Universidad Nacional Autónoma de México

F A C U L T A D D E I N G E N I E R Í A

Diseño e Implementación de un Sistema Distribuido para el Hospital Regional ISSSTE 1° de Octubre.

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

Ingeniera en Computación

PRESENTA

Arlette Vázquez Estrada

Directora: Dra. María del Pilar Angeles

Ciudad Universitaria, 2010.

ÍNDICE GENERAL

INTRODUCCIÓN

CAPÍTULO 1. ANÁLISIS DEL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA 1

1.2 JUSTIFICACIÓN 3

1.3 OBJETIVOS GENERALES 3

1.4 REQUERIMIENTOS DE LOS USUARIOS 3

1.5 RECOPILACIÓN DE INFORMACIÓN 8

1.5.1 SOLICITUDES 8

1.5.1.1 SOLICITUD DE PIEZAS ANATÓMICAS GENERALES Y ONCOLÓGICAS 8

1.5.1.2 SOLICITUD DE CITOLOGÍA 9

1.5.2 INFORMES 10

1.5.2.1 INFORME DE PIEZAS ANATÓMICAS GENERALES Y ONCOLÓGICAS 10

1.5.2.2INFORME CITOLÓGICO DE DIVERSOS 10

1.5.2.3 INFORME CITOLÓGICO CERVICO VAGINAL 11

1.6 PLANTEAMIENTO DE LA SOLUCIÓN 11

1.7 REQUERIMIENTOS DE SOFTWARE 15

1.7.1 REQUERIMIENTOS PARA EL BACK-END 15

1.7.1.1 POSTGRESQL Y MYSQL 15

1.7.1.1.1 POSTGRESQL 15

1.7.1.1.2 MYSQL 19

1.7.1.1.3 COMPARATIVA ENTRE POSTGRESQL Y MYSQL 22

1.7.1.1.4 SELECCIÓN DEL MANEJADOR DE BASE DE DATOS 24

1.7.2 REQUERIMIENTOS PARA EL *FRONT-END* 25

1.7.2.1 JAVA Y .NET 25

1.7.2.1.1 JAVA 25

1.7.2.1.2 .NET 28

1.7.2.1.3 COMPARATIVA ENTRE JAVA Y .NET 29

1.7.2.1.4 VENTAJAS Y DESVENTAJAS DE CADA PLATAFORMA 31

1.7.2.1.5 SELECCIÓN DE LA PLATAFORMA DE SOFTWARE 31

1.8 REQUERIMIENTOS DE HARDWARE 32

CAPÍTULO 2. MARCO TEÓRICO

2.1 MANEJADOR DE BASE DE DATOS 35
2.2 BASES DE DATOS RELACIONALES 36
2.2.1 PROPIEDADES DE LAS BASES DE DATOS 36
2.2.2 CARACTERÍSTICAS DE UNA BASE DE DATOS 36
2.2.3 NIVELES DE ABSTRACCIÓN EN UNA BASE DE DATOS 37
2.2.4 VENTAJAS DE LAS BASES DE DATOS 38
2.2.5 DESVENTAJAS DE LAS BASES DE DATOS 40
2.3 BASES DE DATOS DISTRIBUIDAS 40
2.3.1 TIPOS DE BASES DE DATOS DISTRIBUIDAS 40
2.3.2 VENTAJAS DE LAS BASES DE DATOS DISTRIBUIDAS 41
2.3.3 DESVENTAJAS DE LAS BASES DE DATOS DISTRIBUIDAS 41
2.3.4 COMPONENTES DE LAS BASES DE DATOS DISTRIBUIDAS 42
2.3.5 ARQUITECTURA DE LAS BASES DE DATOS DISTRIBUIDAS 42
2.3.6 TRANSPARENCIA 44
2.3.7 FRAGMENTACIÓN 44
2.3.7.1 GRADO DE FRAGMENTACIÓN 45
2.3.7.2 TIPOS DE FRAGMENTACIÓN 46
2.3.7.2.1 FRAGMENTACIÓN HORIZONTAL 46
2.3.7.2.1.1 FRAGMENTACIÓN HORIZONTAL PRIMARIA 46
2.3.7.2.1.2 FRAGMENTACIÓN HORIZONTAL DERIVADA 47
2.3.7.2.2 FRAGMENTACIÓN VERTICAL 48
2.3.7.2.3 FRAGMENTACIÓN MIXTA 48
2.3.8 REPLICACIÓN 49
2.3.8.1 TIPOS DE REPLICACIÓN 50
2.3.8.2 VENTAJAS Y DESVENTAJAS DE LA REPLICACIÓN 50
2.3.9 COLOCACIÓN DE LOS DATOS 50
2.4 REDES DE DATOS INALÁMBRICAS 52
2.4.1 VENTAJAS Y DESVENTAJAS 52
2.4.2 COMPONENTES 53
2.4.3 ARQUITECTURA Y PROTOCOLOS DE LAS REDES INALÁMBRICAS 54

2.4.3.1 PROTOCOLOS DE ACCESO AL MEDIO 56
2.4.3.2 ESPACIADO ENTRE TRAMAS IFS 58
2.4.3.3 FORMATO DE LAS TRAMAS MAC 58
2.4.4 TOPOLOGÍAS 59
2.4.5 ESTÁNDARES 61
2.4.6 SEGURIDAD 63
2.4.6.1 PROTOCOLO WEP (Wired Equivalent Protocol) 63
2.4.6.2 PROTOCOLO WAP (PROTOCOLO DE APLICACIONES INALÁMBRICAS) 64
2.5 METODOLOGÍAS DE DISEÑO Y CONSTRUCCIÓN DE SISTEMAS 66
2.5.1 CICLO DE VIDA DE UNA APLICACIONES DE BASES DE DATOS 66
2.5.2 MODELO DE CICLO DE VIDA 69
2.5.2.1 MODELO CASCADA 69
2.5.2.2 MODELO DE DESARROLLO INCREMENTAL 70
2.6 UML 71
2.7 SEGURIDAD SSL 71

CAPÍTULO 3. DISEÑO

3.1 DISEÑO DE LA BASE DE DATOS DISTRIBUIDA 73
3.1.1 DISEÑO CONCEPTUAL 73
3.1.1.1 NORMALIZACIÓN 76
3.1.1.2 DICCIONARIO DE DATOS 80
3.1.1.3 NIVELES DE TRANSPARENCIA 80
3.1.1.4 FRAGMENTACIÓN Y REPLICACIÓN DE DATOS 82
3.1.1.5 DISTRIBUCIÓN DE LOS DATOS 91
3.1.2 DISEÑO LÓGICO Y FÍSICO 92
3.2 DISEÑO DE LA APLICACIÓN 100
3.3 DISEÑO DE LA RED DE DATOS INALÁMBRICA 125

CAPÍTULO 4. IMPLEMENTACIÓN

4.1 IMPLEMENTACIÓN DEL SISTEMA SIDIPAT 131

CAPÍTULO 5. PRUEBAS Y SEGURIDAD

5.1 PRUEBAS 142
5.2 SEGURIDAD 148
5.3 MANTENIMIENTO 151

CONCLUSIONES 153

APÉNDICES

APÉNDICE A GLOSARIO DE TÉRMINOS 155
APÉNDICE B DICCIONARIO DE DATOS 162
APÉNDICE C MANUAL TÉCNICO 172
APÉNCICE D MANUAL DE USUARIO 184

BIBLIOGRAFÍA Y MESOGRAFÍA 218

ÍNDICE DE FIGURAS

CAPÍTULO 1. ANÁLISIS DEL PROBLEMA

1.1 Diagrama del proceso del Departamento de Servicio de Anatomía Patológica 2

1.2 Diagrama de la información recabada..... 4

1.3 Distribución de tipos de solicitudes, informes y médicos especialistas. 6

1.4 Plano del Departamento de Servicio de Anatomía Patológica. 7

1.5 Plano del Departamento de Servicio de Anatomía Patológica con la configuración de la red de datos inalámbrica 14

1.6 Procesos cooperativos de la arquitectura de PostgreSQL 18

1.7 Arquitectura de MySQL..... 21

1.8 Pirámide de manejadores de bases de datos 22

1.9 Diagrama de las características de JAVA..... 26

CAPÍTULO 2. MARCO TEÓRICO

2.1 Funcionamiento de un manejador de base de datos 35

2.2 Niveles de Abstracción..... 38

2.3 Representación del concepto de base de datos distribuida. 40

2.4 Arquitectura de las bases de datos distribuidas..... 43

2.5 Tipos de fragmentación 46

2.6 Ejemplo del árbol de la estructura de la fragmentación mixta..... 49

2.7 Ejemplo de colocación centralizada de los datos. 50

2.8 Ejemplo de colocación particionada de los datos 51

2.9 Ejemplo de colocación replicada de los datos 51

2.10 Ejemplo de una red de datos inalámbrica. 52

2.11 Ejemplo de una unidad base inalámbrica..... 54

2.12 Ejemplos de clientes inalámbricos 54

2.13 Arquitectura MAC..... 55

2.14 Funcionamiento del protocolo CSMA/CA..... 57

2.15 Ejemplo de nodos ocultos..... 57

2.16 Espaciado entre tramas. 58

2.17 Formato de una trama 59
2.18 Ejemplo de una red ad-hoc 59
2.19 Ejemplo de una red de modo infraestructura 60
2.20 Funcionamiento del protocolo WEP. 64
2.21 Modelo de ciclo de vida en cascada 70
2.22 Modelo de desarrollo incremental 70
2.23 Modelo de la seguridad SSL 72

CAPÍTULO 3. DISEÑO

3.1 Diagrama Entidad – Relación de la base de datos del Departamento de Servicio de Anatomía Patológica 75
3.2 Diagrama de la entidad SOLICITUD 76
3.3 Ejemplo del incumplimiento de la 1 FN en la entidad SOLICITUD. 77
3.4 Solución a la 1FN de la entidad SOLICITUD. 77
3.5 Ejemplo donde no es válida la 2 FN y 3 FN en la entidad SOLICITUD 78
3.6 Solución a la 2 FN y 3 FN de la entidad SOLICITUD. 78
3.7 Ejemplo de la entidad SOL_CITOLOGICA donde no se aplica la normalización. 79
3.8 División de los datos de la entidad SOL_ANATOMOPATOLOGICAS en cinco nodos 80
3.9 Diagrama del protocolo commit de dos fases 82
3.10 Representación de la fragmentación horizontal primaria de la entidad SOL_ANATOMOPATOLOGICAS 83
3.11 Representación de la fragmentación derivada de las solicitudes anatomopatológicas. 84
3.12 Fragmentación vertical de las solicitudes citológicas de diversos. 84
3.13 Ejemplo de la fragmentación vertical en las solicitudes citológicas de diversos. 84
3.14 Características de las solicitudes citológicas cervico vaginales 85
3.15 Ejemplo de la fragmentación vertical en las solicitudes citológicas cervico vaginales. 85
3.16 Fragmentación horizontal primaria (A) y fragmentación derivada (B) de la solicitud citológica de diversos 85
3.17 Fragmentación horizontal primaria (A) y fragmentación derivada (B) de la solicitud citológica cervico vaginal 86
3.18 Fragmentación vertical de los informes citológicos de diversos 87
3.19 Ejemplo de la fragmentación vertical en los informes citológicos de diversos 87

3.20 Fragmentación vertical de los informes citológicos cervico vaginales 87
3.21 Ejemplo de la fragmentación vertical en los informes citológicos cervico vaginales..... 88
3.22 Fragmentación primaria (A) y Fragmentación derivada (B) de los informes citológicos de diversos. 88
3.23 Fragmentación primaria (A) y Fragmentación derivada (B) de los informes citológicos cervico vaginales..... 88
3.24 Diagrama de la distribución de los datos fragmentados y replicados..... 91
3.25 Selección de la pestaña “Tablespaces”. 94
3.26 Creación de un nuevo tablespace 94
3.27 Configuración del tablespace..... 94
3.28 Visualización de tablespaces existentes 95
3.29 Selección de la pestaña “Databases”..... 95
3.30 Creación de una nueva base de datos. 95
3.31 Configuración de la base de datos..... 96
3.32 Visualización de las bases de datos existentes 96
3.33 Selección de la opción SQL..... 96
3.34 Ventana especial para correr los scripts..... 97
3.35 Selección de la opción para realizar el respaldo de la base de datos “Patologia” 99
3.36 Configuración del respaldo de la base de datos “Patologia” 99
3.37 Creación del repaldo de la base de datos “Patologia”..... 99
3.38 Pantalla principal para la conexión con el manejador de la base de datos 100
3.39 Opciones de funcionalidad del sistema *SIDIPAT*. 102
3.40 Diagrama jerárquico funcional de la pantalla SOLICITUDES..... 102
3.41 Pantalla de SOLICITUDES..... 103
3.42 Diagrama de flujo de datos de la función REGISTRAR 103
3.43 Diagrama de flujo de datos de la función ACLARACIÓN 104
3.44 Pantalla de la función BÚSQUEDA..... 104
3.45 Diagrama de flujo de datos de la función BÚSQUEDA..... 105
3.46 Diagrama de flujo de datos de la función ASIGNACIÓN 106
3.47 Diagrama de flujo de datos de la función REPORTES 107
3.48 Hoja para el reporte diario de solicitudes. 108
3.49 Hoja para el reporte de aclaraciones de las solicitudes. 108

3.50 Procedimiento para realizar el filtro de solicitudes..... 109
3.51 Pantalla que sirva para realizar el filtro de las solicitudes para informes anatomopatológicos..... 109
3.52 Pantalla que sirva para realizar el filtro de las solicitudes para informes anatomopatológicos. 110
3.53 Pantalla INFORMES de tipo anatomopatológicos 110
3.54 Pantalla INFORMES de tipo citológicos..... 111
3.55 Diagrama jerárquico funcional de la pantalla INFORMES..... 111
3.56 Diagrama de flujo de datos de la función GENERAR 112
3.57 Diagrama de flujo de datos de la función ACTUALIZAR. 113
3.58 Diagrama de flujo de datos de la función ENVIAR..... 113
3.59 Diagrama de flujo de datos de la función REPORTES 114
3.60 Formato de la hoja de reportes de informes 114
3.61 Pantalla NOTAS..... 115
3.62 Menú Abrir de la pantalla NOTAS..... 115
3.63 Menú Guardar de la pantalla NOTAS..... 115
3.64 Pantalla RESULTADOS..... 116
3.65 Diagrama de flujo de datos de la función BUSCAR..... 116
3.66 Ejemplo de un informe de resultado de tipo anatomopatologicos (generales y oncológicos). 117
3.67 Ejemplo de un informe de resultado de tipo citológico de diversos..... 118
3.68 Ejemplo de un informe de resultado de tipo citológico cervico vaginal..... 119
3.69 Pantalla REPORTES 120
3.70 Diagrama de flujo de datos de la función BUSCAR..... 120
3.71 Ejemplo de un reporte por patológo..... 121
3.72 Ejemplo de un reporte por paciente 121
3.73 Ejemplo de un reporte por citotecnólogo..... 121
3.74 Procedimiento para cambiar el estado de un patólogo 122
3.75 Procedimiento para cambiar el estado de un informe..... 122
3.76 Calendario dinámico de la librería JCalendar..... 123
3.77 Diagrama de conexión de la red de datos inalámbrica 125
3.78 Página del Explorador de Internet con la dirección del AP..... 126

3.79 Pantalla del AP para ingresar el usuario y el password 126
3.80 Elección del modo Access point 126
3.81 Elección del canal 6..... 127
3.82 Ejemplo de un identificador de servicio 127
3.83 Tipos de autenticación..... 127
3.84 Configuración de la seguridad WEP en el AP 127
3.85 Ejemplo del icono de la red inalámbrica..... 128
3.86 Submenú del icono de red inalámbrica 128
3.87 Configuración de la seguridad WEP en cada nodo de la red inalámbrica 129
3.88 Ejecución en consola del comando ping..... 130
3.89 Pantalla para cambiar o unirse a un grupo 130

CAPÍTULO 4. IMPLEMENTACIÓN

4.1 Icono de la versión de JAVA 131
4.2 Imagen del directorio que contiene las librerías..... 131
4.3 Procedimiento para la creación de usuarios..... 132
4.4 Pantalla que muestra la ejecución del comando Slon. 134
4.5 Pantalla que muestra la elección de la base de datos a replicar..... 135
4.6 Pantalla que muestra el mensaje que aparece cuando se especifica la ruta de los archivos de Slony –I 135
4.7 Pantalla que muestra la ejecución del comando slonik..... 137
4.8 Ubicación del archivo pg_hba.conf 138
4.9 Configuración de la replicación 139
4.10 Pantalla de la ejecución del comando slonik 139
4.11 Pantalla de la ejecución del comando slon, inicio de la replicación..... 140
4.12 Servicio de Windows Slony –I..... 141

CAPÍTULO 5. PRUEBAS Y SEGURIDAD

5.1 Pantalla de error de validación de password..... 142
5.2 Pantalla de error de validación de usuario y password..... 143
5.3 Pantalla de error de validación de campos..... 143

5.4 Ejemplo de mejora en la letra de los campos 144
5.5 Pantalla que contiene el campo del médico asignado 144
5.6 Pantalla que contiene la funcionalidad para cambiar el estado de los patólogos e informes 145
5.7 Pantalla que justifica las modificaciones de la base de datos 145
5.8 Gráfica que muestra el porcentaje en el que ocurren los errores 146
5.9 Prueba del funcionamiento del programa DB Stress 147
5.10 Gráfica del número de iteraciones por proceso 148
5.11 Procedimiento para la creación de la llave primario y del certificado SSL 150
5.12 Gráfica que muestra los porcentajes de costo entre las diferentes etapas del desarrollo de una aplicación 151

ÍNDICE DE TABLAS

CAPÍTULO 1. ANÁLISIS DEL PROBLEMA

1.1 Funciones que realiza cada unos de los procesos 4
1.2 Información de las solicitudes de piezas anatómicas 8
1.3 Información de solicitudes citológicas. 9
1.4 Información de la solicitud citológica de diversos 9
1.5 Información de la solicitud citológica cervico vaginal 10
1.6 Limitaciones que presenta el manejador de base de datos PostgreSQL 17
1.7 Puntaje de evaluación 22
1.8 Comparativa entre los manejadores PostgreSQL y MySQL 24
1.9 Puntaje final de la comparación de manejadores 24
1.10 Comparativa realizada entre JAVA y .NET. 29
1.11 Ventajas y desventajas de JAVA y .NET. 31
1.12 Requerimientos de hardware para las aplicaciones elegidas. 32

CAPÍTULO 2. MARCO TEÓRICO

2.1 Ejemplos de conexiones inalámbricas que utilizan ondas electromagnéticas 61
2.2 Descripción de los estándares físicos 62
2.3 Características principales de los estándares físicos. 63

CAPÍTULO 3. DISEÑO

3.1 Formato que tiene el diccionario de datos para mostrar la información 80
3.2 Consultas que ejemplifican la transparencia de distribución 81
3.3 Ejemplo del fragmento de las solicitudes anatomopatológicas generales. 83
3.4 Ejemplo del fragmento de las solicitudes anatomopatológicas oncológicas. 83
3.5 Entidades que serán replicadas al nodo RECEPCIÓN 86
3.6 Entidades que serán replicadas al nodo CITOTECNÓLOGO 89
3.7 Entidades que serán replicadas al nodo PATÓLOGO 3 89
3.8 Esquema de fragmentación y ubicación del sistema 90
3.9 Tamaño aproximado de la base de datos en 5 años 92

3.10 Expresiones regulares para la validación de los datos ingresados en los campos. 123

CAPÍTULO 4. IMPLEMENTACIÓN

4.1 Lista de usuario para el sistema *SIDIPAT* 133

4.2 Ejemplo de direcciones IP que están contenidas en el archivo pg_hba.conf 138

CAPÍTULO 5. PRUEBAS Y SEGURIDAD

5.1 Número de iteraciones por proceso 147

5.2 Resultados de la prueba de Stress realizada 148

5.3 Explicación de los parámetros del archivo server.req. 150

APÉNDICE B DICCIONARIO DE DATOS

B.1 Características del catálogo CANCER 162

B.2 Características del catálogo CARACTERISTICAS_MUESTRA 162

B.3 Características de la entidad CITOTECNOLOGOS. 162

B.4 Características del catálogo DIAGNOSTICOS_CITOLOGICOS. 163

B.5 Características del catálogo EXPLORACIONES 163

B.6 Características de la entidad HALLAZGOS 163

B.7 Características de la entidad GINECOBSTETRICOS 164

B.8 Características de la entidad INF_ANATOMOPATOLOGICOS 164

B.9 Características de la entidad INF_CITOLOGICOS 165

B.10 Características de la entidad INFORMES. 165

B.11 Características del catálogo METODOS_OBTENCION 166

B.12 Características del catálogo MUESTRAS_PROCE. 166

B.13 Características de la entidad PACIENTES 166

B.14 Características de la entidad PATOLOGOS 167

B.15 Características del catálogo PRODUCTOS. 167

B.16 Características de la entidad REGISTRO. 167

B.17 Características del catálogo SERVICIOS 168

B.18 Características del catálogo SINTOMAS 168

B.19 Características de la entidad SOL_ANATOMOPATOLOGICAS 168

B.20 Características de la entidad SOL_CITOLOGICAS..... 169
B.21 Características de la entidad SOLICITUDES 169
B.22 Características del catálogo TIPO_ANATOMOPATOLOGICAS. 170
B.23 Características del catálogo TIPO_CITOLOGICAS. 170
B.24 Características del catálogo TIPO_SOL_INF. 170
B.25 Características de la entidad UNIDADES_MEDICAS..... 171
B.26 Características del catálogo UTENSILIOS. 171

APÉNDICE D MANUAL DE USUARIO

D.1 Simbología que utilizan los diagramas de flujo que componen el manual de usuario.. 185
D.2 Formatos de los campos de la pantalla solicitud 191
D.3 Combinaciones posibles para buscar una solicitud. 196
D.4 Combinaciones para la búsqueda de informes de resultados. 211
D.5 Combinaciones posibles para generar reportes..... 214

INTRODUCCIÓN

La información que se maneja en el Departamento de Servicio de Anatomía Patológica es muy importante y confidencial, ya que los informes de resultados de las biopsias o muestras ayudan a diagnosticar y ofrecer un tratamiento a los padecimientos que pueden presentar los pacientes.

Al conocer los procedimientos que se realizan en dicho departamento se detecta que no se está brindando un servicio de calidad al paciente por el tiempo que se lleva el personal en generar los informes de resultados. Por lo tanto, se propone implementar un sistema distribuido que permita mejorar el proceso de registro de solicitudes de biopsias y muestras.

El sistema distribuido se llamará **SIDIPAT** que significa Sistema distribuido para el Departamento del Servicio de Patología, será diseñado con software libre para evitar cualquier problema de licencias. El propósito del sistema distribuido es utilizar las ventajas y las características que ofrecen las bases de datos distribuidas para mejor el ambiente en el que se encuentra actualmente el Departamento de Servicio de Anatomía Patológica.

Al concluir dicho sistema se intenta mejorar la calidad del servicio que ofrece el Departamento como consecuencia de las mejoras proporcionadas al procedimiento interno de registro de solicitudes y generación de informes de resultados que se realiza en dicho Departamento de Servicio.

La documentación del proyecto final ***Diseño e implementación de un sistema distribuido para el Hospital Regional ISSSTE 1° de octubre*** ésta compuesta de cinco capítulos que se describen a continuación:

El capítulo 1 denominado ***Análisis del problema*** trata sobre la problemática que se presenta en el Departamento de Servicio de Anatomía Patológica, en base a ello se conocen los requerimientos del usuario y se recolecta la información requerida. Con lo anterior se propone realizar un sistema distribuido donde se justifica cada una de las características que contemplará dicho sistema, así como las aplicaciones que se utilizaran para poderlo diseñar e implementar.

En el capítulo 2 ***Marco teórico*** se encuentra una investigación sobre los temas relacionados a los sistemas distribuidos como son: bases de datos centralizadas, redes de datos, bases de datos distribuidas, seguridad, ciclo de vida del software, etc.

En el capítulo 3 ***Diseño*** se describe de manera práctica todas las etapas que se requieren para diseñar una base de datos distribuida, así como las funciones que tendrá la interface del usuario para interactuar con la base de datos a través de una red de datos.

El capítulo 4 ***Implementación*** contempla los procesos requeridos para poner en funcionamiento el sistema distribuido en el Departamento de Servicio de Anatomía Patológica.

En el capítulo 5 ***Prueba y Seguridad*** se explican las diferentes pruebas que se le realizan al sistema distribuido para verificar que esté funcione adecuadamente y cubra las necesidades que se tienen en dicho Departamento de Servicio, en caso contrario se realicen modificaciones. También se describe las características y configuraciones de seguridad con que cuenta el sistema distribuido. Por último se menciona el mantenimiento que se aplicará al sistema después de cierto tiempo de funcionamiento.

La documentación también contempla ciertos apéndices que contienen el glosario de términos, el diccionario de datos de la base de datos, el manual técnico y el manual de usuario.

AGRADECIMIENTOS

Quiero dedicar este trabajo a las personas que en cada etapa de mi vida me han acompañado y han dejado huella en ella.

A mi ***madre Angela Estrada Martínez*** quien con su amistad, consejos, dedicación, amor y valores enseñados ha colaborado en el crecimiento de la persona que soy. Me ha acompañado en todo momento principalmente en esta etapa en la que finalizado un esfuerzo mutuo. Gracias por estar a mi lado para darme ánimos de seguir adelante y luchar por mis sueños. Te quiero mucho mami.

