitos convenidos.
- Luego de cada cambio significativo en el sistema, se realizaron pruebas de unidad automatizadas a las principales funcionalidades por clases, exigiéndose un 100 % de efectividad para cada una.
- La descripción de las pruebas utilizadas para asegurar la calidad del software, permitirá obtener resultados satisfactorios, asegurando que el sistema implementado no contiene errores y tiene la aceptación requerida.

## CONCLUSIONES

- El estudio de la bibliografía y análisis de las tecnologías de la información y las comunicaciones aplicadas al área de Anatomía Patológica, demostró la necesidad de un módulo de biopsia para el sistema informático de registro y control en la búsqueda del mayor aprovechamiento de los estudios patológicos.
- La aplicación de la metodología XP, así como la incorporación de patrones de diseño, permitió realizar con calidad el análisis y diseño de la propuesta de solución sentando las bases para el proceso de implementación.
- La realización de pruebas de software permitió identificar y corregir los errores presentes en la solución, proporcionando mayor usabilidad, rendimiento y calidad a las funcionalidades del módulo desarrollado.
- El sistema informático obtenido responde a las condiciones de la red hospitalaria cubana, garantizando con su generalización, el estudio de un mayor número de casos en apoyo a la tomade decisiones.

## RECOMENDACIONES

Una vez concluida la investigación y el desarrollo de la propuesta de solución, la autora del presente trabajo recomienda:

- La mejora de la conectividad interdepartamental de los centros de salud cubanos, propiciaría la adaptación del sistema con un esfuerzo mínimo para su ejecución desde la web, aprovechando las potencialidades de la red para la transmisión de los datos en tiempo real a una base de datos centralizada.
- Con la extensión del soporte a nuevos métodos estadísticos, se podría obtener una mayor utilidad sobre los datos, a la vez que la incorporación de un módulo de Minería de Datos, permitiría el descubrimiento de conocimiento implícito en los mismos.
- Utilizar el Módulo de Biopsias para el Sistema Informático de Registro y Control en Anatomía Patológica en el Hospital Militar Dr. Luis Diaz Soto, con el objetivo de probar las funcionalidades desarrolladas en un entorno real.

## REFERENCIAS BIBLIOGRÁFICAS

1. Advance, R. C. 2019, *Patrones de Software Patrones de Software*, [<https://rjcodeadvance.com/patrones-de-software-que-es-patron-de-arquitectura-parte-3/>] (28/2).
2. Alicante, U. d. 2014, *Modelo vista controlador (MVC)*, [<https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>].
3. Alur, D., Crupi, J. & Malks, D. 2003, *Core J2EE™ Patterns: Best Practices and Design Strategies*, Prentice Hall PTR.
4. Amat, H. d. M., José 2014, 'Autopsia. Garantía de calidad en la Medicina', Editorial Ciencias Médicas, La Habana, p. 239.
5. Amat, J. H. d. M. 2019, 'Entrevista'.
6. Arango Astorga, P. & Cabrera Nicolau, L. 2014, *Sistema Informático de Registro y Control en Anatomía Patológica pra la Red Hospitalaria de Cuba* UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS.
7. Arango Astorga, P., Nicolau Cabrera, L. & Amat, J. H. d. M. 2019, 'Sistema informático para la gestión del conocimiento en los departamentos de anatomía patológica', *Revista Cubana Informática Médica*, vol. 2, pp. 48-64.
8. Báez, G. B. S. B. 2001, *Metodología DoRCU para la Ingeniería de Requerimientos*, Instituto Superior Politécnico Jose Antonio Echeverría.
9. Beck, K. & Andres, C. 2004, *Extreme Programming Explained: Embrace Change, Second Edition*, Addison Wesley Professional.
10. Bell, D. & Parr, M. 2003, *Java para estudiantes*, Pearson Editorial.
11. Borrajero Martínez, I. 2018, 'Despejando dudas', In *Revista Cubana Bohemia*, Cabrera Marieta, Salud.
12. Cap 2019, 'Anatomic pathology computer systems', In *CAP TODAY*, pp. 55-64.
13. Castaño , P., Ángel 2013, 'Información estructurada en Anatomía Patológica. Interoperabilidad con otros sistemas de información sanitaria.', in *Libro Blanco 2013 de la Anatomía Patológica en España*, ed. SEAP, S. E. d. A. P., Madrid, pp. 59-66.