A mi ***padre Gilberto Vázquez Sánchez*** quien confió y me dio todo su apoyo para realizar este trabajo. Gracias a tus enseñanzas y momentos compartidos durante este tiempo de trabajo y en cada etapa de mi vida he logrado culminar mis estudios profesionales. Gracias por el amor, las alegrías, las tristezas, por tu esfuerzo de darme lo mejor día con día he llegado a ser una persona completa en todos los aspectos. Te quiero papi.

A mi ***hermana Iliana Vázquez Estrada*** que a pesar de la diferencia de carácter siempre se encuentra a mi lado para apoyarme en cualquier situación. Gracias por las alegrías, las pláticas, las risas y las tristezas que hemos compartido juntas, siempre las llevaré en mi corazón.

A mi ***abuelita Galdina García*** que con su ternura, su cariño y su forma de ser deja en mí las fuerzas de luchar y seguir adelante por mis sueños y por mis seres queridos.

A mi ***novio Alberto Segura*** quien llego en el mejor momento de mi vida. Quien con su cariño, amor, paciencia y confianza me ha apoyado en esta etapa tan importante para mí. Quien me alentaba a seguir cuando me sentía derrotada. Gracias por estar siempre a mi lado sin importar las circunstancias o mi estado de ánimo. Te quiero mucho.

Al ***padre Alberto Rubio Palacio*** quien con sus enseñanzas han dejado en mi el deseo de llevar una vida por el camino de la verdad, de la rectitud, de la humildad y la caridad. Gracias por ser parte de mi familia. Lo estimo mucho.

A mis ***amigos*** que a lo largo de mi vida he conocido y han dejado una huella en mi vida. Gracias por todas las travesuras realizadas, las risas que no podían faltar nunca, los momentos de estudio y cada uno de los instantes que viví con ustedes.

A mi directora de tesis la ***Dra. Ma. Del Pilar*** quien sin conocerme creyó en este proyecto y en mí. Gracias por todas las enseñanzas, por su tiempo dedicado a cada uno de los capítulos que contemplan este trabajo, por su interés y responsabilidad. Siempre estaré agradecida por darme este voto de confianza.

A la ***Universidad Nacional Autónoma de México y a la Facultad de Ingeniería*** que me dieron todos los conocimientos para desarrollarme profesionalmente y la oportunidad de ser parte de la comunidad universitaria. En cada una de las instalaciones dejo recuerdos que siempre quedaran grabados en mi corazón.

CAPÍTULO 1

ANÁLISIS DEL PROBLEMA

No podemos resolver problemas usando
el mismo tipo de pensamiento que
usamos cuando los creamos.
Albert Einstein

1.1 PLANTEAMIENTO DEL PROBLEMA

El Hospital Regional "1° de Octubre" del ISSSTE cuenta con diferentes Departamentos de Servicios los cuales proporcionan atención médica a los pacientes que acuden a ésta Unidad Hospitalaria. Uno de los Departamentos de Servicio es el de Anatomía Patológica que tiene a su cargo el análisis y la administración de los estudios de biopsias como son piezas anatómicas y citologías.

Con base en el modo de trabajo que desempeña dicho Departamento de Servicio, se sigue un procedimiento para el trámite y registro de solicitudes de los estudios de biopsias, así como para la entrega de los informes de resultados de los estudios por parte de los Médicos especialistas de dicho Departamento de Servicio. Dicho procedimiento es el siguiente (Véase la figura 1.1):

Los médicos de ésta Unidad Hospitalaria tratan cotidianamente a diferentes pacientes que presentan síntomas y padecimientos diversos, de acuerdo a esto se les da un tratamiento o se solicita la realización de algunos estudios. Entre los estudios que se pueden efectuar está por ejemplo la toma de una biopsia que permite obtener un diagnóstico certero de la enfermedad que presenta el paciente. Para poder conseguir el resultado del análisis de la biopsia se genera una solicitud que contiene los datos del paciente, de la biopsia y del médico que está solicitando dicho estudio. Posteriormente se envía la solicitud junto con la biopsia al Departamento de Servicio de Anatomía Patológica.

Cuando llega la solicitud y la biopsia a dicho Departamento de Servicio, el personal administrativo de éste, lleva a cabo el registro poniendo los datos que vienen en la solicitud en algunas libretas de control administrativo. Se establece un número de registro dependiendo del tipo de solicitud que llegue, las cuales pueden ser cervico vaginales, de diversos que son líquidos y células de descamación, generales que son pequeñas muestras de diferentes órganos, y oncológicas. Posteriormente se asignan las biopsias registradas al médico especialista de dicho Departamento, que sea el adecuado para el análisis de ésta, ya que el Departamento de Servicio de Anatomía Patológica cuenta con patólogos, oncólogos y citotecnólogos.

El médico especialista obtiene la biopsia y determina las técnicas necesarias para efectuar el análisis completo. En el proceso de dicho estudio, él tiene que ir elaborando un informe de resultados que contendrá el diagnóstico final, así como observaciones que él pueda brindar al médico de la Unidad Hospitalaria que solicitó este estudio. Los informes elaborados por el médico especialista pueden ser de tipo cervico vaginal, de diversos, generales, y oncológicos, dependiendo del tipo de solicitud a la que este referida la biopsia. Al culminar dicho informe lo entregará en el tiempo establecido al personal administrativo del Departamento de Servicio de Anatomía Patológica.

Al contar el personal administrativo con los informes de resultados de los diferentes médicos especialistas, los enviará al Departamento de Archivo de la Unidad Hospitalaria, para que los ingrese al expediente del paciente. Entregados estos informes el proceso administrativo y de análisis de los estudios de biopsias culmina por parte del Departamento de Servicio de Anatomía Patológica. El paciente y el médico de ésta Unidad Hospitalaria son responsables de recoger el informe emitido por dicho Departamento.

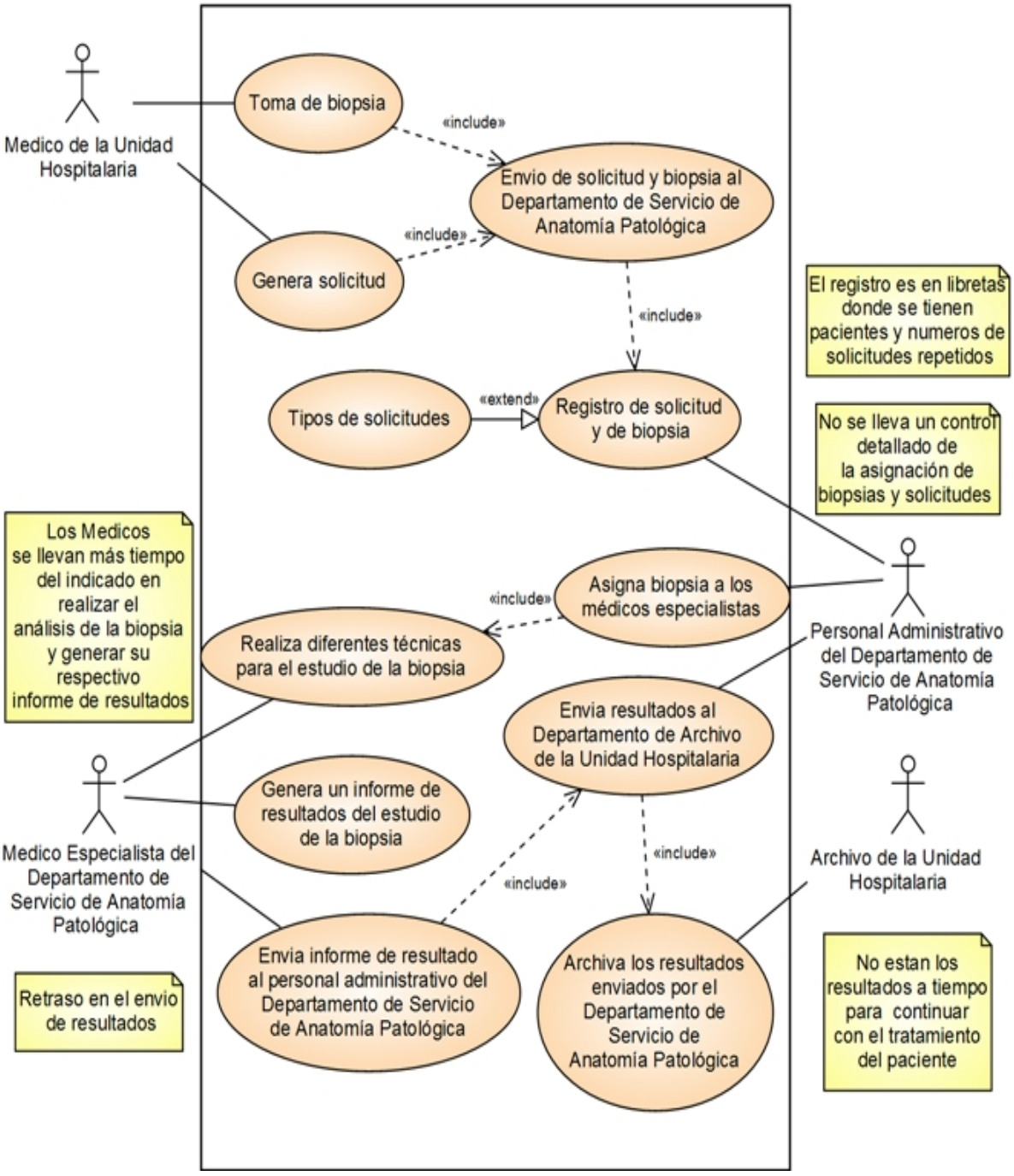


FIGURA 1.1 Diagrama del proceso del Departamento de Servicio de Anatomía Patológica.

Al analizar todos los puntos anteriores se llega a plantear la idea central del problema, el servicio que brinda el Departamento de Servicio de Anatomía Patológica del Hospital Regional 1° de Octubre del ISSSTE, consiste en realizar estudios morfológicos de biopsias anatómicas y citológicas. Este servicio desde hace algún tiempo tiene deficiencias tanto en el proceso de trámite y registro de solicitudes de estudio, así como en las entregas puntuales de los resultados de los estudios por parte de los Médicos de dicho servicio, dado que actualmente es poco eficiente el seguimiento total de las actividades que realiza cada médico para entregar de manera satisfactoria su trabajo. Esto genera un servicio deficiente al paciente, por no tener los resultados de sus estudios en los tiempos estipulados por el Manual de Procedimientos de esta área, retrasando en ocasiones el tratamiento específico y/o complementario en algunos pacientes.

1.2 JUSTIFICACIÓN

Debido a las deficiencias en la administración de los procesos en el Departamento de Servicio de Anatomía Patológica del Hospital Regional 1° de Octubre del ISSSTE, surge la necesidad de diseñar e implementar una Sistema de base de datos distribuida, que permita llevar a cabo el proceso administrativo en todas sus etapas, con la finalidad de cumplir con los lineamientos estipulados en los Manuales de Procedimientos, para optimizar los recursos humanos y materiales de este Departamento de Servicio y de esta manera proporcionar una atención de mayor calidad a los pacientes.

1.3 OBJETIVOS GENERALES

Diseñar e implementar un Sistema distribuido en el Departamento de Servicio de Anatomía Patológica del Hospital Regional 1° de Octubre del ISSSTE, que optimice los procesos de administración de los estudios de biopsias como son las piezas anatómicas y citologías cervico vaginales y de diversos, con la finalidad de brindar un servicio de calidad a los derechohabientes de esta Unidad Hospitalaria. Por lo tanto se desarrollará un modelo piloto, que en un futuro podría ser incorporado a los demás Departamentos de Servicios Médicos y Administrativos de este Hospital.

1.4 REQUERIMIENTOS DE LOS USUARIOS

Para resolver las dificultades que se presentan en el Departamento de Servicio, se realizaron entrevistas al personal con el fin de conocer sus expectativas y las ineficiencias a las cuales ellos se enfrentan. Se realizó una entrevista con el jefe del Departamento del Servicio, quien conoce más a fondo el proceso completo que se tiene que realizar, al igual que las funciones que cada empleado tiene que realizar. El jefe del Departamento de Servicio nos indicó ciertas características que se deben contemplar en el sistema a fin de incluirlas en la propuesta de solución.

En la figura 1.2 se muestra la representación de la información recabada de las entrevistas.

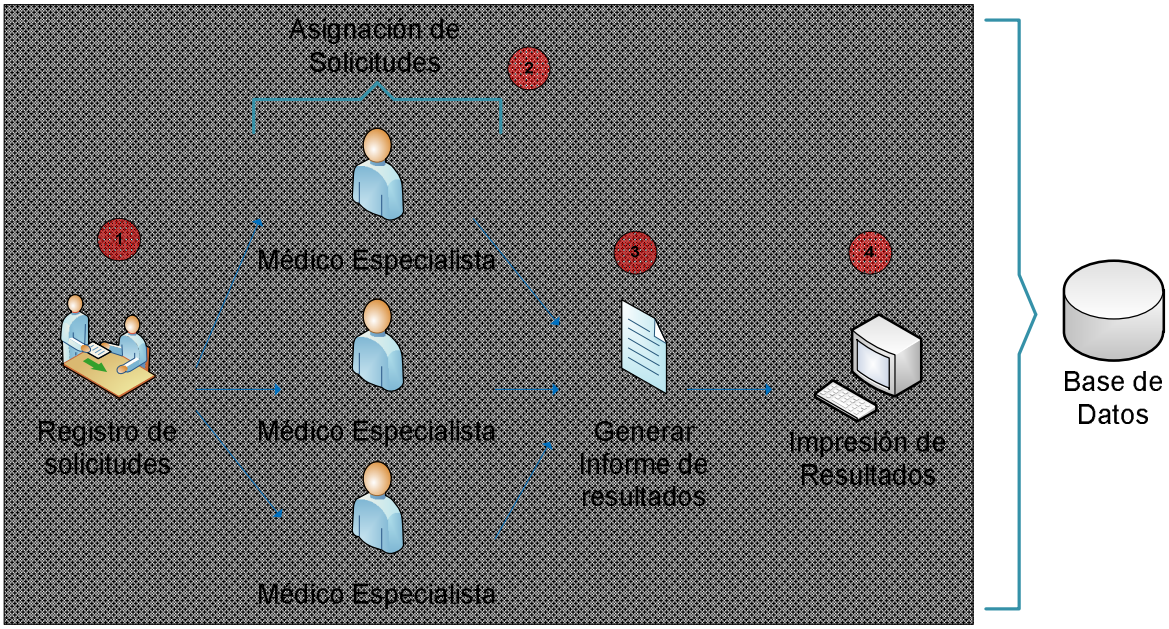


FIGURA 1.2 Diagrama de la información recabada.

Para entender de manera detallada cada uno de los procesos vistos en la figura 1.1 se hace una descripción de las funciones que ejecuta cada uno de ellos. Esta información se visualiza en la Tabla 1.1

Proceso	Descripción
1	Se realizará el registro de las solicitudes y biopsias que llegan de otros Departamentos de Servicio. Los datos ingresados en este punto serán utilizados posteriormente para realizar los informes de resultados, por lo cual esta información tiene que quedar disponible para los médicos especialistas. Se realizará la asignación de solicitudes a los médicos especialistas de acuerdo al tipo que está presente. Esta asignación se tiene que realizar de manera dinámica por el sistema, asignándoles así un cierto número de estudios que sea equitativo para todos los médicos.
2	En este punto cada médico especialista ya recibió las solicitudes que le fueron asignadas. El médico especialista empieza el estudio y el análisis de la biopsia de acuerdo a sus técnicas de trabajo. Podrá ir realizando el informe de resultados, en donde pondrá las características encontradas en la muestra y el diagnóstico final.

2	El jefe del Departamento de Servicio nos pide en este punto tener en cuenta el espacio que se requiere para que el médico pueda plasmar su diagnóstico, y no limitarlo a una cierta cantidad de caracteres. Se pueda obtener un reporte de avances de actividades que tenga cada médico especialista para poder verificar el cumplimiento adecuado de su trabajo y en los tiempos estipulados por el jefe del Departamento de Servicio, con ello poder averiguar la causa de la ineficiencia que se está teniendo en esta parte del proceso que se realiza.
3	Cuando los médicos especialistas hayan terminado adecuadamente el estudio de las biopsias y estén seguros que no requiere ningún cambio en el informe de resultados, podrán generar automáticamente la hoja entregable del resultado, la cual será mandada al personal administrativo del Departamento de Servicio. En este punto se culmina la participación del médico especialista, ya que su trabajo se basa en generar el informe de resultados.
4	Finalmente el personal administrativo recibe el informe de resultados para imprimirlo y llevarlo al Departamento de Archivo. Otro punto importante es llevar el control de los informes de resultados entregados.

TABLA 1.1 Funciones que realiza cada unos de los procesos.

De acuerdo a los procesos descritos anteriormente, el jefe de Departamento de Servicio de Anatomía Patológica quiere que se optimice el proceso de administración de solicitudes e informes de biopsias.

Después de describir el funcionamiento que desea el jefe de Departamento de Servicio, también nos señaló algunas cuestiones que se deben tener en cuenta:

- La característica más importante que se señala, corresponde a que no todos los usuarios o personal del Departamento de Servicio tengan acceso a toda la información que se maneja, sino únicamente a una porción de información que los ayude a realizar su trabajo.
- Se tiene que ocupar software que no requiera licencias, ya que la Unidad Hospitalaria no comprará ninguna licencia que se necesite.
- Si se planea utilizar una red de datos, tiene que ser una extensión de la red interna de la Unidad Hospitalaria por cuestiones de Políticas de la propia Unidad.
- Si se desea instalar una red de datos, no se puede realizar ninguna modificación en la construcción.
- Si la red de datos es alámbrica no se asegura que el material necesario para ésta será dado por la Unidad Hospitalaria. Se recomienda por tanto configurar una red de datos inalámbrica.

- Se tiene que acoplar la solución al equipo de cómputo con que se cuenta en el Departamento de Servicio.
- Desarrollar un sistema que sea fácil de usar, pero que cumpla con todos los requerimientos a fin de mejorar el servicio que se brinda al paciente.
- Se requiere obtener informes semanales, mensuales y anuales de las actividades generales del Departamento de Servicio, así como de cada médico especialista para poder analizar el avance que se ha tenido en la entrega de resultados en el tiempo indicado.
- Préstamo de libretas de años anteriores para obtener la información necesaria para realizar pruebas del sistema.
- Se tenga la posibilidad de indicar qué patólogos se encuentran de vacaciones.
- Tener en cuenta que se manejan diferentes tipos de solicitudes e informes, así como se tienen médicos de diferentes especialidades. Véase la figura 1.3 para entender la distribución de solicitudes, informes y médicos especialistas.

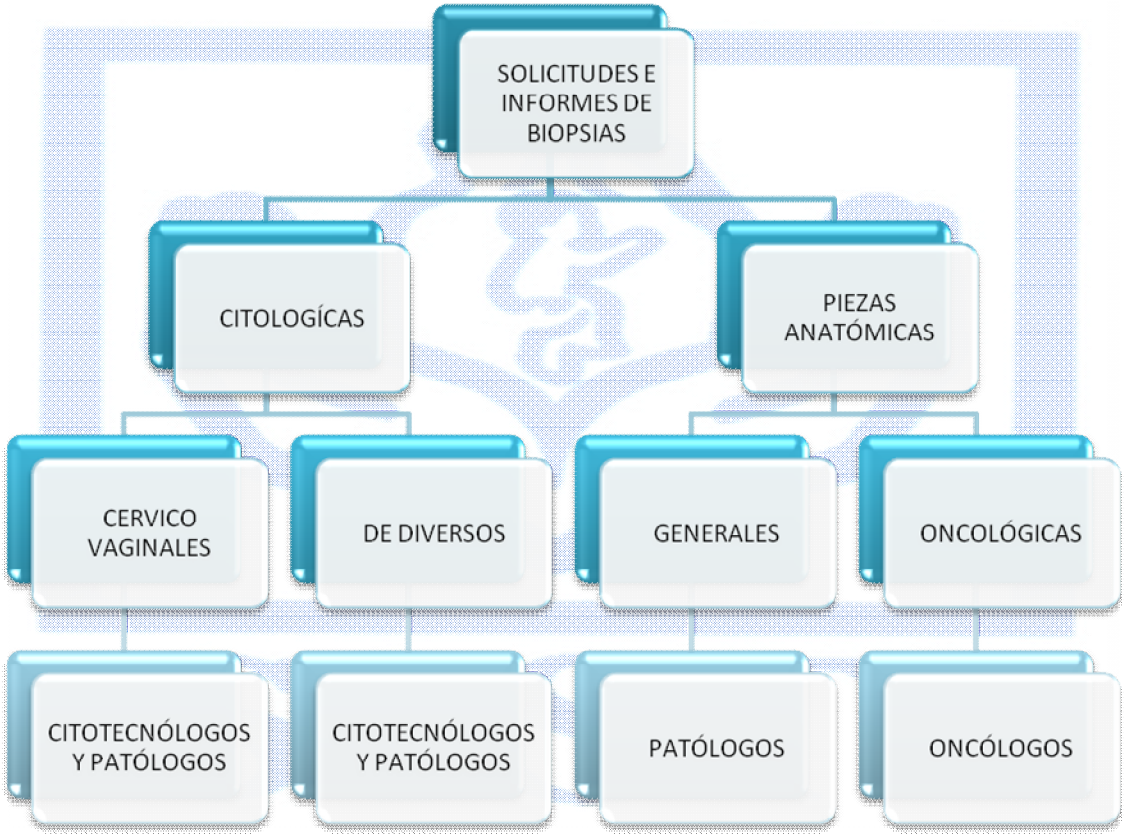


FIGURA 1.3 Distribución de tipos de solicitudes, informes y médicos especialistas.

Se solicita nos proporcionado un plano del Departamento de Servicio, para poder observar las limitaciones de espacio, las características de una red inalámbrica o alámbrica, saber en qué cubículos se puede contemplar la existencia de algún nodo, etc. , también nos indican el número de máquinas que hay en el Departamento de Servicio. Lo anterior se muestra en la figura 1.4.

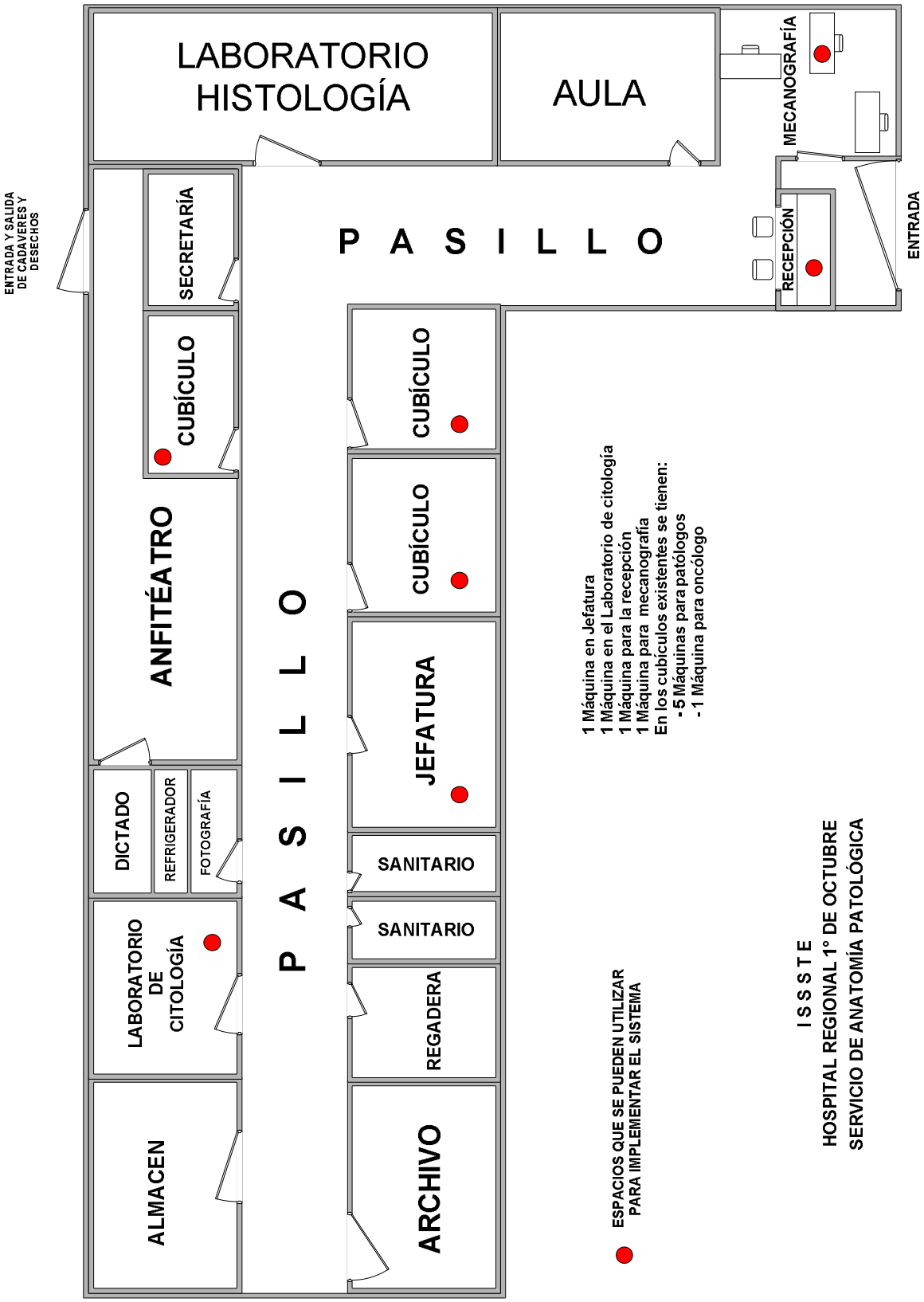


FIGURA 1.4 Plano del Departamento de Servicio de Anatomía Patológica.

1.5 RECOPILACIÓN DE INFORMACIÓN

Para poder realizar los formularios de las solicitudes y los informes, el jefe de Departamento del Servicio de Anatomía Patológica, nos proporciona la información necesaria para el registro de las solicitudes y la generación de los informes de resultados. Dicha información se divide de la siguiente manera:

1.5.1 SOLICITUDES

Como ya se había mencionado existen diferentes tipos de solicitudes y cada una de ellas contiene información distinta, por lo tanto se describe a continuación de manera detallada el contenido de cada solicitud:

1.5.1.1 SOLICITUD DE PIEZAS ANATÓMICAS GENERALES Y ONCOLÓGICAS

En la tabla 1.2 se muestra la información que deben contemplar las solicitudes de piezas anatómicas.

INFORMACIÓN DE LAS SOLICITUDES DE PIEZAS ANATÓMICAS GENERALES Y ONCOLÓGICAS
Nombre Completo del paciente **Número de expediente del paciente** **Sexo del paciente** **Edad del paciente** **Fecha de solicitud del estudio** **Nombre completo del médico tratante de la Unidad Hospitalaria (medico que está pidiendo que se realice dicho estudio)** **Departamento de Servicio que genera la solicitud** **Pieza remitida (descripción de biopsia que fue tomada por el médico)** **Diagnóstico Clínico (previo diagnóstico que se tiene del padecimiento que presenta el paciente)** **En cuanto a los datos relacionados con la sala y cama en la que se encuentra el paciente hospitalizado, el personal de dicho Departamento, no lo están requiriendo.** **Número de Registro, el cual tiene el siguiente formato dependiendo de qué pieza anatómica se está mandando a estudiar** ***Q – XXXXX – AÑO ó Q – O – XXXXX – AÑO*** **Donde:** **Q: Pieza quirúrgica** **O: En caso de que la pieza quirúrgica sea oncológica se indicara con esta letra** **XXXXX: número de registro de la solicitud e informe 00001 – 10000** **AÑO: Año en curso se representa solo con los últimos dos dígitos.** **Por ejemplo 2010 ⟶ 10**

TABLA 1.2 Información de las solicitudes de piezas anatómicas.

1.5.1.2 SOLICITUD DE CITOLOGÍA

En cuanto a las solicitudes de citología no importa el tipo que sea, si es cervico vaginal o de diversos, en el caso del número de registro de dichas solicitudes es el mismo. Dicha información se muestra en las tablas 1.3.

INFORMACIÓN DE LAS SOLICITUDES CITOLÓGICAS
Número de Citología , el cual tiene el siguiente formato ***C – XXXXX – AÑO*** **Donde:** **C: Citología** **XXXXX: número de registro de la solicitud e informe 00001 – 10000** **AÑO: Año en curso se representa solo con los últimos dos dígitos.** **Por ejemplo 2010 ⟶ 10**

TABLA 1.3 Información de solicitudes citológicas.

La información que contiene la solicitud citológica cervico vaginal es diferente a la de diversos, el contenido de la solicitud de diversos se muestra en la tabla 1.4, en el caso de la solicitud cervico vaginal se encuentra en la tabla 1.5.

SOLICITUDES DE CITOLOGÍA DE DIVERSOS

INFORMACIÓN PROPIA DE LA SOLICITUD CITOLÓGICA DE DIVERSOS
Departamento de Servicio que genera la solicitud **Fecha de la realización de la solicitud** **Lugar (puede provenir del D.F. o del interior)** **Unidad Médica (Otro Unidad Médica de la misma red del ISSSTE)** **Nombre Completo del paciente** **Edad del paciente** **Sexo del Paciente** **Número de expediente del paciente** **Procedencia de Muestra (células de descamación como laringe, faringe, esófago, recto, etc.)** **Producto (líquidos como jugo gástrico, jugo duodenal, líquido pleural, orina, saliva, etc.)** **Métodos de obtención (técnica con la que obtuvo la biopsia como aspiración, endoscopio, lavado, sondeo, frotis, etc.)**

TABLA 1.4 Información de la solicitud citológica de diversos.

INFORMACIÓN PROPIA DE LA SOLICITUD CITOLÓGICA CERVICO VAGINAL
Fecha de realización de solicitud **Número de expediente del paciente** **Nombre completo del paciente** **Edad del paciente** **Datos previos realizados al paciente** o **Fecha de la última realización de citología (primera vez, un año o menos, 2 a 3 años, más de 3 años)** o **Síntomas presentados (sangrado anormal, prurito vulvar, flujo ó ninguno)** o **Antecedentes ginecobstétricos (Puerperio postparto ó postaborto, DIU, tratamiento hormonal, embarazo ó otros tratamientos)** o **Observación realizada en la exploración (cuello aparentemente sano, cuello anormal ó no se observa cuello)** **Utensilio utilizado en la toma de nuestra (espátula de Ayre, Cepillo cervical, pipeta, abatelenguas o otros)** **Nombre completo del responsable de la toma citológica**

TABLA 1.5 Información de la solicitud citológica cervico vaginal.

1.5.2 INFORMES

A continuación se mencionan los diferentes tipos de informes y la información que cada uno debe contener:

1.5.2.1 INFORME DE PIEZAS ANATÓMICAS GENERALES Y ONCOLÓGICAS

La información que contiene dicho informe es:

- Descripción macroscópica (descripción y características de la biopsia recibida)
- Diagnóstico Histopatológico (diagnóstico final obtenido después de realizar varias técnicas de estudio)
- Fecha de realización del informe de resultados
- Nombre del patólogo que realizo dicho informe

1.5.2.2 INFORME CITOLÓGICO DE DIVERSOS

El contenido que debe tener este informe es el siguiente:

- Muestra (técnica con la que se realizará el estudio de la biopsia)
- Descripción Microscópica (descripción de las características de la biopsia)
- Diagnóstico citológico (Diagnóstico final obtenido después del proceso de estudio)

1.5.2.3 INFORME CITOLÓGICO CERVICO VAGINAL

El informe citológico cervico vaginal debe incluir la siguiente información:

- Fecha en la que se lleva a cabo la interpretación
- Características de la muestra (características de la biopsia)
- Diagnóstico citológico (diagnóstico final obtenido después del proceso de estudio)
- Se indica si es necesario repetir el estudio de la biopsia y el motivo por el cual el citotecnólogo considera que la muestra es mala para obtener un diagnóstico final
- Nombre completo del citotecnólogo
- Se señala si es requerido que la muestra sea revisada por un patólogo, así como su firma y nombre completo de dicho patólogo
- Observaciones (comentarios que surgieron en el proceso de estudio que son necesarios señalar)

1.6 PLANTEAMIENTO DE LA SOLUCIÓN

Después de recabar toda la información necesaria, el siguiente paso consiste en plantear una solución que mejore el funcionamiento del proceso de administración de solicitudes e informes de biopsias y que reduzca el tiempo de generación y entrega de los informes de resultados.
Por lo anterior se propone realizar un Sistema de administración de base de datos distribuido, con la finalidad de aprovechar las ventajas que nos pueden ofrecer este tipo de bases de datos.

1. Se configurará una red de datos inalámbrica, cumpliendo con esto la condición de no realizar modificaciones a la construcción, así evitamos cualquier conflicto en la instalación de la red de datos. Al ser una red de datos inalámbrica tiene la posibilidad de crecer de manera fácil y rápida, sin afectar las operaciones en los diferentes nodos.

2. Cada usuario manejará datos diferentes, por lo cual ocuparemos la ventaja de poder localizar a los datos en distintos nodos de la red de datos, con la finalidad de cumplir con las necesidades que cada usuario tenga que cubrir. Esto ofrecerá un acceso más rápido a los datos, ya que los usuarios tendrán acceso solo a un subconjunto de datos.

3. Se ofrecerá un procesamiento rápido de los datos, ya que la carga de trabajo se repartirá en los diferentes nodos de la red de datos, con ello un conjunto de datos podrá ser procesado en diferentes nodos al mismo tiempo.

4. La interface del usuario será sencilla, amigable y fácil de utilizar.

5. Si algún nodo de la red inalámbrica falla, las operaciones que se estén realizando se trasladarán hacía otro nodo, con la finalidad de no detener el proceso que se está llevando a cabo en ese momento.

6. Se contemplarán aspectos de seguridad enfocados a la distribución de los datos, así como a la red de datos con la que se contará, dicha seguridad será independiente de la que tenga establecida la red nacional del ISSSTE. Posteriormente a los equipos se les aplicarán las medidas de seguridad que el ISSSTE crea convenientes.

7. La totalidad del diseño contará con la utilización de Software Libre, para evitar problemas con las licencias, por lo tanto se propone realizar la comparación de las siguientes aplicaciones con la finalidad de elegir la más adecuada para llevar a cabo el desarrollo del sistema:

    - Manejador de Bases de Datos
        - MySQL 5.1.50
        - PostgreSQL 8.3.10-1
    - Software de desarrollo de la aplicación
        - .NET 4
        - Java 6
        - JDBC versión 4
    - Diseño de la aplicación
        - UML

Se contará con distintos nodos, los cuales tendrán diferentes funcionalidades dentro de nuestro sistema, a fin de ayudar al mejoramiento del proceso.

Cada nodo contará con dos claves de acceso, una para la clave de la red de datos y otra para iniciar sesión en el servidor de base de datos y poder realizar sus actividades correspondientes. Al tener estas dos claves nos aseguramos que solo las personas autorizadas tendrán acceso directo a los datos que se encuentren en la base de datos.

Nodo Jefatura

Este nodo es el más importante, ya que a través de él se podrá tener acceso a toda la base de datos y al monitoreo de las actividades de cada médico especialista para poder llevar el status en el que se encuentran las solicitudes asignadas a estos y detectar cualquier problema que cause retraso. Este nodo proporcionará facultades para la creación de cuentas de usuarios, acceso a los registros y diagnósticos realizados por los médicos especialistas y el personal administrativo. También se podrá editar diagnósticos y solicitudes, y generar reportes semanales, mensuales y anuales. En este nodo se podrá realizar la revisión de los reportes enviados por otros nodos para poder proporcionar una solución.

Nodo citológico, patólogo, oncólogo

En los nodos anteriormente citados se realizarán los informes de resultados, se podrán realizar consultas y modificaciones de ellos. Existe una excepción en la posibilidad de modificar, esto es, si

el médico especialista ha finalizado el proceso de generación del informe, no podrá realizar ninguna modificación a menos que el **nodo jefatura** autorice volver a poner ese informe en línea. Tampoco se podrá borrar algún informe. El borrado únicamente puede ser realizado por el **nodo jefatura**. Después de haber terminado el llenado de la información del informe de resultados podrá generar y enviar al nodo mecanografía dicho informe.

Nodo recepción

En este nodo se podrá realizar el registro de las solicitudes y biopsias que llegan al Departamento de Servicio, así como será responsable de tener un registro de las solicitudes que presenten algún problema ó sea requerido por el paciente para una segunda revisión, este registro se enviará al **nodo jefatura**.
También realizará la asignación de las solicitudes a los diferentes nodos de especialidades, teniendo en cuenta el tipo de solicitud que se tenga, al igual que enviará los datos del paciente y del médico tratante.

Nodo mecanografía

En este nodo se podrán recibir los informes de resultados generados por los nodos de especialidades, con la finalidad de llevar un control de las solicitudes que han sido ya cubiertas. Se realizará la impresión de dichos informes para ser enviados al Departamento de Archivo.

En la figura 1.5 se presenta un plano señalando la ubicación de cada nodo, así como la configuración de la red de datos inalámbrica.

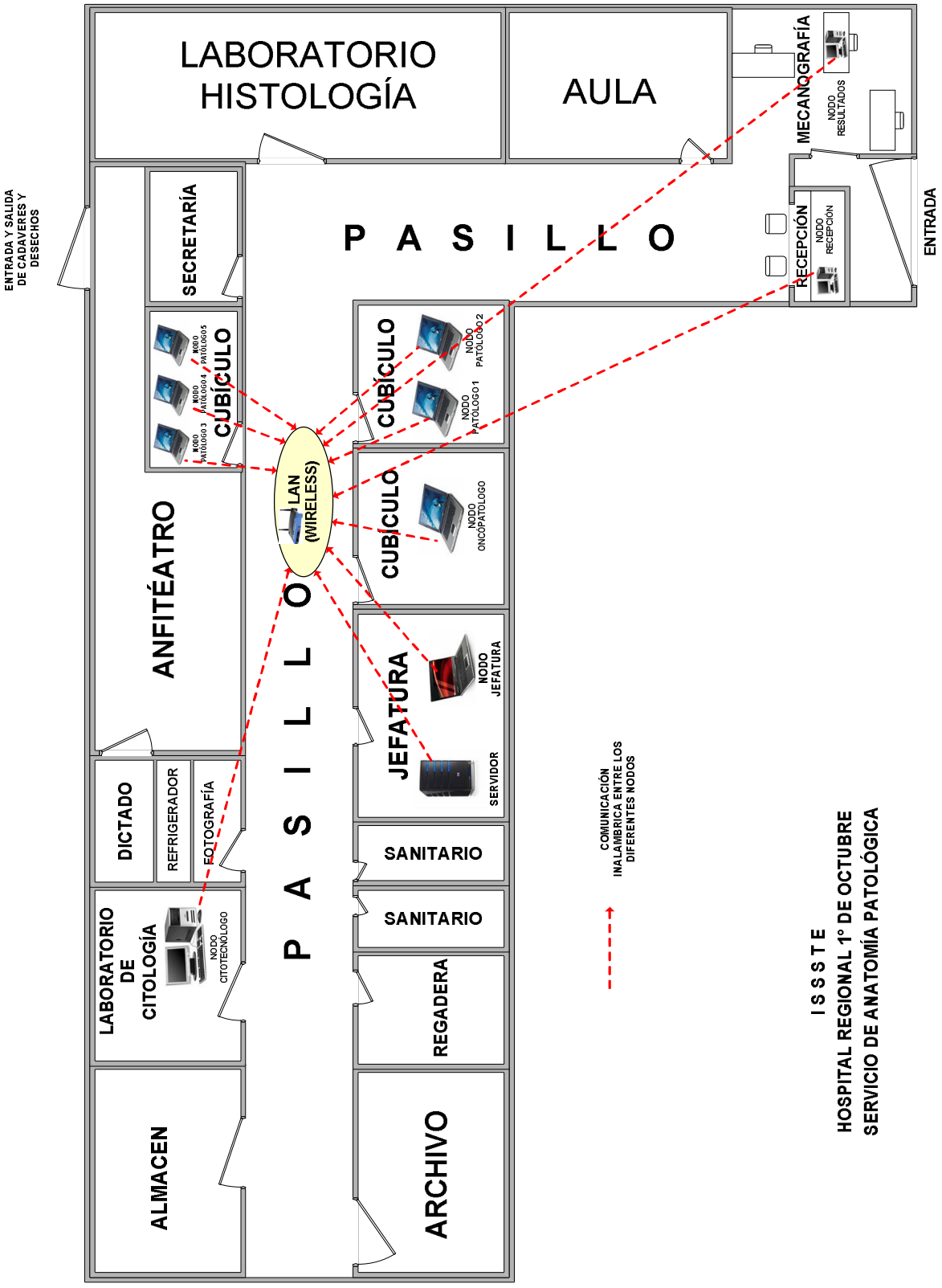


FIGURA 1.5 Plano del Departamento de Servicio de Anatomía Patológica con la configuración de la red de datos inalámbrica.

1.7 REQUERIMIENTOS DE SOFTWARE

Para determinar que software se utilizará en el desarrollo de la aplicación se contemplarán dos factores que compondrán dicha aplicación:

1. *Backend:* esta parte es la encargada de procesar la entrada de los datos que ingresa el usuario, es decir son los procesos que utiliza el encargado de la aplicación para resolver las peticiones del usuario.
2. *Frontend:* es la parte que interactúa con los usuarios, es decir, son todos los formularios que deben ser rellenados con información propuesta por el usuario.

Al separar el desarrollo de la aplicación en dichos factores se resuelven las necesidades de los usuarios teniendo un mejor control en la recolección y procesamiento de los datos.

En las secciones siguientes se explicarán los requerimientos necesarios para el desarrollo del *backend y frontend.*

1.7.1 REQUERIMIENTOS PARA EL *BACK-END*

Un manejador de base de datos se encargará del procesamiento de los datos para la aplicación que se desarrollará, por esta razón se realizará una comparación entre PostgreSQL y MySQL que son dos manejadores de bases de datos con licencia libre, posteriormente se elegirá el que cubra las necesidades de dicha aplicación.

1.7.1.1 POSTGRESQL Y MYSQL

Antes de realizar la comparación entre los dos manejadores de bases de datos es importante conocer cada una de sus características, su arquitectura, etc. para poder seleccionar el que más se adecue a nuestras necesidades.

1.7.1.1.1 POSTGRESQL

PostgreSQL es el manejador la base de datos relacional de código abierto, distribuido bajo licencia BSD (*Berkeley Software Distribution*). Utiliza un modelo cliente/servidor basado en multiprocesos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. (Martínez Rafael, 2009).

PostgreSQL dispone de versiones para prácticamente todos los sistemas operativos y cumple totalmente con las propiedades ACID (*Atomicity, Consistency, Isolation, Durability*). Tiene soporte para llaves foráneas, *joins*, vistas, disparadores y procedimientos almacenados. Incluye la mayoría de los tipos de datos de SQL92 y SQL99 y, así mismo, soporta el almacenamiento de grandes

objetos binarios, como imágenes, sonidos y vídeos, también tiene interfaces de programación nativas. (PostgreSQL Global Development Group, 1996).

PostgreSQL ofrece características tales como control concurrente multiversión, *point in time recovery*, *tablespaces*, replicación asíncrona, transacciones anidadas, *savepoints*, copias de seguridad en caliente/en línea, un planificador/optimizador de consultas y *write ahead logging* para ser tolerante a fallos de hardware. Soporta juegos de caracteres internacionales, codificaciones de caracteres *multibyte, unicode* y realiza ordenaciones dependiendo de la configuración del idioma local, de la diferenciación de mayúscula, minúsculas y del formato. Es altamente escalable tanto en la cantidad de datos que puede manejar como en el número de usuarios concurrentes que puede atender. Hay sistemas activos en producción con PostgreSQL que manejan más de 4 terabytes de datos. (Martínez Rafael, 2009).

HISTORIA

PostgreSQL se inicia en 1986 con un proyecto del profesor *Michael Stonebraker* y un equipo de desarrolladores de la Universidad *Berkeley* (California, USA), cuyo nombre original era POSTGRES. En su diseño se incluyeron algunos conceptos avanzados en bases de datos y soporte parcial a la orientación a objetos. (Martínez Rafael, 2009).

POSTGRES fue comercializado inicialmente por *Illustra,* una empresa que posteriormente se integro a *Informix*, quien comercializaba el conocido SGBD del mismo nombre, recientemente absorbida por IBM y su DB/2. En 1993 se liberó la versión 4.5 y oficialmente se dio por finalizado dicho proyecto. (Lockhart Thomas, n.d.).

En 1994, *Andrew Yu y Jolly Chen* incluyen SQL en Postgres para posteriormente liberar su código en la web con el nombre de Postgres95. El proyecto constituía múltiples cambios al código original lo que permitió la mejora de su rendimiento y legibilidad. En 1995 salió la versión 1.0 de Postgre95, el código era 100% ANSI C, un 25% más corto en relación con la versión 4.2 y un 30-50% más rápido. (Martínez Rafael, 2009).

En 1996 el nombre cambió a PostgreSQL retomando la secuencia original de versiones, por lo que se liberó la versión 6.0. En el año 2004 la versión estable oficial es la 7.4.6, mientras que la versión 8.0 estaba en fase de estabilización. (Dataprix, n.d.).

CARACTERÍSTICAS

PostgreSQL destaca por su lista de prestaciones que lo hacen capaz de competir con cualquier SGBD comercial:

- Altamente Extensible
  - PostgreSQL soporta creación de operadores, métodos de acceso y tipos de datos definidos por el usuario.
  - PostGIS, añade mas soporte para tipos creados por el usuario y GeoEspaciales
- Soporta especificación SQL2003
- Integridad Referencial
- API Flexible
  - Java/JDBC , Object Pascal, Python, Perl, PHP, ODBC, Ruby, TCL, C/C++, y Pike
- MVCC (*Multi-Version Concurrency Control*)
  - Evita bloqueos innecesarios.
  - Opera a nivel de transacciones (diferentes versiones de la base de datos).
- Diferentes tipos de niveles de aislamiento
- Cursores
- Lenguajes soportados
  - SQL (instalado en todas las bases de datos)
  - PL/pgSQL
  - C (Por default, instalado en todas las bases de datos)
  - *Tcl ,Perl,Python,Lua*
- *Write Ahead Logging* (WAL)
  - Registro de cambios antes de que estos sean escritos en la base de datos.
  - Permite *PITR -Point in Time Recovery*
  - Existe un registro de las transacciones a partir del cual podremos restaurar la base de datos.
  - Cualquier cambio que no fue escrito en la base de datos puede ser recuperado usando el dato que fue previamente registrado.
- *EXPLAIN*
  - Muestra el plan de ejecución de la consulta que el planificador genera, algoritmos para la unión, búsqueda secuencial o por índice, etc.

Las limitaciones que se encuentran en este gestor de bases de datos se muestran en la tabla 1.6.

Límite	Valor
Máximo tamaño base de datos	Ilimitado (Depende del sistema operativo de almacenamiento)
Máximo tamaño de tabla	32 TB
Máximo tamaño de fila	1.6 TB
Máximo tamaño de campo	1 GB
Máximo número de filas por tabla	Ilimitado
Máximo número de columnas por tabla	250 - 1600 (dependiendo del tipo de dato)
Máximo número de índices por tabla	Ilimitado

TABLA 1.6 Limitaciones que presenta el manejador de base de datos PostgreSQL.

ARQUITECTURA

La arquitectura de Postgres consiste en los siguientes procesos cooperativos mostrados en la figura 1.6:

- *Postmaster*: Es un proceso demonio supervisor, se encarga de ejecutar un nuevo servidor para cada cliente que solicite una conexión (Lockhart Thomas, n.d.). Un único *postmaster* controla una colección de bases de datos dadas en un único host. Entendiendo que una colección de bases de datos se suele llamar instalación. (Martínez Rafael, 2009).

- *Frontend*: Es la aplicación sobre la que trabaja el usuario. (Lockhart Thomas, n.d.)
- Proceso postgres: Uno o más servidores de bases de datos en segundo plano. (Lockhart Thomas, n.d.).

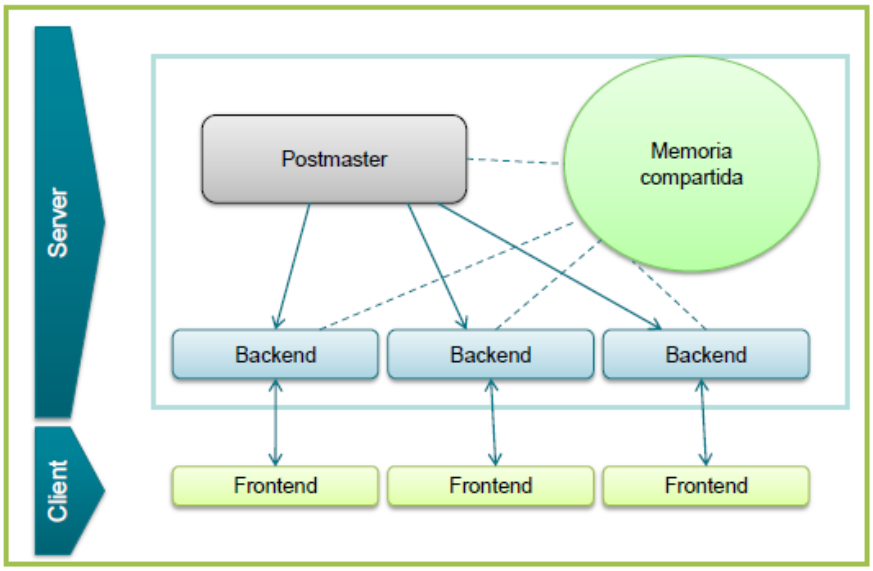


FIGURA 1.6 Procesos cooperativos de la arquitectura de PostgreSQL.

Las aplicaciones de *frontend* que acceden a una determinada base de datos dentro de una instalación hacen llamadas a librerías. Las librerías envían peticiones de usuario a través de la red al *postmaster*, el cual inicia un nuevo proceso en el servidor y conecta el proceso de *frontend* al nuevo servidor. A partir de este punto, el proceso de *frontend* y el servidor en *backend* se comunican sin la intervención del *postmaster*. (Martínez Rafael, 2009).

Mientras el *postmaster* siempre se está ejecutando al estar esperando peticiones, los procesos de *frontend* como los de *backend* se ejecutan continuamente. Aunque, la aplicación *frontend* es un proceso en un único hilo, por el momento *libpq* no soporta conexiones multihilos entre el *frontend* y el *backend*. Una implicación de esta arquitectura es que el *postmaster* y el proceso *backend* siempre se ejecutan en la misma máquina, mientras que la aplicación en *frontend* puede ejecutarse desde cualquier sitio. (Dataprix, n.d.).