14. Cerillo, M. 2020, *Pruebas de Aceptación*, [<https://manuel.cillero.es/doc/metrica-3/tecnicas/pruebas/aceptacion/>] (23/5).
15. Cillero, M. 2020, *Pruebas de aceptación* (23/5).
16. Ciri3n Mart3nez, G. & Herrera P3rez, M. 3. 2005, 'Anatom3a Patol3gica', In *Temas para enfermer3a*, p. 202.
17. Confluence, A. 2019, *SNOMED International Release Management Home*, [<https://confluence.ihtsdotools.org/pages/viewpage.action?pageId=38256207>] (28).
18. Date, C. J. 2001, *Introducci3n a los sistemas de base de datos*, Pearson Educaci3n, M3xico.
19. Fern3ndez Martori, M. 2018, 'Morbilidad por tumores malignos de piel por diagn3stico histopatol3gico. Col3n, Matanzas. 2010-2015', vol. 40.
20. Gamma, E., Helm, R., Johnson, R. & Vlissides, J. 2004, *Design Patterns Elements of Reusable Object-Oriented Software*.
21. Git 2020, *Git --distributed-even-if-your-workflow-isnt*, [<https://git-scm.com/>] (12-2).
22. Group, O. M. 2007, *About the Unified Modeling Language Specification Version 2.1.2.* , [<https://www.omg.org/spec/UML/2.1.2#specification-metadata>] (15 ).
23. Guerra, A. C. 2019a, *Obtenci3n de Requerimientos. T3cnicas y Estrategia*, [<https://sg.com.mx/revista/17/obtencion-requerimientos-tecnicas-y-estrategia>] (26).
24. Guerra, I. 2019b, *Libro Blanco 2019 de la Anatom3a Patol3gica en Espa3a*, Espa3a.
25. Guzm3n, P. F. & Messina, A. 2019, 'C3ncer de pr3stata, el problema del diagn3stico ¿Es la resonancia multiparam3trica de pr3stata la soluci3n?', pp. 60-66.
26. Hansen, G. W. & Hansen, J. V. 1997, *Dise3o y Administraci3n de Bases de Datos*, Prentice Hall.
27. Hardy, J., D , Griffin, J. C. & Rodriguez, J. A. 1961, *Manual para Biopsias*.
28. Heffelfinger, D. R. 2006, *JasperReports for Java Developers*, Packt Publishing.
29. Hidalgo R3os, N. & Ochoa Asse, R. 2014, 'Patolog3a General', Editorial Ciencias M3dicas, La Habana, p. 274.
30. Ian, S. 2011, *Ingenier3a del Software S3ptima Edici3n*
31. IEEE 2004, *Guide to the Software Engineering Body of Knowledge (SWEBOK)*.