SEGURIDAD

Es posible restringir el acceso a usuarios o a direcciones IP modificando las opciones del archivo *pg_hba.conf*, que se encuentra en */postgresql/pg_hba.conf*. Este archivo, junto con */postgresql/postgresql.conf* son particularmente importantes, porque algunos de sus parámetros de configuración por defecto provocan multitud de problemas al conectar inicialmente y porque en ellos se especifican los mecanismos de autenticación que usará PostgreSQL para verificar las credenciales de los usuarios. Para habilitar la conexión a PostgreSQL desde clientes remotos, debemos verificar el parámetro *tcpip_socket* = true en el fichero */postgresql/postgresql.conf.* (Lockhart Thomas, n.d.).

Para examinar los métodos de autenticación y las posibilidades de conexión de clientes externos, se debe revisar el fichero */postgresql/ pg_hba.conf,* donde se señala la acción que hay que emprender para cada conexión proveniente de cada *host* externo, o grupo de *hosts*. (Lockhart Thomas, n.d.).

1.7.1.1.2 MYSQL

Es un sistema de gestión de bases de datos relacional, creada por MySQL AB, empresa sueca, la cual tiene el copyright del código fuente del servidor SQL. MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque se tiene una que ofrece soporte técnico , y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL. (Daniel Pecos, n.d.).

MySQL incluye los elementos necesarios para instalar programas, preparar niveles de acceso de usuario, administrar el sistema y proteger de volcado de datos. Puede desarrollar sus propias aplicaciones en la mayor parte de los lenguajes de programación y ejecutarlos en casi todos los sistemas operativos. Utiliza el lenguaje de consulta estructurado (SQL). (Ian Gilfillan, n.d).

HISTORIA

Surgió alrededor de la década del 90, *Michael Windenis* comenzó a usar mSQL para conectar tablas usando sus propias rutinas de bajo nivel (ISAM). Tras las primeras pruebas, llegó a la conclusión de que mSQL no era flexible ni rápido para lo que necesitaba, por lo que tuvo que desarrollar nuevas funciones. Esto resulto en una interface SQL a su base de datos, totalmente compatible a mSQL. (Marin, Alvaro, n.d.)

El origen del nombre MySQL no se sabe con certeza de donde proviene, por un lado se dice que en sus librerías han llevado el prefijo *my* durante los diez últimos años, por otra parte, la hija de uno de los desarrolladores se llama *My.* Así que no está claramente definido cuál de estas dos causas han dado lugar al nombre de este conocido gestor de bases de datos. (Marin, Alvaro, n.d.)

CARACTERÍSTICAS

Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha.

- Está desarrollado en C/C++.
- Se distribuyen ejecutables para cerca de diecinueve plataformas diferentes.
- La API se encuentra disponible en C, C++, Eiffel , Java, Perl, PHP, Python, Ruby y TCL.
- Está optimizado para equipos de múltiples procesadores.
- Es muy destacable su velocidad de respuesta.
- Se puede utilizar como cliente-servidor o incrustado en aplicaciones.
- Cuenta con un rico conjunto de tipos de datos.
- Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes para poder optimizar el SGBD a cada caso concreto.
- Su administración se basa en usuarios y privilegios.
- Se tiene constancia de casos en los que maneja cincuenta millones de registros, sesenta mil tablas y cinco millones de columnas.
- Sus opciones de conectividad abarcan TCP/IP, *sockets* UNIX y *sockets* NT, además de soportar completamente ODBC.
- Es altamente confiable en cuanto a estabilidad se refiere.

MySQL tiene como principal objetivo ser una base de datos fiable y eficiente. Ninguna característica es implementada en MySQL si antes no se tiene la certeza que funcionará con la mejor velocidad de respuesta y, por supuesto, sin causar problemas de estabilidad.

ARQUITECTURA

La arquitectura (véase la figura 1.7) tiene como característica separar el motor de almacenamiento del resto de los componentes de la arquitectura. Es decir, el diseño del gestor está preparado para poder cambiar el gestor de almacenamiento. Las utilidades y herramientas de MySQL son los programas y aplicaciones que se incluyen con la distribución del gestor, o se pueden instalar como aplicaciones adicionales. Estas incluyen las herramientas de *backup*, el navegador de consultas *(QueryBrowser),* las aplicaciones administrativas de interfaz gráfico y la herramienta de diseño MySQL *Workbench*, entre otras. (Sicilia Miguel Angel, 2008)

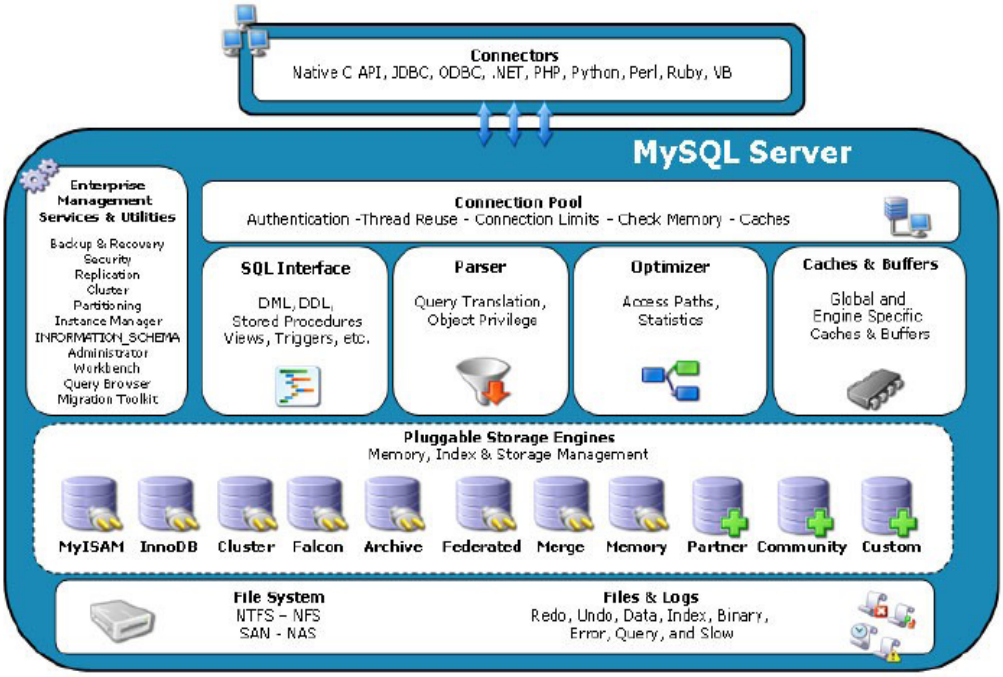


FIGURA 1.7 Arquitectura de MySQL.

- Motores de almacenamiento

La arquitectura de motores se encarga de hacer una interface abstracta con funciones comunes de gestión de datos en el nivel físico. De ese modo, el gestor de almacenamiento puede intercambiarse, e incluso un mismo servidor MySQL puede utilizar diferentes motores de almacenamiento para diferentes bases de datos o para diferentes tablas en la misma base de datos. (Sicilia Miguel Angel, 2008).

- Conectores

Son bibliotecas en diferentes lenguajes de programación que permiten la conexión con servidores MySQL y la ejecución de consultas. (Sicilia Miguel Angel, 2008).

- Gestor de conexiones

La gestión de conexiones es responsable de mantener las múltiples conexiones de los clientes. El gestor de conexiones de MySQL puede configurarse para limitar el número de conexiones concurrentes, y también implementa un pool de conexiones. Se ocupa de la autentificación de los usuarios. El servidor puede configurarse para soportar certificados X.509. (Sicilia Miguel Angel, 2008).

- Procesamiento y optimización de consultas

Cuando llega una consulta al gestor de MySQL, se produce una representación intermedia de la misma. A partir de esa representación, MySQL toma una serie de decisiones, que pueden incluir el determinar el orden de lectura de las tablas, el uso de ciertos índices, o la re-escritura de la consulta en una forma más eficiente. (Sicilia Miguel Angel, 2008).

- Caché de consultas

Donde se guardan las consultas y sus resultados enteros. De este modo, el procesador de consultas busca en primer lugar en la caché. (Sicilia Miguel Angel, 2008).

- Control de Concurrencia

Es el mecanismo que se utiliza para evitar que lecturas o escrituras simultáneas a la misma porción de datos terminen en inconsistencias o efectos no deseados. En este caso se utilizan los bloqueos para controlar el acceso. (Sicilia Miguel Angel, 2008).

- Gestión de transacciones y recuperación

La gestión de transacciones permite dotar de semántica a una consulta o a un conjunto de consultas que se declaran como una sola transacción. Con lo anterior se entiende que si una consulta no se lleva a cabo, el servidor anula el efecto parcial de la parte que ya se haya ejecutado. Con la parte de la recuperación se permite realizar *rollback* de una transacción. (Sicilia Miguel Angel, 2008).

1.7.1.1.3 COMPARATIVA ENTRE POSTGRESQL Y MYSQL

En la figura 1.8 se observa la posición en la que se hallan los dos manejadores anteriormente descritos con relación a otros manejadores que se encuentran en el mercado, con ello se realizará una comparación entre estos para poder elegir el manejador que cumpla con nuestras necesidades.

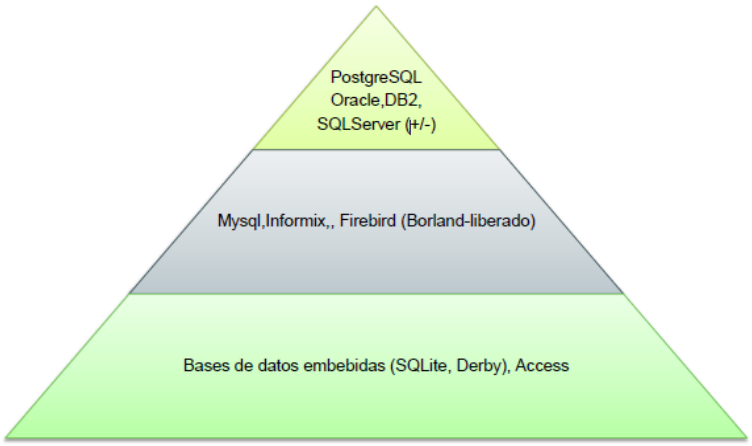


FIGURA 1.8 Pirámide de manejadores de bases de datos.

Se manejará un puntaje para poder evaluara cada uno de los manejadores, este puntaje se describe en la tabla 1.7.

Valores
1 no satisfactorio
2 satisfactorio
3 muy satisfactorio

TABLA 1.7 Puntaje de evaluación.

En la tabla 1.8 se muestra la comparación que se efectúa entre los manejadores.

Concepto	Valoración MySQL	Valoración PostgreSQL
Modelo de licencia, precio	2 La licencia dual que ofrece siempre será un aspecto que se tendrá que tener en cuenta si se utiliza este manejador	3 La licencia BSD no nos limita en ningún aspecto.
Soporte por parte del fabricante	3 Maneja la opción de contratar soporte en varias modalidades	2 No ofrece soporte directamente, este se da a través de otras empresas especializadas.
Prestaciones en creación de las estructuras (tablas, índices, etc.)	2 Es francamente fácil de manejar en este aspecto	3 Es muy potente en este aspecto, ofrece prácticamente todas las prestaciones contempladas en el estándar, y tiene un fantástico sistema de extensión.
Prestaciones en tipos de datos	2 Los tipos de datos soportados así como los operadores incluidos en el manejador de base de datos son suficientes para algunas aplicaciones	3 Contiene un sistema de extensión y definición de datos y dominios
Prestaciones en consultas simples	3 Es una de sus características más potentes, que lo hace colocarse entre los manejadores más potentes	3 Es muy potente en este aspecto, que lo hace ser uno de los mejores.
Prestaciones en consultas complejas	2 Anteriormente no soportaba subconsultas lo que implicaba un mayor esfuerzo, ahora ya es soportado y a un nivel igual que otro manejador	3 Soporta subconsultas, vistas, etc., y su implementación es muy estable.
Prestaciones en manipulación de datos	3 Incluye multitud de opciones no estándares para cargar datos externos, insertar o actualizar sobre la base de consultas complejas y la utilización de operadores como condiciones para la manipulación.	3 Incluye multitud de opciones no estándares para cargar datos externos, insertar o actualiza sobre la base de consultas complejas y utiliza operadores de condición en la manipulación

Facilidad en la administración de usuarios	3 Soporta muy bien el estándar en cuanto a la creación de usuarios y la gestión de sus privilegios con *GRANT* y *REVOKE*.	2 Soporta muy bien la creación de usuarios y la gestión de sus privilegios. Su sistema múltiple de autenticación lo hace demasiado complejo.
Facilidad en la gestión de copias de seguridad	3 Dispone de herramientas de volcado, como la posibilidad de copia binaria de la base de datos. Comprende conectores que proporcionan la capacidad de realizar *backups* de la base de datos en caliente.	3 Se dispone de herramientas de volcado, como de la posibilidad de realizar una copia binaria de la base de datos

TABLA 1.8 Comparativa entres los manejadores PostgreSQL y MySQL.

1.7.1.1.4 SELECCIÓN DEL MANEJADOR DE BASE DE DATOS.

Al finalizar la comparación realizada en la tabla 1.8, donde se manejaron puntajes, se obtienen los siguientes resultados mostrados en la tabla 1.9.

MySQL		**PostgreSQL**	
Valor	Cantidad	Valor	Cantidad
1	0	1	0
2	4	2	2
3	5	3	7

TABLA 1.9 Puntaje final de la comparación de manejadores.

Como se observa en la tabla 1.9 el manejador PostgreSQL obtiene mayor cantidad de valor 3 y una cantidad menor del valor 2, lo que indica que el manejador se califica como muy satisfactorio. El manejador que más se acoplo a las necesidades del sistema de base de datos distribuido es PostgreSQL por las siguientes razones:

- Puntaje: Obtener un alto puntaje en la comparativa que se realizó.
- Costo: No tiene ningún costo, es un software libre publicado bajo licencia BSD, es decir que la redistribución y el uso es bajo forma de código fuente o binario, con o sin modificaciones, están autorizados siempre que se cumplan ciertas condiciones.
- Características Generales: Ofrece al sistema el soporte de implementar todas las características de una base de datos distribuida, con miras en el futuro de poder comunicar el sistema distribuido con otros manejadores sin tener que cambiar la estructura, dado que brinda conectividades heterogéneas. Otro punto sobresaliente es tener la confianza de contar con varios aspectos de seguridad que el otro manejador no

contempla, esto nos lleva a poder configurar el sistema del lado back-end para tener un poco más protegida la información. También contempla características de una base de datos centralizada dando la libertad de crear tipos de datos, tablespace, respaldos en línea, etc. Por último soporta varios lenguajes de programación, con esto facilita la comunicación entre el *back-end* y el *front-end*.

- Modelo subyacente: Relacional y orientada a objetos.

- Portabilidad: Es multiplataforma (Windows y varias distribuciones Linux), en cuanto a lenguajes de programación: PL/PgSQL , C, C++, Java PL/Java web, PL/Perl, plPHP, PL/Python, PL/Ruby, PL/sh, PL/Tcl, PL/Scheme.

1.7.2 REQUERIMIENTOS PARA EL *FRONT-END*

La plataforma de software ayudará al diseño de la interface del usuario, a la recolección de los datos y a la conexión con el manejador de base de datos, por lo cual se realizará una comparación entre JAVA y .NET para elegir la plataforma que brinde mayor cantidad de recursos para poder realizar el diseño de la aplicación cubriendo todas las necesidades existentes.

1.7.2.1 JAVA Y .NET

Como primer punto es necesario conocer las características de cada una de las plataformas para poder realizar una selección fundamentada en lo que cada una de ellas nos puede ofrecer.

1.7.2.1.1 JAVA

Java es una plataforma de software desarrollada por *Sun Microsystems*, de tal manera que los programas creados en ella puedan ejecutarse sin cambios en diferentes tipos de arquitecturas y dispositivos computacionales. (Segura Salazar Juan, n.d.).

Consta de las siguientes tres partes:

- El lenguaje de programación.
- La máquina virtual de Java.
- El API de Java.

Originalmente llamado OAK por los ingenieros de *Sun Microsystems*, Java fue diseñado para ejecutarse en computadoras incrustadas. Sin embargo, en 1995, dada la atención que estaba produciendo la Web, *Sun Microsystems* la distribuyó para sistemas operativos tales como Microsoft Windows. El lenguaje mismo se inspira en la sintaxis de C++. (Segura Salazar Juan, n.d.).

Utilizando una tecnología llamada **JSP**, se hizo muy fácil escribir páginas dinámicas para sitios de Internet. Sumado a esto, la tecnología de JavaBeans, al incorporarse con JSP, permitía adaptar al mundo web el patrón MVC que ya se había aplicado con éxito a interfaces gráficas. (Herb Shildt, 2008).

El modelo J2EE permite una separación entre la presentación de los datos al usuario (*JSP o Applets*), el modelo de datos (EJB), y el control (*Servlets*). Los programas en Java generalmente son compilados a un lenguaje intermedio llamado *bytecode*, que luego son interpretados por una máquina virtual (JVM). Esta última sirve como una plataforma de abstracción entre la máquina y el lenguaje permitiendo que se pueda escribir el programa una vez, y correrlo en cualquier lado. (Segura Salazar Juan, n.d.).

JAVA es un lenguaje orientado a objetos diseñado para ser multiplataforma y poder ser empleado el mismo programa en diversos sistemas operativos. Esta característica, junto con la posibilidad de emplearlo para crear *applets* e insertarlos en páginas HTML, o mediante *servlets* y páginas jsp generar código HTML dinámico, así como la capacidad de acceder a bases de datos, hace de JAVA uno de los lenguajes más utilizados en la actualidad. (Segura Salazar Juan, n.d.). Todas estas características mencionadas se muestran en la figura 1.9.

Java es un lenguaje relativamente sencillo, debido a que la funcionalidad se encuentra en clases que forman parte del API de JAVA. Constantemente están surgiendo nuevos APIs, que proporcionan extensiones a las características del lenguaje.

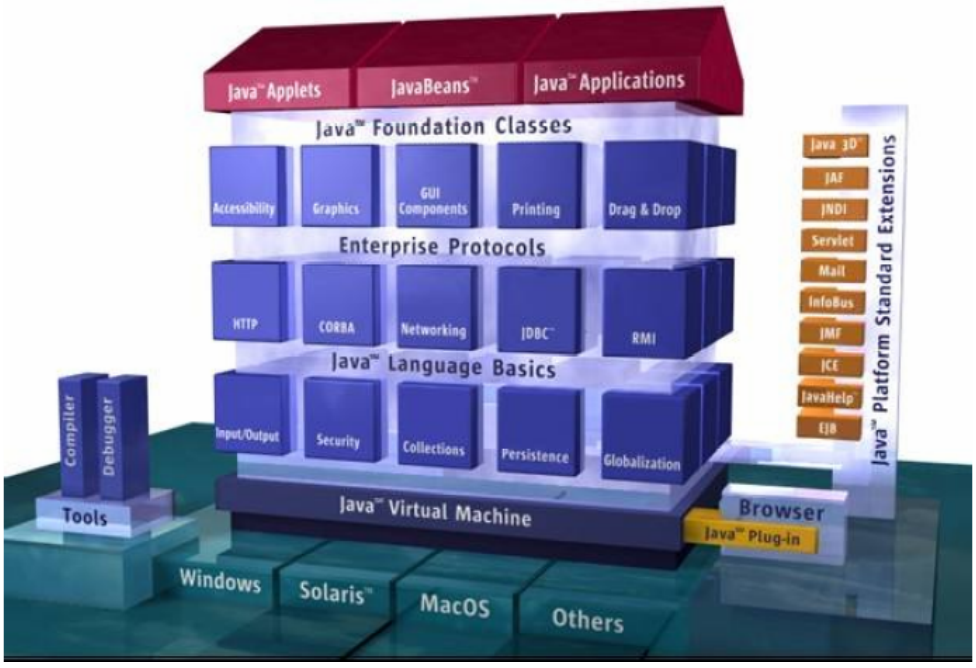


FIGURA 1.9 Diagrama de las características de JAVA.

CARACTERÍSTICAS

Las principales características de Java son

- **Simple**
  Debido a su semejanza con C y C++ resulta más fácil aprender Java.
  - Java tiene una clase *String*, que permite un mejor manejo que los *arrays* de terminación nula del C y C++.
  - Maneja los comandos en línea de diferente manera que C++
  - Java tiene un sistema automático de asignación y liberación de memoria (recolector de basura) que mejora mucho los sistemas del C++.

- **Orientado al objeto**
  - Java da buen soporte a las técnicas de desarrollo OOP y en resumen a la reutilización de componentes de software.

- **Distribuido**
  Java se ha diseñado para trabajar en ambiente de redes, contiene una gran biblioteca de clases para la utilización del protocolo TCP/IP, incluyendo HTTP y FTP.

- **Interpretado**
  El compilador Java traduce cada fichero fuente de clases a código de bytes, que puede ser interpretado por todas las máquinas que den soporte a un visualizador que funcione con Java. Este *bytecode* no es específico de una máquina determinada, por lo que no se compila y enlaza como en el ciclo clásico, sino que se interpreta.

- **Sólido**
  El código Java no se quiebra fácilmente ante errores de programación. Así en Java no es posible escribir en áreas arbitrarias de memoria ni realizar operaciones que corrompan el código. En resumen se eliminan muchas de las posibilidades de "trucos" que ofrecía el C y C++.

- **Seguro**
  Como Java suele funcionar en ambiente de redes el tema de seguridad debe interesar en sobremanera. Actualmente se está trabajando en encriptar el código.

- **Arquitectura neutral**
  - El compilador crea códigos de byte que se envía al visualizador solicitado y se interpreta en la máquina que posee un intérprete de Java o dispone de un visualizador que funciona con Java.

- **Portable**
  Al ser de arquitectura neutral es altamente portable, pero esta característica puede verse de otra manera: Los tipos estándares están igualmente implementados en todas las máquinas por lo que las operaciones aritméticas funcionaran igual en todas las máquinas.

- **Alto desempeño**
  - al ser código interpretado, la ejecución no es tan rápida como el código compilado para una plataforma particular. El compilador Java suele ofrecer la posibilidad de compilar *bytecode* en código máquina de determinadas plataformas, y según Sun este código resulta de una eficacia similar a compilaciones de C y C++.

- **Multihilos**
  Java se utiliza para realizar aplicaciones en las que se ocurra más de una cosa a la vez. Java, apoyándose en un sistema de gestión de eventos basado en el paradigma de condición y monitores C.A.R. permite apoyar la conducta en tiempo real e interactivo en programas.

- **Dinámico**
  Java utiliza un sistema de interfaces que permite aligerar dependencias. Como resultado, los programas Java pueden permitir nuevos métodos y variables en un objeto de biblioteca sin afectar a los objetos dependientes.

1.7.2.1.2 .NET

La plataforma .NET nace el año 2000 como un proyecto de *Microsoft Corp*. Con la idea de crear una plataforma de desarrollo de software que permitiera unificar todos los productos de Microsoft, desde las herramientas de desarrollo hasta el sistema operativo. (Cifuentes Francisco, n.d.).

La idea central es generar un software independiente de la plataforma de hardware y del sistema operativo, poniendo especial énfasis en el desarrollo de aplicaciones distribuidas sobre redes y la interoperabilidad de lenguajes, cada uno asociado a diferentes dominios de resolución de problemas. (Cifuentes Francisco, n.d.).

La primera versión estable de .NET fue lanzada el 5 de enero de 2002 y la última versión estable es la 3.5.30729.1 del 11 de Agosto de 2008.

Los cambios en .NET están proyectados para:

- Simplificar el lenguaje y hacerlo más coherente.
- Agregar nuevas características solicitadas por usuarios.
- Hacer el código más sencillo de leer y mantener.
- Ayudar a los programadores a evitar los errores de programación.
- Crear aplicaciones más sólidas y más sencillas de depurar.

.NET ofrece numerosas características nuevas y mejoradas, como herencia, interfaces y sobrecarga, que lo convierten en un eficaz lenguaje de programación orientado a objetos. Otra característica nueva de Visual Basic .NET incluye el control estructurado de excepciones, atributos personalizados y compatibilidad con CLS (Especificación de lenguajes comunes). (Cifuentes Francisco, n.d.).

1.7.2.1.3 COMPARATIVA ENTRE JAVA Y .NET

En la tabla 1.10 se muestra la comparación que se efectúa entre las plataformas.

CARACTERÍSTICA	.NET	JAVA
Compañía propietaria	Microsoft Corp.	Sun Microsystems
Lenguajes de programación compatibles	C#, Visual Basic .NET, ASP.NET, Turbo Delphi for .NET, C++, C, Java, J#, Perl, Pitón, Fortran, Cobol.NET, ADA, APL, Haskell, Javascript, Lisp, Prolog, Smalltalk y muchos otros, que deben cumplir con las reglas de CLS (*Common Language Specification*)	Java
Paradigma de programación	Orientado a objetos, declarativo y funcional	Orientado a objetos de forma no estricta, no permite herencia múltiple ni sobrecarga de operadores.
Premisas de la tecnología	Disponer de la tecnología apropiada para un problema específico. Distribuir aplicaciones en forma de servicios. Comunicar y combinar aplicaciones independientes del modelo de programación, plataforma y lenguaje. Integrar todos los productos Microsoft, desde herramientas de desarrollo al sistema operativo.	Escríbelo una vez, ejecútalo donde sea. Permitir la ejecución de un mismo programa compilado en múltiples sistemas operativos. Utilizar POO Incluir soporte para trabajo en red. Trabajar de forma segura en sistemas remotos. Desarrollar aplicaciones distribuidas.
Entorno de ejecución	.NET Framework	Java Runtime Environment
Implementaciones de la máquina abstracta	CLR,MONO, DotGNU Portable .NET, SSCLI	JVM, KVM
Formato del código compilado	Compilador JIT del CLR ejecuta código IL | (MSIL) *bytecode*	Compilador JIT de la JVM ejecuta *.class*, código intermedio *bytecode*
Kit de desarrollo	Distribuidos en múltiples lenguajes de programación, permite crear cualquier tipo de aplicación.	J2SE – aplicaciones de escritorio, J3EE – desarrollo empresarial, J2ME . dispositivos con recursos limitados
Entornos de desarrollo	Visual Studio .NET	Eclipse, Netbeans, Visual Studio .NET y muchos otros.
Licenciamiento	Comercial y educacional	GNU
Implementaciones más relevantes en el API	▪ Interacción con dispositivos periféricos	▪ JDBC (Conexión con bases de datos)

	▪ Manejo de datos ▪ Gestión de memoria ▪ Transmisión por XML y TCP/IP ▪ Manejador de excepciones ▪ ASP.NET ▪ Sistema de ventanas ▪ Despliegue gráfico GDI+ ▪ Seguridad e interacción con otras aplicaciones ▪ Cadenas de caracteres y expresiones regulares ▪ Colecciones ▪ Windows API, API ▪ Win32	▪ RMI (Invocación remota de métodos) ▪ Reflexión ▪ Asertos ▪ Expresiones Regulares ▪ Manejo de Excepciones ▪ *Logging* ▪ *Parser XML* ▪ Plantillas ▪ Colecciones ▪ Metadatos ▪ Enumeraciones ▪ *Autoboxing / unboxing* ▪ Servicios Web ▪ JavaBeans ▪ Sockets ▪ Transacciones ▪ Servicios de nombre ▪ GUI: AWT, SWT ▪ Autenticación ▪ Persistencia de Objetos ▪ Gestión de memoria
Acceso a características de bajo nivel	Directamente mediante código no administrado	Utilizando JNI
Posee un estándar de desarrollo para componentes	No, solo ofrece recomendaciones	Un estándar de desarrollo para componentes de acceso a datos y reglas de negocio.
Tipos de estructuras que es posible crear	Clases, interfaces, *struct y enum*	Clases, interfaces y *enum*
Niveles de protección	*Public, privete, internal, protected*	*Public, privete, internal, protected*
Elementos que se pueden definir dentro de una clase	Atributos, métodos, clases internas, propiedades, eventos	Atributos, métodos y clases internas
Polimorfismo	Es posible redefinir métodos, excepto cuando se usa la palabra *sealed*	Es posible redefinir métodos, excepto cuando se usa la palabra final
Encapsulamiento de atributos	Es posible acceder a los atributos a través de propiedades definidas mediante los métodos *set y get*	Se pueden implementar los métodos *set y get* para el acceso y escritura de los atributos
Paquetes y espacios de nombre	Se utilizan los espacios de nombre *namespaces*	Se utilizan los paquetes *packages*
Paso de parámetros a un método	Se pueden pasar parámetros por valor, referencia y número de parámetros variables	Se pueden pasar parámetros solo por valor

TABLA 1.10 Comparativa realizada entre JAVA y .NET.

1.7.2.1.4 VENTAJAS Y DESVENTAJAS DE CADA PLATAFORMA

En la tabla 1.11 se muestran algunas ventajas y desventajas de JAVA y de .NET.

	Ventajas	Desventajas
.NET	• Entorno de desarrollo altamente productivo. • Posibilidad de utilización de una amplia gama de lenguajes de programación. • C# es un lenguaje muy completo y con excelentes características	• Entorno de desarrollo no disponible en otras plataformas. • Implementaciones del .NET aun no son totalmente suficientes.
Java	• Tecnología altamente madura. • Muchos entornos de desarrollo alternativos. • Existe una implementación de la JRE para casi cualquier plataforma.	• Permite solo la utilización de java como lenguaje de programación. • No ofrece un entorno de desarrollo altamente productivo.

TABLA 1.11 Ventajas y desventajas de JAVA y .NET.

1.7.2.1.5 SELECCIÓN DE LA PLATAFORMA DE SOFTWARE.

Al finalizar la comparación realizada en la tabla 1.10 y visualizar las desventajas y ventajas en la tabla 1.11, se elije la plataforma JAVA para el diseño y desarrollo del sistema distribuido ya que cubra con las necesidades que se tienen además por las siguientes razones:

- Se puede implementar en cualquier sistema operativo.
- Existen varios entornos de desarrollo para crear aplicaciones con dicha plataforma.
- Cuenta con una infinidad de librerías que aumenta su potencia para el diseño de un sistema eficiente y completo.
- La conexión con los manejadores de base de datos es sencilla, ya que cuenta con un driver específico para cada manejador.
- Permite proteger el código al realizar un método llamado ofuscar.
- Posee una documentación amplia de su API, las nuevas librerías implementadas, métodos, etc.
- Es una fuente abierta, sin costo alguno.
- Independiente de la plataforma.
- Se utiliza para trabajar en ambiente de redes, que es una característica que se requiere para implementar un sistema distribuido.
- Es multiprocesos.

1.8 REQUERIMIENTOS DE HARDWARE

Cada aplicación que será utilizada tiene ciertas especificaciones mínimas de hardware para su correcto funcionamiento. Estas especificaciones (véase la tabla 1.12) son contempladas para sistemas operativos Windows.

	PostgreSQL	Java	TOTAL
Disco duro	90 MB	4 GB	4 GB
Memoria RAM	64 MB	128 MB	512 MB
Resolución de pantalla	800 x 600	800 x 600	800 x 600

TABLA 1.12 Requerimientos de hardware para las aplicaciones elegidas.

Los requerimientos anteriormente señalados tienen que ser contemplados en las maquinas que serán ocupadas para poner en funcionamiento el sistema.

Debido a que no se proporcionara ninguna computadora por parte de la Unidad Hospitalaria, los médicos prestarán sus computadoras para poder realizar la configuración correspondiente. Las características que presentan las computadoras que serán configuradas en el Departamento de Servicio de Anatomía Patológica son:

- ***Máquina 1 – Jefatura***

Características Generales

Nombre del sistema operativo: Microsoft Windows XP Professional Versión 5.1.2600 Service Pack 2 Compilación 2600
Fabricante del sistema operativo: Microsoft Corporation
Fabricante del sistema: Hewlett-Packard
Modelo del sistema: HP Pavilion dv1000 (PU959LA#ABM)

Memoria

Memoria física total: 2,048.00 MB
Memoria física disponible: 1.35 GB
Memoria virtual total: 2.00 GB
Memoria virtual disponible: 1.96 GB

Disco Duro

Modelo: FUJITSU MHT2060AT PL
Tamaño: 55.89 GB (60,011,642,880 Bytes)

- ***Máquina 2 – Oncopatólogo***

Características Generales

Nombre del sistema operativo: Microsoft Windows XP Professional Versión 5.1.2600 Service Pack 2 Compilación 2600
Fabricante del sistema operativo: Microsoft Corporation
Fabricante del sistema: TOSHIBA
Modelo del sistema: Satellite A15

Memoria

Memoria física total: 256,00 MB
Memoria física disponible: 83,49 MB
Memoria virtual total: 2.00 GB
Memoria virtual disponible: 1.96 GB

Disco Duro

Modelo: TOSHIBA MK3021GAS
Tamaño: 27,95 GB (30.005.821.440 Bytes)

- ***Máquina 3 – Citotecnólogo***

Características Generales

Nombre del sistema operativo: Microsoft Windows XP Professional Versión 5.1.2600 Service Pack 3 Compilación 2600
Fabricante del sistema operativo: Microsoft Corporation
Fabricante del sistema: TOSHIBA
Modelo del sistema: Satellite 1135

Memoria

Memoria física total: 512,00 MB
Memoria física disponible: 128.79 MB
Memoria virtual total: 2.00 GB
Memoria virtual disponible: 1.95 GB

Disco Duro

Modelo: IC25N040ATCS04-0
Tamaño: 37.26 GB (40,007,761,920 Bytes)

- ***Máquina 4 – Patologío***

Características Generales

Nombre del sistema operativo: Microsoft Windows 7 Starter
Versión 6.1.7600 compilación 7600
Fabricante del sistema operativo: Microsoft Corporation
Fabricante del sistema: TOSHIBA
Modelo del sistema: TOSHIBA NB305

Memoria

Memoria física total: 0,99 GB
Memoria física disponible: 327 MB
Memoria virtual total: 1,99 GB
Memoria virtual disponible: 1.18 GB

Disco Duro

Modelo: TOSHIBA MK2555GSX
Tamaño: 232,88 GB (250.056.737.280 bytes)

- ***Máquina 5 – Recepción y resultados***

Características Generales

Nombre del sistema operativo: Microsoft Windows XP Professional
Versión 5.1.2600 Service Pack 2
Fabricante del sistema operativo: Microsoft Corporation
Fabricante del sistema: Hewlett-Packard
Modelo del sistema: Armada

Memoria

Memoria física total: 512.00 MB
Memoria física disponible: 128.79 MB
Memoria virtual total: 2.00 GB
Memoria virtual disponible: 1.95 GB

Disco Duro

Modelo: FUJITSU MHT2060AT PL
Tamaño: 39.89 GB (60,011,642,880 Bytes)

CAPÍTULO 2

MARCO TEÓRICO

En tiempos de cambio, quienes estén abiertos al aprendizaje se adueñarán del futuro, mientras que aquellos que creen saberlo todo estarán bien equipados para un mundo que ya no existe.

Eric Hoffer

Para poder realizar el diseño y la implementación del sistema de base de datos distribuido es necesario tener antecedentes sobre base de datos distribuida, redes de datos, ingeniería de software, etc. Por esta razón, a continuación se explicará cada uno de estos temas.

2.1 MANEJADOR DE BASE DE DATOS

Es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. (Peter Rob, 2004). Sus funciones principales son:

- Crear y organizar una base de datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser consultados rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios.
- Registrar el uso de las bases de datos.
- Interacción con el manejador de archivos. Se da a través de las sentencias DML al comando del sistema de archivos, con ello el manejador de base de datos es el responsable del almacenamiento de los datos.
- Respaldo y recuperación a través de mecanismos implantados que permiten la recuperación de los datos en caso de ocurrir fallas en el sistema.
- Control de concurrencia, que consiste en controlar la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.
- Seguridad e integridad donde cuenta con mecanismos que permiten el control de la consistencia de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.

El manejador de base de datos actúa como interface entre la base de datos física y las peticiones del usuario. Interpreta las peticiones de entrada/salida del usuario y las manda al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria y la memoria principal. (Peter Rob, 2004). Lo anterior se representa en la figura 2.1.

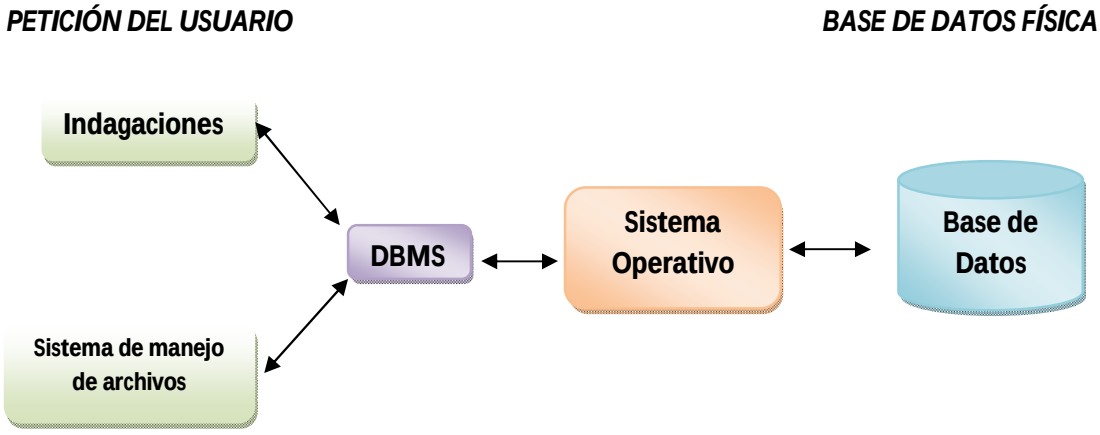


FIGURA 2.1 Funcionamiento de un manejador de base de datos.

2.2 BASES DE DATOS RELACIONALES

Las bases de datos están teniendo un impacto decisivo sobre el crecimiento de la tecnología computacional. Desempeñan un papel crucial en casi todas las áreas de aplicación de la computación, como los negocios, la ingeniería, la medicina, el derecho y educación, por mencionar algunas. El término base de datos es muy común por lo cual se debe de conocer su definición dada por diferentes autores.

- Una base de datos es un conjunto de datos relacionados entre sí. Por datos entendemos hechos conocidos que pueden registrarse y que tienen un significado implícito. (Ramez Elmasri, Shamkant B. Navathe, 2002, p.2).
- Una estructura en una computadora que aloja un conjunto de datos relacionados. Una base de datos contiene dos tipos de datos: datos para el usuario (hechos en bruto) y metadatos. (Peter Rob *et al*, 2006,p.809).
- La colección de datos, normalmente llamada base de datos, contiene información relevante para una empresa. (Silberschatz *et al*,2006,p.1).

2.2.1 PROPIEDADES DE LAS BASES DE DATOS

Una base de datos tiene las siguientes propiedades implícitas:

- Representar algún aspecto del mundo real, llamado universo de discurso. Las modificaciones que sufre dicho universo son reflejadas en la base de datos. (Peter Rob *et al*, 2006).
- Es un conjunto de datos lógicamente coherente, con cierto significado inherente. Una colección aleatoria de datos no puede considerarse propiamente una base de datos. (Peter Rob *et al*, 2006).
- Toda base de datos se diseña, construye y puebla con datos para un propósito específico. Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a dichos usuarios. (Peter Rob *et al*, 2006).

Una base de datos tiene una fuente de la cual se derivan los datos, cierto grado de interacción con los acontecimientos del mundo real que está activamente interesado en el contenido de está.

2.2.2 CARACTERÍSTICAS DE UNA BASE DE DATOS

Entre las principales características de las bases de datos se puede mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.

- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

2.2.3 NIVELES DE ABSTRACCIÓN EN UNA BASE DE DATOS

Se observa que en los sistemas existen dos estructuras distintas, la lógica (vista del usuario) y la física (manera en la que se encuentran los datos almacenados). En las bases de datos se presenta un nuevo nivel de abstracción que se le ha llamado nivel conceptual o estructura lógica global. Este nivel intermedio pretende realizar una representación global de los datos que se utilicen en los niveles físico y lógico, y que sea independiente tanto del equipo como de cada usuario. (Ramez Elmasri, 2002).

Las bases de datos tienen una arquitectura *American National Standard Institute – Standards Planning and Requirements Committee* (ANSI/SPARC). ANSI/SPARC es un grupo de normalización creado en 1969, quien en 1975 propuso el uso de tres niveles de descripción de datos:

- Nivel interno o físico. Se refiere al almacenamiento físico, en él se describe cómo se almacenan realmente los datos en memoria secundaria, en qué archivos, su nombre y dirección. También estarán los registros, longitud, campos, índices y las rutas de acceso a esos archivos. (Peter Rob *et al*, 2006).

- Nivel conceptual. En él se describen cuáles son los datos almacenados en la base de datos y que relaciones existen entre ellas. Este nivel lo definen los administradores de la base de datos, quienes deciden que información se guarda. El nivel conceptual corresponde a la estructura organizacional de los datos obtenida al reunir los requerimientos de todos los usuarios, sin preocuparse de su organización física ni las vías de acceso. (Peter Rob *et al*, 2006). Este nivel puede contener:

- Nivel externo o vistas. Es el nivel más cercano al usuario y representa la perspectiva individual de cada usuario. Describe únicamente la parte de datos para un usuario o grupo de usuarios. Habrá usuarios que podrán acceder a más de un esquema externo y uno de éstos puede ser compartido por varios usuarios, se protege así el acceso a los datos por parte de personas no autorizadas. (Ramez Elmasri, 2002). Cuando se construye un esquema externo:

    - Se puede omitir una o más entidades del sistema.
    - Se puede omitir una o más atributos de una entidad.
    - Se puede omitir una o más relaciones entre los datos.
    - Se puede cambiar el orden de los atributos.

Para una base de datos solo habrá un único esquema interno, un único esquema conceptual, pero puede haber varios esquemas externos. En la figura 2.2 se muestran los niveles de abstracción anteriormente descritos.

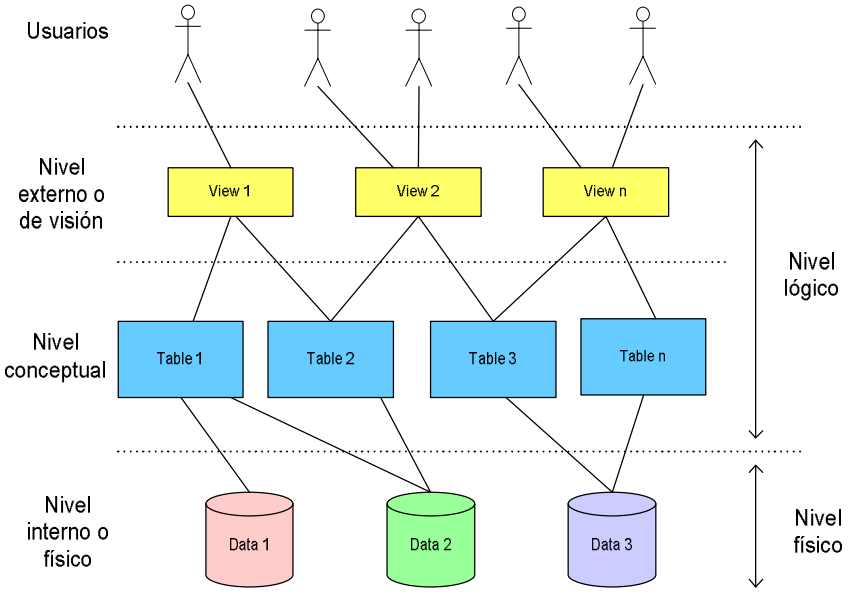


FIGURA 2.2 Niveles de Abstracción.

2.2.4 VENTAJAS DE LAS BASES DE DATOS

Las ventajas de las bases de datos con respecto a los sistemas de archivos son entre otras Mercedes Marqués. (2002):

- Control de la redundancia de los datos: Los sistemas de archivos almacenan varias copias de los mismos datos en archivos distintos, esto ocasiona que se desperdicie espacio de almacenamiento y se presente la inconsistencia en los datos. Cuando se trata de las bases de datos toda la información está integrada, lo cual disminuye en gran cantidad la inconsistencia de los datos que se presenta en los archivos.

- Consistencia de datos: Al controlar la redundancia de los datos se disminuye el riesgo de que se presenten inconsistencias. Si un dato está almacenado una sola vez, cualquier operación realizada sobre él se debe ejecutar sólo una vez, y debe estar disponible para todos los usuarios inmediatamente. En el caso de que un dato este duplicado, el sistema al conocer esta redundancia garantiza que todas las copias de este dato se mantengan consistentes.

- Mantenimiento de estándares: En el caso de los archivos no existen estándares documentados que se deban seguir y sean fáciles de implementar por las empresas o

negocios, en cuanto a las bases de datos es más fácil poner en práctica dichos estándares establecidos por la misma empresa o negocio como los definidos por organizaciones.

- Integridad de datos: En cuanto a integridad se entiende como reglas o restricciones que no se pueden violar. Dichas reglas se pueden aplicar a los datos, a las relaciones que existen entre los datos, y es el SGBD quien se debe encargar de mantenerlas.

- Seguridad: Cuando se tienen sistemas de archivos no se puede controlar el acceso a la información de los usuarios no autorizados. Mientras que en las bases de datos se puede ingresar a los datos si y solo si es usuario autorizado, con esto la integración de los datos es confiable.

- Accesibilidad a los datos: Los manejadores de bases de datos proporcionan lenguajes de consultas que permiten al usuario realizar cualquier tipo de consulta sobre los datos.

- Productividad: En las bases de datos se proporcionan muchas funciones estándar que el DBA necesita escribir en un sistema de archivos. El hecho de disponer de estas funciones permite al DBA centrarse en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel.

- Mantenimiento: En los sistemas de archivos, las descripciones de los datos se encuentran introducidas en los programas de aplicación que los manejan. Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados. Sin embargo, en las bases de datos se separan las descripciones de los datos de las aplicaciones. Esto es lo que se conoce como independencia de datos, gracias a la cual se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

- Aumento de la concurrencia: En algunas ocasiones varios usuarios pueden acceder simultáneamente a un mismo sistema de archivos, esto provoca que se pierda información. En cuanto a las bases de datos se gestiona el acceso concurrente y se garantiza que no ocurra pérdida de información.

- Servicios de copias de seguridad: Muchos sistemas de archivos dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos. Sin embargo, las bases de datos funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

2.2.5 DESVENTAJAS DE LAS BASES DE DATOS

Así como las bases de datos presentan ventajas también se deben considerar las desventajas de dichas.

- Complejidad: Los SGBD son conjuntos de programas que pueden llegar a ser complejos con una gran funcionalidad. Es preciso comprender muy bien esta funcionalidad para poder realizar un buen uso de ellos. (Silberschatz, 2007).
- Coste del equipamiento adicional: Tanto el SGBD, como la propia base de datos, pueden hacer que sea necesario adquirir más espacio de almacenamiento. Además, para alcanzar las prestaciones deseadas, es posible que sea necesario adquirir una máquina más grande o una máquina que se dedique solamente al SGBD. Todo esto hará que la implantación de un sistema de bases de datos sea más cara. (Silberschatz, 2007).
- Vulnerable a los fallos: El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse. Es por ello que deben tenerse copias de seguridad (*Backup*). (Silberschatz, 2007).

2.3 BASES DE DATOS DISTRIBUIDAS

Las bases de datos distribuidas son una colección de datos que corresponde a un sistema, pero a su vez está repartida entre diferentes nodos de una red de datos, encontrándose a nivel local y geográfico, cada uno de estos nodos es autónomo en sus capacidades de procesamiento y capaz de realizar operaciones locales y globales que permitan la consulta de todos los datos como si se tratase de uno solo. (Peter Rob, 2004). En la figura 2.3 se ejemplifica el concepto de base de datos distribuida.

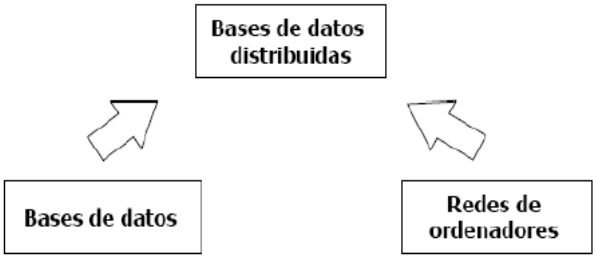


FIGURA 2.3 Representación del concepto de base de datos distribuida.

2.3.1 TIPOS DE BASES DE DATOS DISTRIBUIDAS

Las bases de datos distribuidas se dividen en dos tipos:

- Bases de datos distribuidas homogéneas todos los nodos tienen idéntico software de sistemas gestores de bases de datos, son conscientes de la existencia de los demás nodos y acuerdan cooperar en el procesamiento de las solicitudes de los usuarios. (Peter Rob, 2004).

- Bases de datos distribuidas heterogéneas los nodos pueden utilizar esquemas y software de gestión de sistemas de bases de datos diferentes. Puede que unos nodos no sean conscientes de la existencia de los demás y puede que sólo proporcionen facilidades limitadas para la cooperación en el procesamiento de las transacciones. La diferencia entre los esquemas puede constituir un problema importante para el procesamiento de las consultas, mientras que la discrepancia de software supone un inconveniente para el procesamiento de transacciones que tengan acceso a varios nodos. (Peter Rob, 2004).

2.3.2 VENTAJAS DE LAS BASES DE DATOS DISTRIBUIDAS

Las características más sobre salientes de las bases de datos distribuidas con respecto a las bases de datos centralizadas son (Peter Rob, 2004):

- Los datos se localizan en diferentes puntos de acuerdo a las necesidades de negocio que tenga el usuario.
- Existirá un acceso más rápido a los datos, dado que los usuarios únicamente utilizan un subconjunto de datos. Este subconjunto se guardará localmente, por lo tanto la base de datos permitirá el uso de los datos más rápido ya que no se encuentran remotamente localizados como en las bases de datos centralizadas.
- Al estar distribuidos todos los datos en diferentes nodos de una red de datos, se podrá realizar el procesamiento de los mismos en varios nodos, así la carga de trabajo que se tenga, se repartirá en toda la red.
- La red de datos tiene la posibilidad de crecer de manera fácil y rápida, sin afectar las operaciones en los diferentes nodos.
- Hay una comunicación más rápida, esto se deba a que la información se divide y se almacena más cerca del usuario local.
- Como los sitios de almacenamiento local son pequeños y están más cerca del usuario, proveen una mejor comunicación entre los diferentes sitios, esto ayudará a mejorar el funcionamiento del sistema gestor de la base de datos.
- Los costos de operación serán reducidos porque sale más barato aumentar nodos en la red de datos que actualizar un sistema.
- La base de datos distribuida es capaz de trasladar las operaciones cuando unos de los nodos existentes falle.
- Los usuarios pueden acceder a cualquier subconjunto de datos disponible, y la solicitud es procesada en cualquier procesador. Las solicitudes no dependen de un procesador específico, cualquiera de éstos es capaz de manejar dicha solicitud.