32. ISO 2018, *¿Quiere saber lo que significa la gestión de calidad?*, [<https://www.nueva-iso-9001-2015.com/2018/10/quiere-saber-lo-que-significa-la-gestion-de-calidad/>] (29/2).
33. Kendall, K. E. & Kendall, J. E. 2005, *Análisis y diseño de sistemas*, Pearson Education.
34. Larman, C. 1999a, *UML y Patrones. Introducción al análisis y diseño orientado a objetos*, Prentice Hall, México.
35. Larman, C. 1999b, *UML y Patrones. Introducció n al anà lisis y diseño orientado a objetos*.
36. Lars, V. 2019, *Unit Testing with JUnit-Tutorial*, [<https://www.vogella.com/tutorials/JUnit/article.html>] (23/5).
37. Letelier, P. & Pénade , M. C. 2019, ' Métodologías ágiles para el desarrollo de software:eXtreme Programming (XP)', Universidad Politécnica de Valencia.
38. Letelier, P. P. M. C. 2003, *Métodologías Ágiles en el Desarrollo de Software: eXtreme Programming (XP)*, Universidad Politécnica de Valencia.
39. Marquina, E. & Parra, J. D. 2008, 'Guía de Patrones, Prácticas y Arquitectura .NET', ed. Microsoft.
40. Matellán Olivera, V. 2004, *Compilación de ensayos sobre software libre*, Dykinson, Madrid.
41. Mato García, R. M. 2005, *Sistemas de Base de Datos*, Pueblo y Educación, La Habana.
42. Medell, G., Mariana 2004, *AvanPat 1.1. Sistema de Información para Anatomía Patológica*, [<http://conganat.uninet.edu/6CVHAP/autores/trabajos/T067/index.html>] (14/11/2019).
43. Mejía Alvarez, D. P. 2009, 'INGENIERIA DE SOFTWARE.Obtención de Requerimientos.'
44. Mena, O. G., Benavides, R. & Freire Álvaro, N. A. 2017, 'Biopsia de mama con guía estereotáxica digital directa, revisión y actualidad', In *Federación Ecuatoriana de Radiología e Imagen*, vol. 10.
45. Oracle 2019a, *JavaFX Scene Builder*, [<https://www.oracle.com/technetwork/java/javase/downloads/javafxscenebuilder-info-2157684.html>] (12-2).
46. ORACLE 2019b, *Welcome to NetBeans*, [[https://netbeans.org/index\\_es.html](https://netbeans.org/index_es.html)] (20/11/2019).
47. Ottinger, J. B., Linwood & Minter 2016, *Beginning Hibernate: For Hibernate 5*, [<https://books.google.com.cu/books?id=kOt6DQAAQBAJ>].
48. Pablo, B. 2018, 'La terminología SNOMED CT'.

49. Paradigm, V. 2019, *Ideal Modeling & Diagramming Tool for Agile Team Collaboration.*, [<https://www.visual-paradigm.com/>] (10).
50. Pérez, A. 1958, *Biopsias*.
51. PostgreSQL 2020a, 'Acerca de PostgreSQL. In Comunidad de PostgreSQL'.
52. PostgreSQL 2020b, *What is PostgreSQL?*, [<https://www.postgresql.org/docs/10/static/intro-what-is.html>] (20/03/2020).
53. Pressman, R. 2005, *Ingeniería de Software. Un enfoque práctico*, Mc Graw Hill.
54. Rodríguez Bacallao, R. 2019, *Desarrollo de la hoja de consulta de la especialidad Cardiología para el módulo Consulta Externa del Sistema de Información Hospitalaria XAVIA HIS.*, Universidad de las Ciencias Informáticas.
55. Rodríguez Moreno, Á. M. 2014, '¿Análisis Complementarios vs. Método Clínico?', *Revista Habanera de Ciencias Médicas*, vol. 13, pp. 650-656.
56. Rouse, M. 2007, *What is integrate development environment (IDE)*, [<http://searchsoftwarequality.techtarget.com/definition/integrated-development-environment>] (07/11/2013).
57. Sánchez Asenjo, J. 2013, *Gestión de Base de Datos*.
58. Schmidt, F. 2019, *Ciencia y Ecología*
59. "Biopsia líquida": nuevo test detecta el cáncer de mama en la sangre, [<https://p.dw.com/p/3DnpK>].
60. Sommerville, I. 2007, *Software Engineering*, Pearson Education.
61. Uddin Azeem, A. A. 2019, 'Importance of Software Testing in the Process of Software Development', *International Journal for Scientific Research & Development*, vol. 6, no. 12.
62. Valdez Yáñez, G., Hernández-Pacheco, J. & Arellán-Bravo, L. 2018, 'Glomerulonefritis por biopsia en población atendida en el Hospital Nacional Dos de Mayo'.
63. Valladarez Mellendez , B. M. S., Gailtan, B. M. E. & Pérez Reyes, B. N. N. 2016, *Metodología Ágil de desarrollo de software de programación extrema.*, UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA, MANAGUA.
64. Winn, T. & Calder, P. 2002, 'Is This a Pattern?', *IEEE Software*.