2.3.3 DESVENTAJAS DE LAS BASES DE DATOS DISTRIBUIDAS

Las bases de datos distribuidas también presentan ciertas desventajas que se indican a continuación (Silberschatz, 2007):

- Hay una menor seguridad en cuanto al control de acceso a los datos: control de replica y errores que puedan producirse en la red.
- Mayor complejidad en el diseño e implementación del sistema. Además si la replicación de datos no se hace de forma adecuada, las ventajas se pueden transformar en desventajas.
- Falta de estándares y de experiencia.
- No se puede garantizar al 100% el rendimiento y la fiabilidad del funcionamiento.
- Las bases de datos distribuidas guardan fragmentos de datos en distintos lugares, por lo que se requiere contar con un mayor espacio de almacenamiento en disco.

2.3.4 COMPONENTES DE LAS BASES DE DATOS DISTRIBUIDAS

Las bases de datos distribuidas se forman principalmente de los siguientes componentes (M. Tamer Özsu, Patrick Valduriez, 1991):

- Estaciones de trabajo: nodos que forman la red de datos.
- Procesador de datos locales: se encarga de la gestión local de los datos de forma parecida al software de una base de datos centralizada, por lo que además de ejecutar transacciones locales se encarga de la concurrencia y la recuperación ante fallos a nivel local.
- Diccionario global: se guarda información acerca de dónde y cómo se almacenan los datos, el modo de acceso y características físicas. Contiene las especificaciones necesarias para pasar de la representación externa de los datos a la representación interna de los mismos.
- Procesador de aplicaciones distribuidas: es el responsable de las funciones distribuidas. Accede a la información sobre la ubicación de los datos, que se encuentra en el diccionario, y se ocupa de procesar todas las peticiones que involucran más de una sede para generar un plan de ejecución distribuido.
- Componentes de software y hardware: los componentes de red permiten que todos los nodos interactúen e intercambien datos. Como es probable que el software y hardware sea surtido por diferentes proveedores, es deseable garantizar que las funciones de la base de datos distribuida puedan ejecutarse en múltiples plataformas.
- Medio de comunicación: el que trasportará los datos de una estación de trabajo a otra.

2.3.5 ARQUITECTURA DE LAS BASES DE DATOS DISTRIBUIDAS

La arquitectura de las bases de datos distribuidas está basada en la arquitectura ANSI/SPARC (M. Tamer Özsu, 1991) mostrada en la figura 2.4:

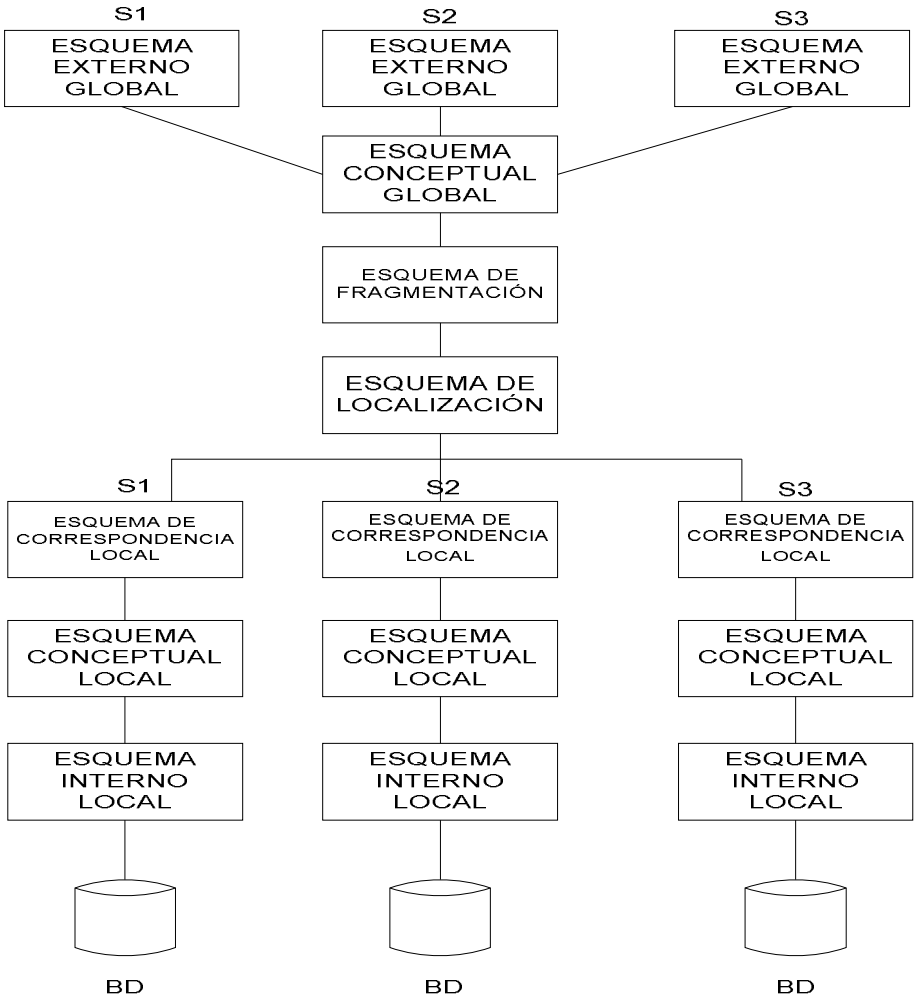


FIGURA 2.4 Arquitectura de las bases de datos distribuidas.

- Esquema conceptual global: Es la descripción lógica de la base de datos completa, como si no estuviera distribuida. Este nivel corresponde al nivel conceptual de la arquitectura ANSI/SPARC y contiene definiciones de entidades, relaciones, constantes e información sobre seguridad e integridad. Proporciona independencia de datos físicos desde el entorno distribuido. Los esquemas externos globales proporcionan independencia de datos lógica.

- Esquemas de fragmentación y localización: El esquema de fragmentación es una descripción de cómo los datos están particionados lógicamente. El esquema de localización es una descripción de dónde están localizados los datos y tiene en cuenta cualquier replicación.

- Esquemas locales: Cada sistema de gestión de base de datos local tiene su propio conjunto de esquemas. Los esquemas conceptual e interno locales corresponden a los equivalentes de la arquitectura ANSI/SPARC.

2.3.6 TRANSPARENCIA

Las bases de datos distribuidas deben proporcionar transparencia para permitir que el usuario sienta que es el único que esta utilizando la base de datos. Existen diferentes tipos de transparencia que se mencionan a continuación:

- Transparencia de distribución: la cual permite que una transacción actualice datos en varios nodos de la red. Maneja la base de datos físicamente dispersa como si fuera centralizada. (Peter Rob, 2004). Se conocen tres niveles de transparencia distribuida:
    - Transparencia de fragmentación: no se requiere saber que la base de datos está fragmentada, esto es, no saber ni el nombre ni la ubicación del fragmento.
    - Transparencia de ubicación: específica el nombre del fragmento pero no su ubicación.
    - Transparencia de ubicación local: se conoce el nombre y la ubicación del fragmento.
- Transparencia de transacción: Permite que se actualicen los datos en varios nodos de la red, garantizando que la transacción se complete o aborte según sea el caso. (Peter Rob, 2004).
- Transparencia de falla: Permite que el sistema siga funcionando en caso de que algún nodo de la red falle, y las operaciones que esté realizando el nodo que falló serán traspasadas a otro nodo. (Peter Rob, 2004).
- Transparencia de desempeño: Permite que el sistema funcione como un sistema centralizado. Se encarga de encontrar la ruta de acceso más barata a los datos remotos. (Peter Rob, 2004).

2.3.7 FRAGMENTACIÓN

La fragmentación se utiliza para dividir un objeto de la base de datos en diferentes segmentos de datos. Estos fragmentos pueden alojarse en diferentes nodos de la red de datos. Los datos fragmentados se guardan en el catálogo de datos distribuidos, desde donde se accede a través del procesador para dar una solución al usuario. (Peter Rob, 2004).

La fragmentación es necesaria debido a tres aspectos:

- Encontrar la unidad de relación apropiada; ya que el acceso a la aplicación no se hace en la totalidad de las relaciones de la base pero si en un subconjunto de ellas. Un ejemplo es cuando una sucursal de una empresa localizada en Pachuca únicamente ocupa los datos de las ventas realizadas en dicha entidad.

- Si se tiene sub-relaciones como unidades de relación, cuando se trata de acceder a los datos que están en otros sitios, será más conveniente hacer la replicación de los datos que están en las sub-relaciones en lugar de traer los datos de todas las relaciones. Un ejemplo es cuando una sucursal de una empresa controla dos sucursales encontradas en otras entidades, para no copiar toda la información de dichas sucursales únicamente replica ciertos datos que necesitará.

- Si se toman las sub-relaciones como unidades de relación; se puede ejecutar una cantidad de transacciones concurrentemente.

Al realizar el proceso de fragmentación debe cumplir con las siguientes tres reglas, las cuales evitan cambios semánticos en la base de datos. (M. Tamer Özsu, 1991).

- Plenitud: Si una relación R se descompone en una serie de fragmentos R1, R2,…,Rn, cada elemento de datos que pueda encontrarse en R deberá poder encontrarse en uno o varios fragmentos Ri. Esta propiedad es extremadamente importante porque asegura que los datos de la relación global se proyecten sobre los fragmentos sin pérdida de datos.

- Reconstrucción: Debe ser posible definir una operación relacional que permita reconstruir la relación R a partir de los fragmentos. Esta regla asegura que se preserven las dependencias funcionales.

- Disyunción: Esta regla asegura que los fragmentos horizontales sean disjuntos. La fragmentación vertical es una excepción a esta regla, donde las claves primarias deben repetirse para permitir la reconstrucción. Esta regla asegura la mínima redundancia de los datos.

2.3.7.1 GRADO DE FRAGMENTACIÓN

Cuando se va a fragmentar una base de datos se debe determinar el grado de fragmentación que va a alcanzar, ya que éste es un factor que influirá notablemente en el desarrollo de la ejecución de las consultas. El grado de fragmentación puede variar desde una ausencia de particionamiento, considerando a las relaciones unidades de fragmentación; o bien, fragmentar a un grado en el cada tupla o atributo forme un fragmento. Ante estos dos casos extremos, evidentemente se ha de buscar un compromiso intermedio, el cual se deberá establecer sobre las características de las aplicaciones que hacen uso de la base de datos. Dichas características se podrán formalizar en una serie de parámetros. De acuerdo con sus valores, se podrá establecer el grado de fragmentación de los de datos. (Silberschatz, 2007).

2.3.7.2 TIPOS DE FRAGMENTACIÓN

En la figura 2.5 se muestran los tipos de fragmentación, los cuales se describen detalladamente en las siguientes secciones:

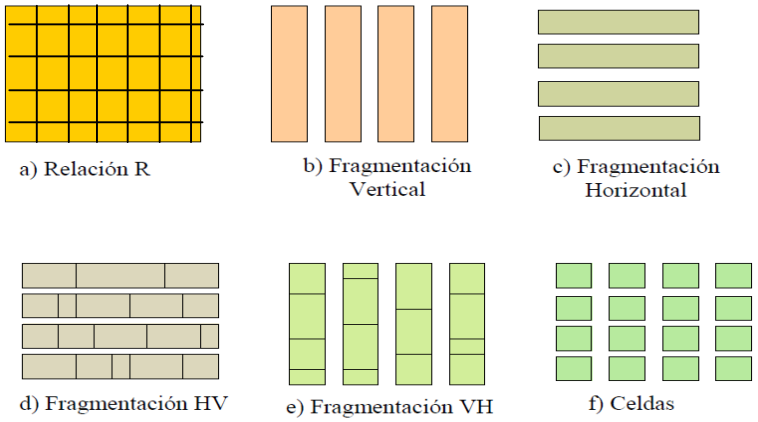


FIGURA 2.5 Tipos de fragmentación.

2.3.7.2.1 FRAGMENTACIÓN HORIZONTAL

La fragmentación horizontal se realiza sobre los registros de la relación. Cada fragmento será un subconjunto de los registros de la relación. Existen dos variantes de la fragmentación horizontal: la primaria y la derivada.

La fragmentación horizontal requiere tener la información sobre la base de datos, que implica el esquema conceptual global donde es importante señalar cómo las relaciones de la base de datos se conectan con otras. Así como la Información sobre la aplicación, donde se necesita tanto información cualitativa como cuantitativa. La información cualitativa guiará la fragmentación, mientras que la cuantitativa se necesitará en los modelos de asignación. La principal información de carácter cualitativo son los predicados empleados en las consultas de usuario. A parte de los predicados simples, las consultas emplean predicados más complejos resultado de combinaciones lógicas de los simples. Una combinación especialmente interesante es la conjunción de predicados simples, al predicado resultante se le denomina predicado mintérmino. Sobre la información cuantitativa relativa a las aplicaciones, se necesita definir dos conjuntos de datos. (M. Tamer Özsu, 1991).

- Selectividad mintérmino. Es el número de registros de una relación a los que accede una consulta de acuerdo a un predicado mintérmino dado.
- Frecuencia de acceso. Es la frecuencia con la que un usuario accede a los datos.

2.3.7.2.1.1 FRAGMENTACIÓN HORIZONTAL PRIMARIA

La fragmentación horizontal primaria se define como una operación de selección de las relaciones pertenecientes al esquema de la base de datos.

Un fragmento horizontal Ri de una relación R contiene todos los registros de R que satisfacen un predicado mintérmino mi. Dado un conjunto de predicados mintérmino M, existen tantos fragmentos horizontales de la relación R como predicados mintérminos. (M. Tamer Özsu, 1991).

El primer paso para el algoritmo de fragmentación consiste en establecer un conjunto de predicados con ciertas propiedades. El segundo paso en el proceso de fragmentación primaria consiste en derivar el conjunto de predicados mintérmino que pueden definirse sobre un conjunto de predicados simples. Estos predicados mintérmino establecen los fragmentos candidatos para el proceso de asignación. El establecimiento de los predicados mintérmino es trivial; la dificultad radica en el tamaño del conjunto de predicados mintérmino, que puede ser muy grande. En el paso siguiente se presentarán formas de reducir el número de predicados mintérmino necesarios para la fragmentación. El tercer paso aborda la eliminación de algunos fragmentos mintérmino que puedan ser redundantes. Esta eliminación se desarrolla identificando aquellos mintérminos que puedan resultar contradictorios sobre un conjunto de implicaciones. (M. Tamer Özsu, 1991).

A continuación se presenta el pseudocódigo del algoritmo de fragmentación horizontal primaria descrito en el párrafo anterior.

Entrada:
R_i: relación
Pr_i: conjunto de predicados simples
Salida:
M_i: conjunto de fragmentos mintérmino
Inicio
Pr_i <- COM_MIN(R_i , Pr_i)
Determinar el conjunto M_i de predicados mintérmino;
Determinar el conjunto I_i de implicaciones de $p_i \epsilon Pr'_i$;
Para cada $m_i \epsilon M_i$ hacer
Si m_i es contradictorio respecto a I
Entonces
M_i <- M_i - m_i
Fin de si
Fin de para
FIN

2.3.7.2.1.2 FRAGMENTACIÓN HORIZONTAL DERIVADA

La fragmentación horizontal derivada no puede basarse en una propiedad de sus propios atributos, sino que se deriva de la fragmentación horizontal de otra relación. Por lo anterior se debe prestar atención al enlace entre las relaciones propietaria y miembro, el cual se define como un equi-join. Entendiendo que un equi-join puede desarrollarse a través de semi - join. Por tanto, dado un enlace *L* donde *propietaria* (*L*) = *S* y *miembro* (*L*) = *R*, los fragmentos horizontales derivados de *R* se definen como (M. Tamer Özsu, 1991):

$$R_j = R \ltimes S_i, 1 \leq i \leq w$$

donde *w* es el número máximo de fragmentos que se definirán sobre *R*, y *Si* = *Fi*(*S*), donde *Fi* es la fórmula según la cual se define el fragmento horizontal primario *Si*.

Las tres entradas necesarias para desarrollar la fragmentación horizontal derivada son las siguientes (M. Tamer Özsu, 1991):

- El conjunto de particiones de la relación propietaria
- La relación miembro
- El conjunto se predicados resultados de aplicar el semi-join entre la propietaria y la miembro.

En un esquema de base de datos, resulta frecuente que existan más de dos enlaces sobre una relación R. En este caso, aparece más de una posibilidad de fragmentación horizontal derivada. (M. Tamer Özsu, 1991).

2.3.7.2.2 FRAGMENTACIÓN VERTICAL

La fragmentación vertical de una relación R produce una serie de fragmentos R1, R2, ..., Rr, cada uno de los cuales contiene un subconjunto de los atributos de R así como la clave primaria de R. El objetivo de la fragmentación vertical consiste en dividir la relación en un conjunto de relaciones más pequeñas tal que algunas de las aplicaciones de usuario sólo hagan uso de un fragmento. (M. Tamer Özsu, 1991).

Una fragmentación óptima es aquella que produce un esquema de división que minimiza el tiempo de ejecución de las aplicaciones que emplean esos fragmentos. La partición vertical resulta más complicada que la horizontal. Esto se debe al aumento del número total de alternativas que tenemos disponibles. (M. Tamer Özsu, 1991).

Existen dos enfoques heurísticos para la fragmentación vertical de relaciones:

- Agrupación. Comienza asignando cada atributo a un fragmento, y en cada paso, junta algunos de los fragmentos hasta que satisface un determinado criterio.

- Escisión. A partir de la relación se decide que fragmentos resultan mejores, basándose en las características de acceso de las aplicaciones a los atributos.

2.3.7.2.3 FRAGMENTACIÓN MIXTA

En muchos casos la fragmentación vertical u horizontal del esquema de la base de datos no será suficiente para satisfacer los requisitos de las aplicaciones. Cuando al proceso de fragmentación vertical le sigue una horizontal, es decir, se fragmentan horizontalmente los fragmentos verticales resultantes, se habla de la fragmentación mixta HV. En el caso contrario, estaremos ante una fragmentación VH. Una característica común a ambas es la generación de árboles que representan la estructura de fragmentación como se observa en la figura 2.6. (M. Tamer Özsu, 1991).

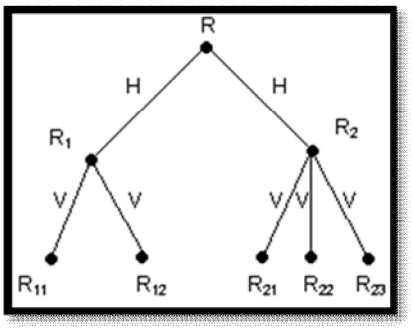


FIGURA 2.6 Ejemplo del árbol de la estructura de la fragmentación mixta.

También debe tenerse en cuenta el número de niveles que el árbol forme, es decir, nada impide que tras realizar una fragmentación VH, se pueda aplicar a los fragmentos resultantes una nueva fragmentación vertical, y a estos últimos una nueva fragmentación horizontal, etc. Dicho número puede ser grande, pero también será ciertamente finito. En el caso horizontal, el nivel máximo de profundidad se alcanzará cuando cada fragmento albergue un único registro, mientras que en el caso vertical el final llegará cuando cada fragmento contenga un único atributo. (M. Tamer Özsu, 1991).

La técnica de fragmentación mixta basada en celdas, genera celdas de rejilla, definiéndolas como un fragmento horizontal y vertical simultáneo. La técnica aplica un algoritmo de fragmentación vertical y otro horizontal de manera concurrente sobre la relación. Los algoritmos realizan una fragmentación máxima, es decir, se persigue que en cada celda únicamente haya un atributo y un registro. Una vez generadas las celdas se aplica un método para optimizar la rejilla mediante fusión o desfragmentación, de acuerdo, a las aplicaciones que actúen sobre esos fragmentos. El método, por tanto, persigue una fragmentación lo más específicamente posible acorde con las aplicaciones y los sitios existentes en la red. (M. Tamer Özsu, 1991).

2.3.8 REPLICACIÓN

La replicación de datos se refiere al almacenamiento de copias de datos en sitios múltiples existentes en la red de datos. El sistema conserva varias réplicas idénticas de la relación. Cada réplica se guarda en una localización diferente. La información sobre la replicación de los datos se guarda en el catálogo de datos distribuidos (DDC), cuyo contenido es utilizado por el procesador de transacciones para decidir que copia de un fragmento acceder. (Peter Rob, 2004).

Los datos replicados cumplen con la regla de consistencia mutua, que indica que todas las copias de fragmentos sean idénticas, para garantizar la consistencia de los datos se debe realizar la actualización de la base de datos en todos los nodos que componen la red de datos. (Peter Rob, 2004).

Es importante considerar los factores que influyen en la decisión de utilizar la replicación:

- Frecuencia de uso de los datos (es recomendable replicar los datos que se utilizan para sólo lectura).
- Tamaño de la base de datos.
- Costo que es asociado con la sincronización de las transacciones.

2.3.8.1 TIPOS DE REPLICACIÓN

Existen tres tipos de replicación (Silberschatz, 2007):

- Replicación total: Es aquella que guarda varias copias de cada fragmento de la base de datos en varios nodos. No es muy recomendada por que aumenta considerablemente la carga al sistema.
- Replicación parcial: En ella se guardar múltiples copias de algunos fragmentos de la base de datos en múltiples nodos.
- No replicación: En este caso se guarda cada fragmento de la base de datos en un solo nodo.

2.3.8.2 VENTAJAS Y DESVENTAJAS DE LA REPLICACIÓN

Al realizar la replicación siempre se debe tener en cuenta las ventajas e inconvenientes que ésta nos puede proporcionar en la implementación del sistema distribuido (Peter Rob, 2004):

- Disponibilidad: Si falla uno de los nodos que contiene la relación R, ésta aún se podrá encontrar en otro nodo. Por tanto, el sistema puede seguir procesando las consultas que impliquen a R, a pesar del fallo en un nodo.
- Aumento del paralelismo: Los accesos de lectura a la relación R se pueden procesar en paralelo en los nodos donde está replicada R. Por tanto, la replicación minimiza el tráfico de datos entre los nodos.
- Aumento de la sobrecarga en las actualizaciones: El sistema debe asegurarse que todas las réplicas de la relación R estén consistentes, en caso contrario, dará lugar a lecturas sucias y resultados erróneos.

2.3.9 COLOCACIÓN DE LOS DATOS

El proceso de ubicación de los datos se puede llevar a cabo utilizando las siguientes estrategias:

- Colocación centralizada de los datos: En un solo nodo se guarda toda la base de datos. En la figura 2.7 se muestra un ejemplo.

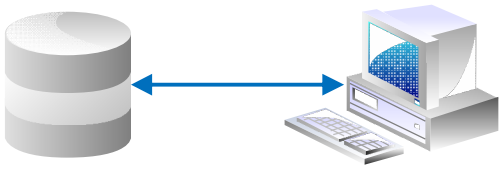


FIGURA 2.7 Ejemplo de colocación centralizada de los datos.

- Colocación particionada de los datos: La base de datos se divide en varias partes separadas y se guardan en varios nodos (véase la figura 2.8).

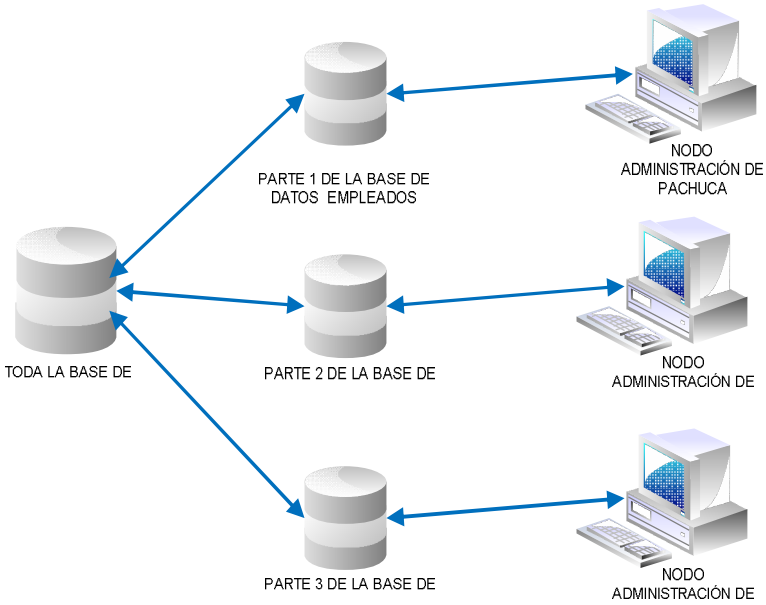


FIGURA 2.8 Ejemplo de colocación particionada de los datos.

- Colocación replicada de los datos: se guardan copias de uno o más fragmentos de la base de datos en varios nodos de la red. En la figura 2.9 se muestra un ejemplo de la colocación replicada de los datos.

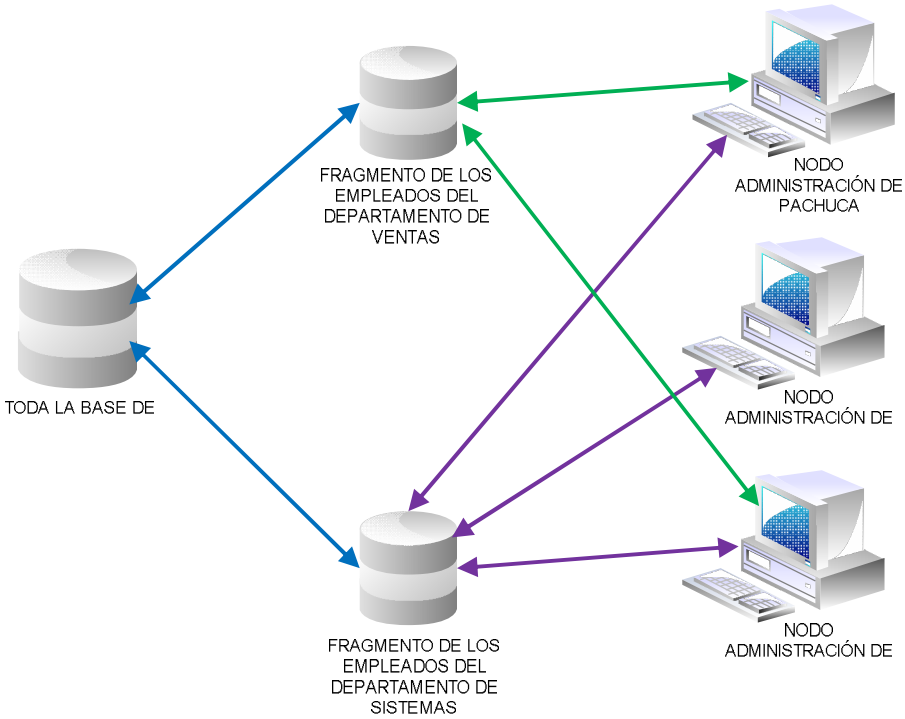


FIGURA 2.9 Ejemplo de colocación replicada de los datos.

2.4 REDES DE DATOS INALÁMBRICAS

Una red inalámbrica es un sistema de comunicación de datos que proporciona conexión móvil entre equipos situados dentro de la misma área interior o exterior. En lugar de utilizar el par trenzado, el cable coaxial o la fibra óptica, utilizado en las redes LAN convencionales, las redes inalámbricas transmiten y reciben datos a través de ondas electromagnéticas. (Andrew S. Tanenbaum, 2003).

En una red inalámbrica cada nodo dispone de un adaptador de red inalámbrico. Estos adaptadores se conectan enviando y recibiendo ondas de radio a través de un transceptor (transmisor-receptor), que puede situarse en cualquier lugar dentro del área de cobertura, sin la preocupación del cableado. Las redes inalámbricas permiten la transmisión de datos a velocidades de 11 Mbps o incluso superiores. Un ejemplo de una red de datos inalámbrica se muestra en la figura 2.10.

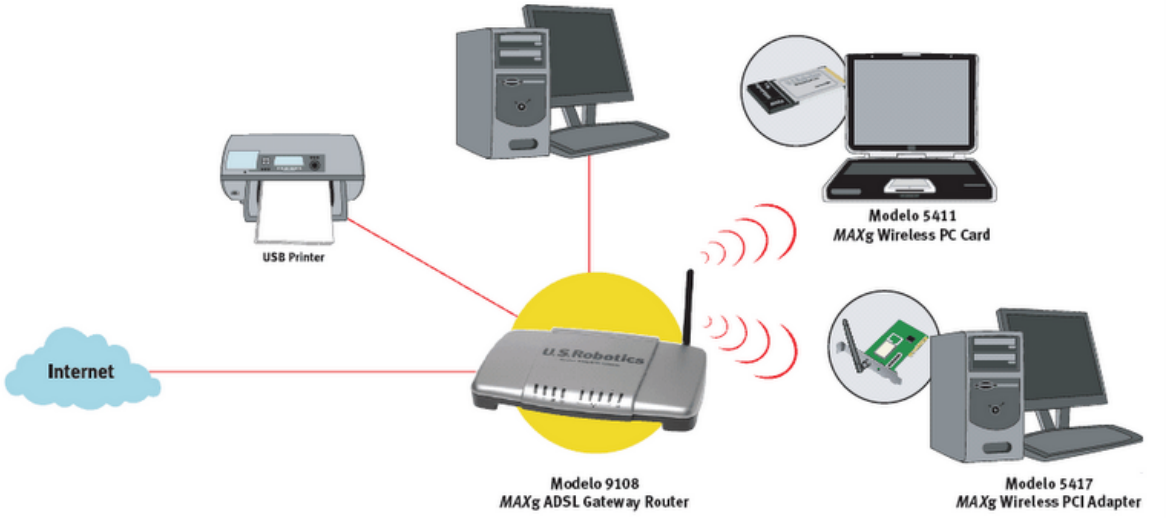


FIGURA 2.10 Ejemplo de una red de datos inalámbrica.

2.4.1 VENTAJAS Y DESVENTAJAS

Las principales ventajas que ofrecen las redes inalámbricas con respecto a las redes alambricas son (Heikki Koivo, 2009):

- Movilidad: las redes inalámbricas ofrecen acceso a la red local desde cualquier sitio dentro de su cobertura, manteniendo el ancho de banda y sin perder conexión incluso encontrándose en movimiento.
- Rápida instalación: más simplicidad, dado que no se requiere la instalación de cables.
- Flexibilidad: es posible disponer de acceso a una red en entornos donde es complicada la instalación de cableado.

- Facilidad: es posible instalar nuevas redes inalámbricas o cambiar de lugar las existentes, de forma rápida y sencilla.
- Adaptabilidad: permite frecuentes cambios de la topología de la red y facilita su escalabilidad.
- Facilita el aumento de nuevos usuarios a la red: sin la necesidad de extender un cable al nuevo nodo de trabajo perteneciente al usuario.
- Permite organizar redes en sitios cambiantes o situaciones no estables como pudieran ser lugares de emergencia, congresos, sedes temporales, etc.

Los principales inconvenientes de las redes inalámbricas son:

- Menor velocidad: Las redes de cable trabajan de 100 a 10,000 Mbps, mientras que las redes inalámbricas trabajan de 11 a 108 Mbps.
- Mayor inversión inicial: Para la mayoría de las configuraciones de la red local, el costo de los equipos de red inalámbricos es superior al de los equipos de red cableada.
- Seguridad: Las redes inalámbricas tienen la particularidad de no necesitar un medio físico para funcionar. Esto fundamentalmente es una ventaja, pero se convierte en una desventaja cuando cualquier computadora que está dentro del área de cobertura de la red puede lograr accesar a otra computadora.
- Interferencias: Las redes inalámbricas funcionan utilizando el medio radio electrónico en la banda de 2.4 GHz. Esta banda de frecuencia no requiere de licencia administrativa para ser utilizada por lo que muchos equipos electrónicos utilizan la misma banda de frecuencia. Este hecho hace que no se tenga la garantía de que el entorno radio eléctrico este completamente limpio para que la red inalámbrica funcione a su máximo nivel de rendimiento.
- Alcance: el alcance de una red inalámbrica está determinado por la potencia de los equipos y la garantía de las antenas, si estos no tienen una buena calidad no habrá una buena cobertura.

2.4.2 COMPONENTES

Existen fundamentalmente dos tipos de dispositivos básicos inalámbricos (Tutorial de Redes WIFI, n.d.):

Unidad base inalámbrica

También se conocen como punto de acceso o pasarela inalámbrica (véase la figura 2.11). Su funcionalidad básica consiste en:

- Realizar la conversión de la señal de datos Ethernet a señales de radio, pudiendo ser un punto de conexión entre ambas redes.
- Actúa como elemento de interconexión entre diferentes clientes inalámbricos.

- Proporciona un área de cobertura para los clientes inalámbricos. El espacio cubierto dependerá de la capacidad del equipo y sobre todo del entorno físico que se quiera cubrir, espacios exteriores o interiores.
- Pueden ofrecer funciones de *firewall* que permite aumentar la seguridad de la red.
- Pueden ofrecer mecanismos de autenticación para los clientes inalámbricos.
- Pueden ser configurados para crear diferentes escenarios de trabajo.

FIGURA 2.11 Ejemplo de una unidad base inalámbrica.

Clientes inalámbricos

Son adaptadores inalámbricos que convierten las señales de datos Ethernet a señales de radio y permiten a un equipo acceder a la red inalámbrica. Los sistemas operativos los tratan como adaptadores de red, análogos a las tarjetas Ethernet, por lo que desde el punto de vista del usuario final no existe diferencia entre disponer de uno u otro adaptador, ni de estar conectado a una u otra red. Una terminal equipada con un cliente inalámbrico y situada dentro del área de cobertura de una unidad base, puede comunicarse con los demás dispositivos de la misma red local sin necesidad de cables. En la figura 2.12 se muestran algunos ejemplos.

Tipos de clientes inalámbricos:

- **Adaptador USB inalámbrico**: Se conecta al puerto USB del ordenador o dispositivo.
- Tarjeta PCMCIA inalámbrica: Se instala en una ranura PCMCIA de un PC portátil.

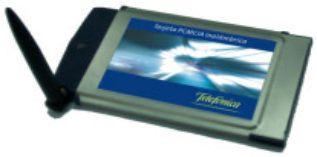


FIGURA 2.12 Ejemplos de clientes inalámbricos.

2.4.3 ARQUITECTURA Y PROTOCOLOS DE LAS REDES INALÁMBRICAS

La arquitectura de las redes inalámbricas se basa en la capa física que proporciona una serie de servicios a la capa de acceso al medio (MAC). Los diferentes métodos de acceso están diseñados según el modelo OSI y se encuentran ubicados en el nivel físico y en la parte inferior del nivel de enlace o MAC. (Heikki Koivo, 2009).

MAC provee el acceso compartido de las tarjetas de red, es decir, se define la forma en la que se va a acceder al medio físico empleado en la red para el intercambio de datos. Además controla aspectos de sincronización y algoritmos del sistema de distribución.

La arquitectura MAC se compone de dos funcionalidades mostradas en la figura 2.13:

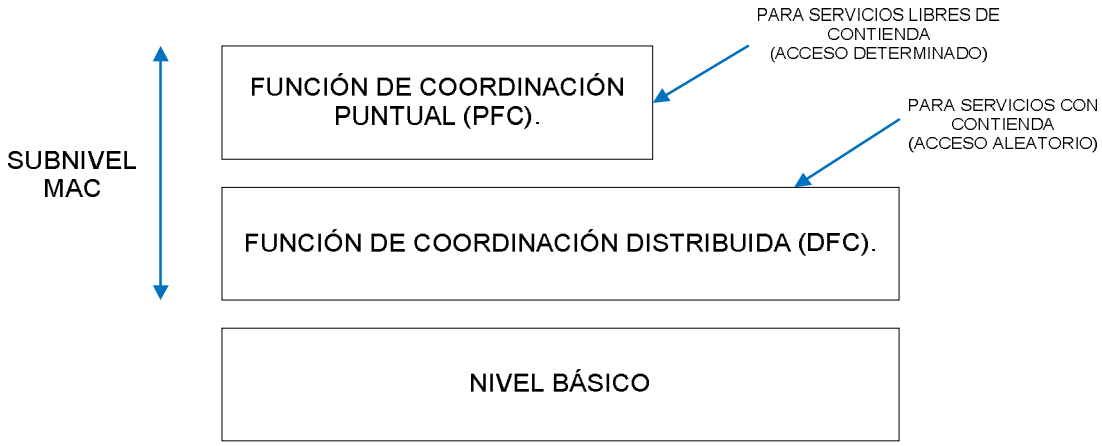


FIGURA 2.13 Arquitectura MAC.

- **La función de coordinación puntual (PCF)**

Es asociada a las transmisiones libres de contienda que utilizan técnicas de acceso deterministas. Es utilizada para servicios de tipo síncrono que no tolera retardos aleatorios en el acceso al medio. (Heikki Koivo, 2009).

- **La función de coordinación distribuida (DCF)**

Determina dentro de un conjunto básico de servicios cuándo un nodo puede transmitir y/o recibir datos de protocolo a nivel MAC a través del medio inalámbrico. (Heikki Koivo, 2009). Sus características son:

- Utiliza el acceso múltiple con prevención de colisiones (MACA) como protocolo de acceso al medio, se compone del acceso múltiple por detección de portadora con evasión de colisiones (CSMA/CA) con la petición de enviar/ listo para enviar – request to sent/ clear to send (RTS/CTS).

- Reconocimientos necesarios ACKs (acuse de recibo), provocando retransmisiones si no se reciben.

- Utiliza el campo Duration/ID que contiene el tiempo de reserva para transmisión y ACK. Es decir que todos los nodos sabrán al escuchar cuando el canal vuelva a quedar libre.

- Implementación de la fragmentación de datos.
- Concede prioridad a tramas mediante el espaciado entre tramas (**IFS**).
- Soporta Broadcast y multicast sin ACKs.

2.4.3.1 PROTOCOLOS DE ACCESO AL MEDIO

Los dos protocolos de acceso al medio en las redes inalámbricas son CSMA/CA y MACA.

El protocolo CSMA/CA presenta el siguiente funcionamiento que se puede observar en la figura 2.14 (Heikki Koivo, 2009):

1. Antes de transmitir información a un nodo debe analizar el canal inalámbrico, para determinar su estado, que puede ser libre o ocupado.

2. Si el canal no está ocupado por ninguna otra trama el nodo ejecuta una acción adicional llamada espaciado entre tramas.

3. Si durante este intervalo temporal, o bien desde el principio, el canal se determina ocupado, entonces el nodo debe esperar hasta el final de la transacción actual antes de realizar cualquier acción.

4. Una vez finalizada esta acción como consecuencia del canal ocupado el nodo ejecuta el algoritmo de *Backoff*, en el cual se determina una espera adicional y aleatoria escogida uniformemente en un intervalo llamado ventana de contienda (CW).
   El algoritmo de *Backoff* proporciona un número aleatorio y entero de ranura de tiempo (*slot time*) y su función es la de reducir la probabilidad de colisión que es máxima cuando varios nodos están esperando a que el canal quede libre para transmitir.

5. Mientras se ejecuta la espera marcada por el algoritmo se continúa escuchando el canal de tal manera que si el canal se determina libre durante un tiempo de al menos IFS esta espera va avanzando temporalmente hasta que el nodo consume todas las ranuras temporales asignadas. En cambio, si el canal no permanece libre durante un tiempo igual o superior a IFS el algoritmo queda suspendido hasta que se cumpla esta condición.

Cada retransmisión provoca que el valor de CW, se encuentre entre CW min y CW max, por otra parte el valor del *slot time* es de 20μseg.

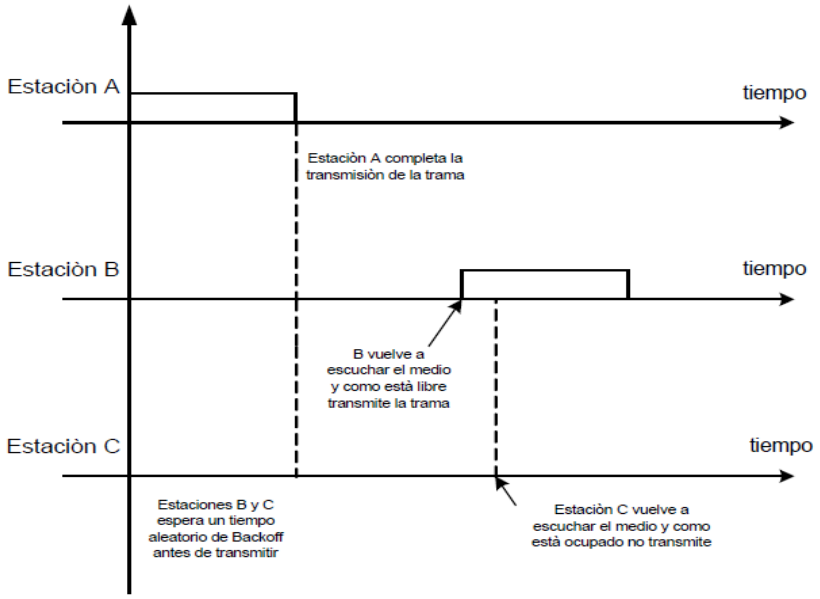


FIGURA 2.14 Funcionamiento del protocolo CSMA/CA.

Sin embargo, CSMA/CA en un entorno inalámbrico presenta una serie de problemas que se intentan resolver con alguna modificación. Los dos principales problemas que se detectan son (Heikki Koivo, 2009):

- Nodos ocultos: Un nodo cree que el canal está libre, pero en realidad está ocupado por otro nodo que no oye. (véase la figura 2.15).

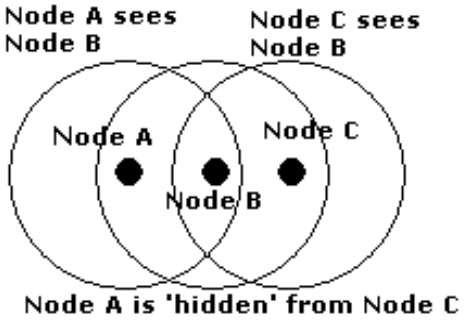


FIGURA 2.15 Ejemplo de nodos ocultos.

- Nodos expuestos: Un nodo cree que el canal está ocupado, pero en realidad está libre pues el nodo al que oye no le interfiere para transmitir a otro destino.

La solución que se propone a estos problemas es el protocolo MACA o *MultiAccess Collision Avoidance.* Según este protocolo, antes de transmitir el emisor envía una trama RTS (*Request to Send*), indicando la longitud de datos que se desea enviar. El receptor le contesta con una trama CTS (*Clear to send*), repitiendo la longitud. Al recibir el CTS, el emisor envía sus datos. (Heikki Koivo, 2009).

Los nodos se rigen por una serie de normas para evitar los nodos ocultos y expuestos:

- Al escuchar un RTS, hay que esperar un tiempo por el CTS
- Al escuchar un CTS, hay que esperar según la longitud de datos que se enviaran.

Con esto se llega a una solución final utilizando MACA con CSMA/CA para enviar los RTS y CTS.

2.4.3.2 ESPACIADO ENTRE TRAMAS IFS

El tiempo de intervalo entre tramas se define como espaciado entre tramas. Durante este periodo, un nodo estará escuchando el canal antes de trasmitir. Se puede encontrar con cuatro espaciados para dar prioridades de acceso al medio inalámbrico como se muestra en la figura 2.16. (Heikki Koivo, 2009).

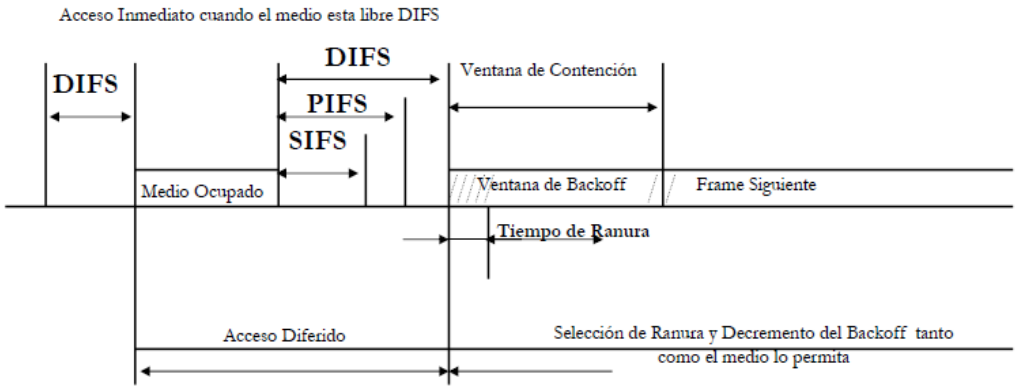


FIGURA 2.16 Espaciado entre tramas.

- SIFS – *Short IFS*: es el periodo más corto, se utiliza fundamentalmente para transmitir los reconocimientos. Trasmite cada uno de los fragmentos de una trama.
- PIFS: utilizado para ganar prioridad de acceso en los periodos libres de contienda.
- DIFS: es tiempo de espera habitual en las contiendas con mecanismo MACA.
- EIFS – *Extended IFS*: controla la espera en los casos en los que se detecta la llegada de una trama errónea y espera hasta que le vuelven a enviar la trama o una solución.

2.4.3.3 FORMATO DE LAS TRAMAS MAC

Las tramas MAC tienen los siguientes componentes básicos (Andrew S. Tanenbaum, 2003):

- Cabecera MAC – tiene campos de control, duración, direccionamiento y control de secuencia.
- Cuerpo de trama de longitud variable – contiene información específica del tipo de trama.
- Secuencia *checksum* – contiene un código de redundancia CRC de 32 bits.

A continuación se muestra la figura 2.17 donde se define el formato de una trama.

Marco de Control de	Duración ID	Dirección1 (fuente)	Address2 (destino)	Dirección3 (RX nodo)	Secuencia de control de	Dirección4 (TX nodo)	Datos	FCS
2	2	6	6	6	2	6	0 - 2.312	4

Campo	Bits	Valores	Notas / Descripción
Marco Control	15 a 14		Versión del protocolo. Actualmente 0
	13 a 12		Tipo
	11-8		Subtipo
	7		Para DS. 1 = el sistema de distribución.
	6		De DS. 1 = salida del sistema de distribución.
	5		Más Frag. 1 = más marcos fragmento de seguir marco (pasada o sin fragmentar = 0)
	4		Volver a intentar. 1 = esta es una re-transmisión.
	3		Power Mgt. 1 = la estación en modo ahorro de energía, 1 = el modo activo.
	2		Más datos. 1 = buffer marcos adicionales para la dirección de destino (dirección x).
	1		WEP. 1 = los datos procesados con el algoritmo WEP. 0 = sin WEP.
	0		Orden. 1 = marcos deben ser estrictamente ordenada.
Duración ID	15 a 0		Para tramas de datos = duración de la trama. Para los marcos de control asociados a la identidad de la estación transmisora.
Dirección 1	47 a 0		Dirección de origen (6 bytes).
Dirección 2	47 a 0		Dirección de destino (6 bytes).
Dirección 3	47 a 0		Recepción de dirección de la estación (la estación de destino inalámbrico)
Secuencia de control de	15 a 0		
Dirección 4	47 a 0		La transmisión de la estación inalámbrica.
De la estructura corporal			0 a 2312 octetos (bytes).
FCS	31 a 0		Frame Check Sequence (32 bits CRC). se define en P802.11.

FIGURA 2.17 Formato de una trama.

2.4.4 TOPOLOGÍAS

El grado de complejidad de una red inalámbrica es variable, dependiendo de las necesidades a cubrir y en función de los requerimientos del sistema que se desean implementar, se puede utilizar diversas configuraciones de red. (Andrew S. Tanenbaum, 2003).

1) Modo Peer to peer o redes ad-hoc

En el modo ad hoc (véase la figura 2.18) los equipos inalámbricos se conectan entre sí para formar una red punto a punto, es decir, una red en la que cada equipo actúa como cliente y como punto de acceso simultáneamente.

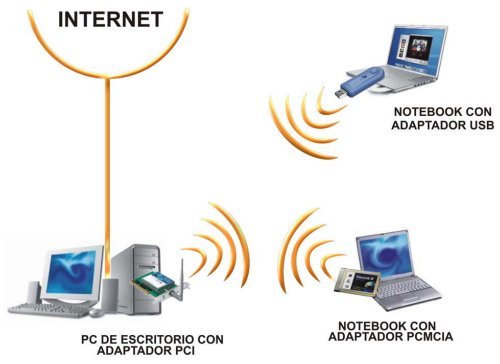


FIGURA 2.18 Ejemplo de una red ad-hoc.

La configuración que forman los nodos se llama conjunto de servicio básico independiente o IBSS. Un IBSS es una red inalámbrica que tiene comunicación al menos con dos nodos sin la intervención de un punto de acceso. En una red ad hoc, el rango del IBSS está determinado por el rango de cada nodo. Esto significa que si dos nodos de la red están fuera del rango del otro, no podrán comunicarse, ni siquiera cuando se puedan ver otros nodos. No tiene un sistema de distribución que pueda enviar tramas de datos desde un nodo a otro nodo. Entonces, por definición, un IBSS es una red inalámbrica restringida. (Andrew S. Tanenbaum, 2003).

2) Modo Infraestructura

En el modo de infraestructura (véase la figura 2.19), cada nodo se conecta a un punto de acceso a través de un enlace inalámbrico. La configuración formada por el punto de acceso y los nodos ubicados dentro del área de cobertura se llama conjunto de servicio básico (BSS). Cada BSS se identifica a través de un identificador de BSS, que es un identificador de 6 bytes representando al punto de acceso de la dirección MAC. (Andrew S. Tanenbaum, 2003).

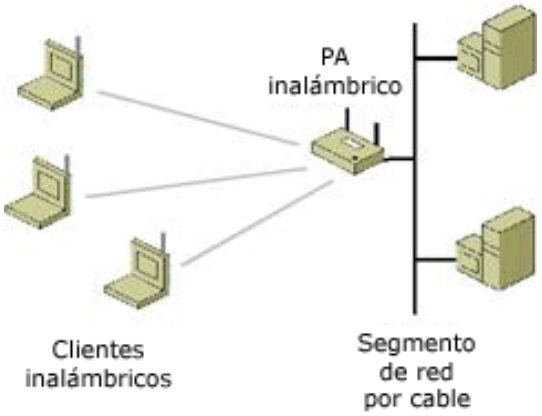


FIGURA 2.19 Ejemplo de una red de modo infraestructura.

Se pueden comunicar varios puntos de acceso en una conexión llamada sistema de distribución para formar un conjunto de servicio extendido (ESS). Un ESS se identifica a través de un identificador del conjunto de servicio extendido, que es un identificador de 32 caracteres en formato ASCII que muestra el nombre de la red. (Heikki Koivo, 2009).

Los puntos de acceso se comunican entre sí a través de un sistema de distribución con el fin de intercambiar información sobre los nodos y, si es necesario, para transmitir datos desde nodos móviles. Esta característica que permite a los nodos moverse de forma transparente de un punto de acceso al otro se denomina itinerancia. (Heikki Koivo, 2009).