**ANEXOS**

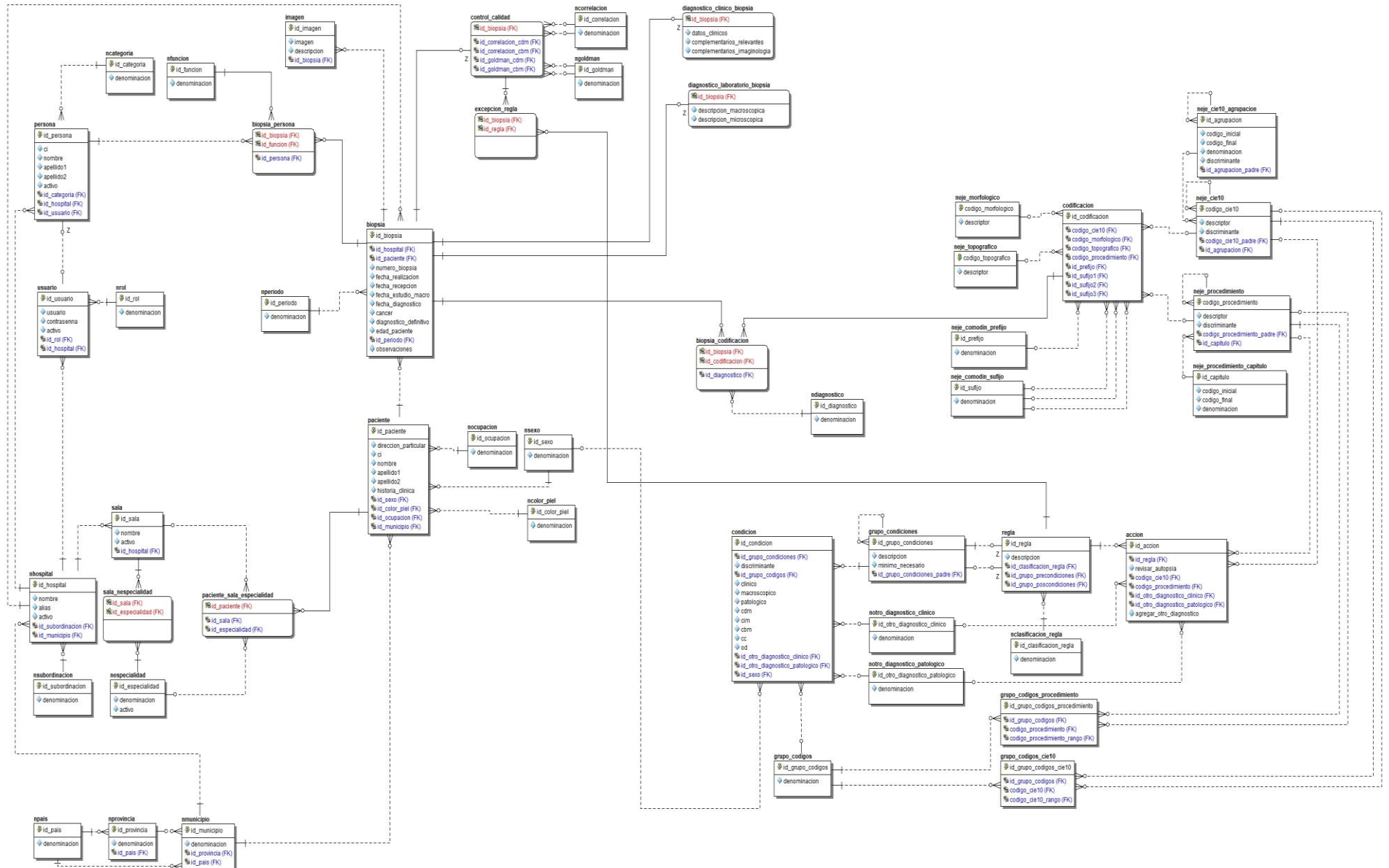
**Anexo 1: Modelo de Solicitud de Biopsia(Citología) 1era parte.**

MODELO 57-01 MINISTERIO SALUD PÚBLICA ANATOMÍA PATOLÓGICA UNIDAD:		<b>MODELO DE SOLICITUD DE BIOPSIA (CITOLOGÍA)</b>			
BIOPSIA:		PROCEDENCIA:			
1 <sup>er</sup> . APELLIDO:		2 <sup>do</sup> . APELLIDO:		NOMBRES:	H. CLÍNICA:
EDAD:	SEXO: M <input type="checkbox"/> F <input type="checkbox"/>	COLOR PIEL: B <input type="checkbox"/> N <input type="checkbox"/> M <input type="checkbox"/> A <input type="checkbox"/>		ESPECIALIDAD:	INDICADA POR:
DIRECCIÓN PARTICULAR:				PROVINCIA:	
FECHA REALIZACIÓN:	FECHA RECEPCIÓN:	FECHA ESTUDIO MACRO:	FECHA DE DIAGNÓSTICO:	SALA/CAMA:	
DATOS CLÍNICOS (INCLUYE ASPECTOS QUIRÚRGICOS Y BIOPSIAS ANTERIORES):          					
COMPLEMENTARIOS RELEVANTES DE LABORATORIOS:          					
COMPLEMENTARIOS DE IMAGINOLOGÍA:          					
IMPRESIÓN DIAGNÓSTICA CLÍNICA:          					
PIEZA ENVIADA A ANATOMÍA PATOLÓGICA (TIPO DE ESTUDIO CITOLÓGICO REALIZADO):          					

**Anexo 2: Modelo de Solicitud de Biopsia(Citología) 2da parte.**

DATOS PARA SER LLENADOS POR ANATOMÍA PATOLÓGICA			
FECHA DE ENTRADA:	FECHA ESTUDIO MACRO:	REALIZADO POR:	
FECHA ESTUDIO MICRO:	REALIZADO POR:	CÁNCER: SI <input type="checkbox"/> NO <input type="checkbox"/>	BIOPSIA POR CONGELACIÓN:
ESTUDIOS ESPECIALES (ESPECIFICAR):			
DIAGNÓSTICO DEFINITIVO:	SI <input type="checkbox"/> NO <input type="checkbox"/>	COINCIDENCIA DIAGNÓSTICA:	SI <input type="checkbox"/> NO <input type="checkbox"/>
DESCRIPCIÓN MACROSCÓPICA:			
DESCRIPCIÓN MICROSCÓPICA:			
DIAGNÓSTICOS ANAMOPATOLÓGICOS:			
OBSERVACIONES:			
<hr style="width: 20%; margin: auto;"/> FIRMA DEL PATÓLOGO			

Anexo 3: Modelo de Base de Dato.



**Anexo 4:** Tarjeta CRC:AdicionarBiopsiaController.java.

TARJETA CRC	
<b>Clase:</b> AdicionarBiopsiaController.java	
<b>Súper Clase:</b> AnchorPane.java	
<b>Responsabilidades:</b>	<b>Colaboraciones:</b>
Adicionar los datos relacionados con la biopsia.	AdicionarBiopsia.java ProtocoloController.java

**Anexo 5:**Tarjeta CRC: AdicionarImpresiónDiagnósticaClínica.java.

TARJETA CRC	
<b>Clase:</b> AdicionarImpresiónDiagnósticaClínica.java	
<b>Súper Clase:</b> AnchorPane.java	
<b>Responsabilidades:</b>	<b>Colaboraciones:</b>
Adicionar los datos relacionados con el diagnóstico clínico.	AdicionarImpresiónDiagnósticaClínica.java ProtocoloController.java

**Anexo 6:** Tarjeta CRC: ExportarReporteDinámico.java

TARJETA CRC	
<b>Clase:</b> ExportarReporteDinámico.java	
<b>Súper Clase:</b> AnchorPane.java	
<b>Responsabilidades:</b>	<b>Colaboraciones:</b>
Visualización de los datos Opción para exportar datos en un	ExportarReporteDinámico.java

documento .pdf .xls o .csv Respuesta del sistema	ProtocoloController.java
---	--------------------------

**Anexo 7:** AdicionarDiagnósticoAnatomopatológico.java.

TARJETA CRC	
<b>Clase:</b> AdicionarDiagnósticoAnatomopatológico.java	
<b>Súper Clase:</b> AnchorPane.java	
<b>Responsabilidades:</b>	<b>Colaboraciones:</b>
Adicionar los datos relacionados con el diagnóstico anatomopatológico.	AdicionarDiagnósticoAnatomopatológico.java ProtocoloController.java