La comunicación con los puntos de acceso se realiza siguiendo un específico procedimiento (Heikki Koivo, 2009):

- Cuando un nodo se une a una red, envía una solicitud para sondear cada canal. Esta solicitud contiene el identificador del conjunto de servicio extendido que la red está

configurada para usar y también el volumen de tráfico que su adaptador inalámbrico puede admitir.

- Cada punto de acceso transmite una señal en intervalos regulares. Esta señal provee información de su BSSID, sus características y su identificador del conjunto de servicio extendido, si corresponde, el identificador del conjunto de servicio extendido se transmite automáticamente en forma predeterminada.
- Cuando se recibe una solicitud, el punto de acceso verifica el identificador del conjunto de servicio extendido y la solicitud del volumen de tráfico encontrado en la señal. Si el identificador del conjunto de servicio extendido dado concuerda con el del punto de acceso, éste envía una respuesta con datos de sincronización e información sobre su carga de tráfico. Así, el nodo que recibe la respuesta puede verificar la calidad de la señal que envía el punto de acceso para determinar cuán lejos está. En términos generales, mientras más cerca un punto de acceso esté, más grande será su capacidad de transferencia de datos. Por lo tanto, un nodo dentro del rango de muchos puntos de acceso puede elegir el punto que ofrezca la mejor proporción entre capacidad de carga de tráfico y carga de tráfico actual.

2.4.5 ESTÁNDARES

IEEE 802.11 es un estándar internacional que define las características de una red inalámbrica. Wi-Fi es el nombre de la certificación otorgada por la *Wi-Fi Alliance*, anteriormente *WECA*, grupo que garantiza la compatibilidad entre dispositivos que utilizan el estándar 802.11. (Heikki Koivo, 2009).

El estándar 802.11 establece los niveles inferiores del modelo OSI para las conexiones inalámbricas que utilizan ondas electromagnéticas, en la tabla 2.1 se muestran algunos ejemplos:

- La capa física (PHY) ofrece tres tipos de codificación de información DSSS. FHSS e infrarrojo.
- La capa de enlace de datos compuesta por dos subcapas: control de enlace lógico (LLC) y control de acceso al medio (MAC).

Capa de enlace de datos (MAC)	**802.2**
	802.11
Capa física (PHY)	DSSS
	FHSS
	Infrarrojo

TABLA 2.1 Ejemplos de conexiones inalámbricas que utilizan ondas electromagnéticas.

El estándar original se ha modificado para optimizar el ancho de banda o para especificar componentes de mejor manera con el fin de garantizar mayor seguridad o compatibilidad. Los estándares 802.11a, 802.11b y 802.11g llamados estándares físicos, son modificaciones del estándar 802.11 y operan de modos diferentes, lo que les permite alcanzar distintas velocidades en la transferencia de datos según sus rangos. (Heikki Koivo, 2009). En la tabla 2.2 se describen los estándares físicos (Heikki Koivo, 2009):

Nombre del estándar	Nombre	Descripción
802.11a	Wifi5	Opera en la banda de 5 Ghz y utiliza 52 subportadoras *ortogonal frequency-division multiplexing (OFDM)* con una velocidad máxima de 54 Mbit/s. Tiene 12 canales no solapados, 8 para red inalámbricas y 4 para conexiones punto a punto. No puede tener comunicación con equipos que tengan el estándar 802.11b. Al utilizar la banda de 5 Ghz representa una ventaja del estándar, dado que se presentan menos interferencias, pero también tiene como desventaja que los equipos no pueden penetrar tan lejos como los estándares 802.11b dado que sus ondas son más fácilmente absorbidas.
802.11b	Wifi	Tiene una velocidad máxima de transmisión de 11 Mbit/s y utiliza el método de acceso CSMA/CA. Funciona en la banda de 2.4 Ghz, la velocidad máxima de transmisión es aproximadamente 5.9 Mbit/s sobre TCP y 7.1 Mbit/s sobre UDP. Es una extensión directa de la técnica de modulación DSSS. Consta de 11 canales disponibles para equipos inalámbricos y 3 sin solapamiento.
802.11d	Internacionalización	El estándar 802.11d es un complemento del estándar 802.11 que está pensado para permitir el uso internacional de las redes 802.11 locales
802.11g		Utiliza la banda de 2.4 Ghz y opera a una velocidad máxima de 54 Mbit/s. Es compatible con el estándar 802.11b. Emplea *OFDM*, el esquema de modulación usado en 802.11a, para obtener mayor velocidad de datos.
802.11i		El estándar 802.11i mejora la seguridad en la transferencia de datos. Este estándar se basa en el estándar de cifrado avanzado y puede cifrar transmisiones que se ejecutan en las tecnologías

	802.11a, 802.11b y 802.11g.
802.11n	El estándar 802.11n **se basa en la tecnología MIMO** opera en la banda de 2.4 GHz y es compatible con 802.11b y 802.11g, también puede operar en la banda de 5 GHz que es compatible con 802.11a.

TABLA 2.2 Descripción de los estándares físicos.

En la tabla 2.3 se muestra un resumen de los estándares físicos anteriormente descritos.

Protocolo	Año de creación	Frecuencia	Rango (aprox.)
802.11a	1999	5 Ghz	30 m
802.11b	1999	2.4 Ghz	30 m
802.11g	2003	2.4 Ghz	30 m
802.11n	2008	2.4 ó 5 Ghz	50 m

TABLA 2.3 Características principales de los estándares físicos.

2.4.6 SEGURIDAD

2.4.6.1 PROTOCOLO WEP (Wired Equivalent Protocol)

WEP es un sistema de encriptación débil que permite la protección de los datos. Sus características principales son:

- Forma parte de la especificación del estándar 802.11
- Opera en el nivel 2 del modelo OSI (subcapa MAC)
- Utiliza el algoritmo de encriptación RC4

El funcionamiento que presenta el protocolo WEP (véase la figura 2.20) es el siguiente (Heikki Koivo, 2009):

- Se tiene una clave secreta compartida entre un emisor y un receptor que puede tener una longitud de 40 o 128 bits.
- A la trama que se desea enviar, se le aplica un código de integración que se denomina *Integrity check value* (ICV) mediante el algoritmo CRC-32. Este código se asegura que lo recibido por el receptor corresponde exactamente con lo enviado por el emisor, es decir, la trama no fue modificada durante su trayecto.
- Posteriormente, se concatena la clave secreta con un número aleatorio llamado vector de inicialización, que tiene una longitud de 24 bits.
- El algoritmo de encriptación RC4 dispondrá de dos entradas:
  - La primera entrada será la clave secreta + vector de inicialización = semilla
  - Los datos modificados con el código de integridad.

Este algoritmo se basa en un proceso de XOR bit por bit generando la trama cifrada

- Se envía al receptor la trama cifrada junto con el vector de inicialización y el código de integridad sin encriptar.
- El receptor utiliza la clave secreta que tiene compartida con el emisor, junto con el vector de inicialización enviado para generar la semilla.
- Por último, el receptor calcula el código de integridad de los datos recibidos y los compara con el código de integridad recibido, y si no concuerda, se descarta tanto la trama como al emisor de la misma.

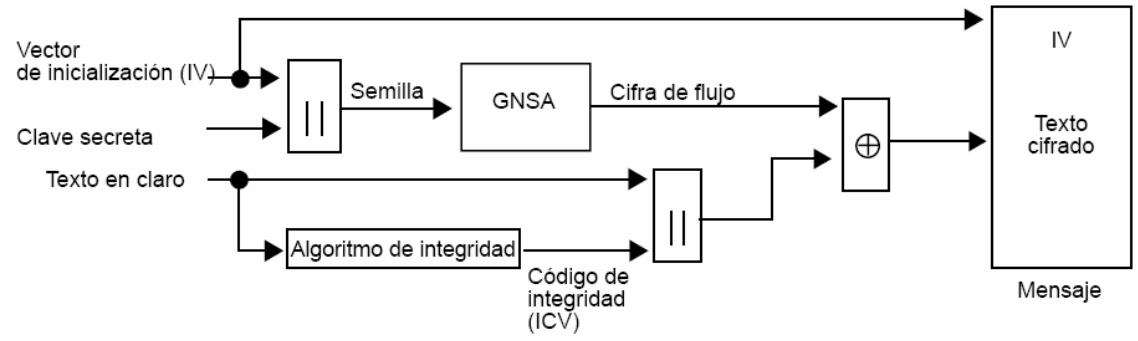


FIGURA 2.20 Funcionamiento del protocolo WEP.

Desventajas que presenta el protocolo WEP son (Heikki Koivo, 2009):

- Utiliza una clave estática.
- La modificación de la clave ha de hacerse de forma manual.
- El *password* del administrador es directamente la clave. Por ello la clave puede ser descubierta fácilmente.
- Todos los nodos que comparten el punto de acceso utilizan la misma clave.
- Resulta bastante sencillo romper la clave por fuerza bruta cuando se acumulan grandes cantidades de tráfico cifradas con la misma clave.
- El vector de inicialización utilizado es de longitud insuficiente.
- Existen problemas con el código de integridad. Dicho código, sirve para solucionar problemas del medio de transmisión, pero no permiten evitar modificaciones maliciosas, cambiando ciertos bits de datos y calculando los cambios del CRC-32 para mantenerlo coherente.

2.4.6.2 PROTOCOLO WAP (PROTOCOLO DE APLICACIONES INALÁMBRICAS)

El protocolo WAP se ha desarrollado para subsanar las debilidades de WEP, mejorando el cifrado de los datos y ofreciendo un mecanismo de autentificación.

Las características principales del protocolo WAP son (Andrew S. Tanenbaum, 2003):

- Propuesto por los miembros de la *Wi-Fi Alliance* en colaboración con la IEEE.

- Basado en el protocolo para cifrado TKIP (*Temporary Key Integrity Protocol*).
- La longitud de las claves pasa de 40 a 128 bits y el vector de inicialización, de 24 a 48 bits.
- La clave es generada de forma dinámica, para cada usuario, para cada sesión, y para cada paquete enviado, así como la distribución de claves, que también es realizada de forma automática.
- El mecanismo de autentificación basado en WPA emplea 802.1x/EAP

Se detalla brevemente el funcionamiento del protocolo de cifrado TKIP (Heikki Koivo, 2009):

- Basado en el algoritmo *Michael* para garantizar la integridad.
- Genera un bloque de 4 bytes (MIC) a partir de la dirección MAC de origen, de destino, y de los datos.
- Añade el MIC calculado a la unidad de datos a enviar.
- Posteriormente los datos se fragmentan y se les asigna un número de secuencia.
- La mezcla del número de secuencia con la clave temporal, genera la clave que será utilizada para cada fragmento.

El mecanismo de autentificación que incorpora WAP y que supone una mejora con respecto a WEP. En el mecanismo se describen dos modos de autentificación dependiendo de la modalidad en la que opere el punto de acceso (Heikki Koivo, 2009):

- Modalidad red empresarial: En este caso, se emplean los métodos soportados por EAP y se precisa de la existencia de un servidor *RADIUS* por ello estamos en el caso de redes considerablemente grandes, donde se precisa de una cierta infraestructura. Esta modalidad se basa en la existencia de 3 componentes:
  - El solicitante se encuentra en el nodo inalámbrico.
  - El autenticador se encuentra en el punto de acceso.
  - El servidor de autenticación o servidor *RADIUS*.

  El autenticador, va a crear un puerto lógico por cliente, una vez que el solicitante entra dentro del radio de cobertura del punto de acceso, dicho punto de acceso creará un puerto para éste, mientras el cliente no se haya autenticado. Solo se permitirá tráfico 802.1x/EAP hacia el servidor de autenticación bloqueando el resto del tráfico. (Heikki Koivo, 2009). La autenticación del cliente pasa por varias fases:

  - El cliente envía un mensaje *EAP Start*.
  - El autenticador responde con un mensaje *EAP Request Identity* para obtener la identidad del cliente.
  - El solicitante responde con *EAP Response* donde indica su identificador.
  - El autenticador reenviará la petición al servidor de autentificación (*RADIUS*).

- El cliente y servidor *RADIUS* pasarán a comunicarse directamente a partir de este momento, utilizando cierto algoritmo de autenticación negociado entre los dos.
- Una vez aceptada la autentificación del cliente por el servidor, el punto de acceso pasará el puerto asignado inicialmente al cliente, a un estado autorizado donde no se impondrán las restricciones de tráfico existentes inicialmente.

- Modalidad de red casera: También llamada PSK (*Pre- Shared Key*) utilizada cuando no se dispone de servidor *RADIUS*. La solución adoptada entonces, es introducir una contraseña compartida entre los clientes y el punto de acceso. De este modo solo podrán acceder al punto de acceso. Las terminales cuya contraseña coincida con la del punto de acceso. Una vez realizada la autenticación TKIP, entra en funcionamiento para garantizar la seguridad de acceso. (Heikki Koivo, 2009).

2.5 METODOLOGÍAS DE DISEÑO Y CONSTRUCCIÓN DE SISTEMAS

2.5.1 CICLO DE VIDA DE UNA APLICACIÓN DE BASES DE DATOS

Las etapas del ciclo de vida de una aplicación de bases de datos son las siguientes:

1. Planificación del proyecto

Esta etapa contiene las actividades a realizar durante las diferentes etapas del ciclo de vida del software, a fin de que se lleve a cabo de la manera más eficiente. Como apoyo a esta etapa, se necesita un modelo de datos corporativo en donde se muestren las entidades principales y sus relaciones. Normalmente, este modelo de datos se representa mediante un diagrama entidad-relación. La planificación de la base de datos también incluye el desarrollo de estándares que especifiquen cómo realizar la recolección de datos, cómo especificar su formato, qué documentación será necesaria y cómo se va a llevar a cabo el diseño y la implementación. (Mercedes Marqués, 2003).

2. Definición del sistema

En esta etapa se especifica el ámbito y los límites de la aplicación de bases de datos, así como con qué otros sistemas interactúan. También hay que determinar quiénes son los usuarios y las áreas de aplicación. (Mercedes Marqués, 2003).

3. Recolección y análisis de los requisitos

En esta etapa se recogen y analizan los requerimientos de los usuarios y de las áreas de aplicación. (Mercedes Marqués, 2003). Esta información se puede recoger de varias formas:

- Entrevistando al personal, concretamente, a aquellos que son considerados expertos en las áreas de interés.
- Observando el funcionamiento del área.

- Examinando documentos, sobre todo aquellos que se utilizan para recoger o visualizar información.
- Utilizando cuestionarios para recoger información de grandes grupos de usuarios.
- Utilizando la experiencia adquirida en el diseño de sistemas similares.

Esta etapa tiene como resultado un conjunto de documentos con las especificaciones de requisitos de los usuarios, en donde se describen las operaciones que se realizan desde distintos puntos de vista. La información recogida se debe estructurar utilizando técnicas de especificación de requisitos, como por ejemplo técnicas de análisis y diseño estructurado y diagramas de flujo de datos. También las herramientas *CASE* pueden proporcionar una asistencia automatizada que garantice que los requisitos son completos y consistentes. (Mercedes Marqués, 2003).

4. Diseño de la base de datos

Esta etapa consta de tres fases: diseño conceptual, lógico y físico. Hay varias estrategias a seguir para realizar el diseño (Mercedes Marqués, 2003):

1. La estrategia de abajo a arriba parte de todos los atributos y los va agrupando en entidades y relaciones. Es apropiada cuando la base de datos es simple, con pocos atributos. de arriba abajo.
2. La estrategia de arriba a abajo es más apropiada cuando se trata de bases de datos complejas. Se comienza con un esquema con entidades de alto nivel, que se van refinando para obtener entidades de bajo nivel, atributos y relaciones.
3. La estrategia de dentro a fuera es similar a la estrategia de abajo a arriba, pero difiere en que se parte de los conceptos principales y se va extendiendo el esquema para considerar también otros conceptos, asociados con los que se han identificado en primer lugar.
4. La estrategia mixta utiliza ambas estrategias, de abajo a arriba y de arriba a abajo, con un esquema de divide y vencerás. Se obtiene un esquema inicial de alto nivel, se divide en partes, y de cada parte se obtiene un subesquema. Estos subesquemas se integran después para obtener el modelo final.

5. Selección del SGBD

Se debe escoger un SGBD que sea adecuado para el sistema de información. Esta elección se debe hacer en cualquier momento antes del diseño lógico. Para seleccionar el SGBD se toman los siguientes criterios: costos, características generales, modelo subyacente, portabilidad, etc. (Mercedes Marqués, 2003).

6. Diseño de la aplicación

En esta etapa se diseñan los programas de aplicación que usarán y procesarán la base de datos. Esta etapa y el diseño de la base de datos, son paralelas. Por otro lado, la base de datos existe para dar soporte a las aplicaciones, por lo que habrá una realimentación desde el diseño de las aplicaciones al diseño de la base de datos. Hay que asegurarse de que toda la funcionalidad especificada en los requisitos de usuario se encuentre en el diseño de la aplicación. Habrá algunos

programas que utilicen y procesen los datos de la base de datos. Además, habrá que diseñar las interfaces de usuario, aspecto muy importante que se suele ignorar. El sistema debe ser fácil de aprender, de usar y ser directo. Si la interface no tiene estas características, el sistema no sería útil y/o usable. (Mercedes Marqués, 2003).

7. Prototipado

Esta etapa opcional, es para construir prototipos de la aplicación que permitan a los diseñadores y a los usuarios probar el sistema. Un prototipo es un modelo de trabajo de las aplicaciones del sistema. El prototipo no tiene toda la funcionalidad del sistema final, pero es suficiente para que los usuarios puedan utilizar el sistema e identificar qué aspectos están bien y cuáles no son adecuados, además de poder sugerir mejoras o la inclusión de nuevos elementos. Este proceso permite que quienes diseñan e implementan el sistema sepan si han interpretado correctamente los requisitos de los usuarios. (Mercedes Marqués, 2003)

8. Implementación

En esta etapa se crean las definiciones de la base de datos a nivel conceptual, externo e interno, así como los programas de aplicación. La implementación de la base de datos se realiza mediante las sentencias del lenguaje de definición de datos del SGBD escogido. Estas sentencias se encargan de crear el esquema de la base de datos, los archivos en donde se almacenarán los datos y las vistas de los usuarios. (Mercedes Marqués, 2003).

Los programas de aplicación se implementan utilizando lenguajes de tercera o cuarta generación. Parte de estas aplicaciones son transacciones sobre la base de datos, que se implementan mediante el lenguaje de manejo de datos del SGBD (SQL). Las sentencias SQL pueden estar inmersas dentro del código de un lenguaje de programación anfitrión como Visual Basic, Delphi, C, C++, Java, COBOL, Fortran, Ada o Pascal. En esta etapa, también se implementan los menús, los formularios para la introducción de datos y los informes de visualización de datos. Para ello, el SGBD puede disponer de lenguajes de cuarta generación que permiten el desarrollo rápido de aplicaciones mediante lenguajes de consultas no procedurales, generadores de informes, de formularios, de gráficos y de aplicaciones. También se implementan en esta etapa todos los controles de seguridad e integridad. Algunos de estos controles se pueden implementar mediante el lenguaje de definición de datos y otros implementarlos mediante utilidades del SGBD o mediante programas de aplicación. (Mercedes Marqués, 2003).

9. Conversión y carga de datos

Esta etapa es necesaria cuando se está reemplazando un sistema antiguo por uno nuevo. Los datos se cargan desde el sistema viejo al nuevo directamente o, si es necesario, se convierten al formato que requiera el nuevo SGBD y luego se cargan. Si es posible, los programas de aplicación del sistema antiguo también se convierten para que se puedan utilizar en el sistema nuevo. (Mercedes Marqués, 2003).

10. Prueba

En esta etapa se prueba y valida el sistema con los requisitos especificados por los usuarios. Si la fase de prueba se lleva a cabo correctamente, se descubrirán los errores en los programas de aplicación y en la estructura de la base de datos. Por último, en las pruebas se podrá hacer una medida de la fiabilidad y la calidad del software desarrollado. (Mercedes Marqués, 2003).

11. Mantenimiento

Una vez que el sistema está completamente implementado y probado, se pone en marcha. (Mercedes Marqués, 2003). El sistema está ahora en la fase de mantenimiento en la que se llevan a cabo las siguientes tareas:

- Monitorear las prestaciones del sistema. Si las prestaciones caen por debajo de un determinado nivel, puede ser necesario reorganizar la base de datos.
- Mantenimiento y actualización del sistema. Cuando sea necesario, los nuevos requisitos que vayan surgiendo se incorporarán al sistema, siguiendo de nuevo las etapas del ciclo de vida que se acaban de presentar.

2.5.2 MODELO DE CICLO DE VIDA

Los modelos de ciclo de vida se han actualizado para reflejar las etapas de desarrollo involucradas y la documentación requerida, de manera que cada etapa se valide antes de continuar con la siguiente etapa. (Roger S. Pressman, 2002).

2.5.2.1 MODELO CASCADA

Este es el más básico de todos los modelos, y sirve como bloque de construcción para los demás modelos de ciclo de vida. La visión del modelo cascada del desarrollo de software es muy simple; dice que el desarrollo de software puede ser a través de una secuencia simple de fases. Cada fase tiene un conjunto de metas bien definidas, y las actividades dentro de una fase contribuyen a la satisfacción de metas de esa fase o quizás a una subsecuencia de metas de la fase. (Roger S. Pressman, 2002)

Las flechas muestran el flujo de información entre las fases. La flecha de avance muestra el flujo normal. Las flechas hacia atrás representan la retroalimentación.
El modelo de ciclo de vida en cascada (véase la figura 2.21), captura algunos principios básicos:

- Planear un proyecto antes de embarcarse en él.
- Definir el comportamiento externo deseado del sistema antes de diseñar su arquitectura interna.
- Documentar los resultados de cada actividad.
- Diseñar un sistema antes de codificarlo.
- Probar un sistema después de construirlo.

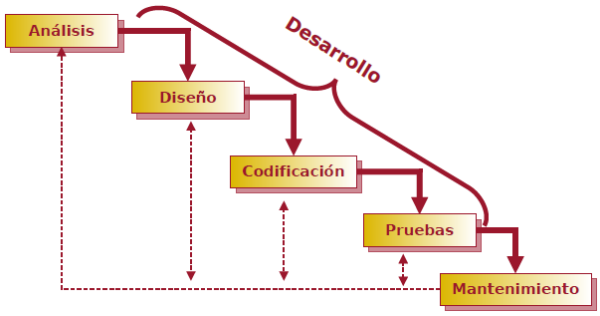


FIGURA 2.21 Modelo de ciclo de vida en cascada.

2.5.2.2 MODELO DE DESARROLLO INCREMENTAL

Los riesgos asociados con el desarrollo de sistemas largos y complejos son enormes. Una forma de reducir los riesgos es construir sólo una parte del sistema, reservando otros aspectos para niveles posteriores. El desarrollo incremental es el proceso de construcción siempre incrementando subconjuntos de requerimientos del sistema. Típicamente, un documento de requerimientos es escrito al capturar todos los requerimientos para el sistema completo. (Roger S. Pressman, 2002).

El desarrollo incremental no demanda una forma específica de observar el desarrollo de algún otro incremento. Así, el modelo cascada puede ser usado para administrar cada esfuerzo de desarrollo. El modelo de desarrollo incremental (véase la figura 2.22) provee algunos beneficios significativos para los proyectos:

- Construir un sistema pequeño es siempre menos riesgoso que construir un sistema grande.
- Al ir desarrollando parte de las funcionalidades, es más fácil determinar si los requerimientos planeados para los niveles subsiguientes son correctos.
- Si un error importante es realizado, sólo la última iteración necesita ser descartada.
- Reduciendo el tiempo de desarrollo de un sistema decrecen las probabilidades que esos requerimientos de usuarios puedan cambiar durante el desarrollo.
- Si un error importante es realizado, el incremento previo puede ser usado.
- Los errores de desarrollo realizados en un incremento, pueden ser arreglados antes del comienzo del próximo incremento.

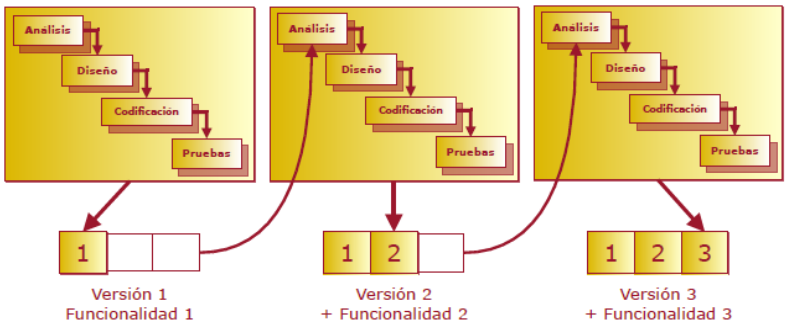


FIGURA 2.22 Modelo de desarrollo incremental.

2.6 UML

El Lenguaje Unificado de Modelado preescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas. (Roger S. Pressman, 2002).

- Diagramas de Casos de Uso para modelar los procesos '*business*'.
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.
- Diagramas de Implementación para modelar la distribución del sistema.

2.7 SEGURIDAD SSL

SSL (*Secure Socket Layers*) es un proceso que administra la seguridad de las transacciones que se realizan en un sistema. El estándar SSL fue desarrollado por *Netscape*, junto con *Mastercard*, *Bank of America*, *MCI* y *Silicon Graphics*. Se basa en un proceso de cifrado de clave pública que garantiza la seguridad de los datos que se envían. Su principio consiste en el establecimiento de un canal de comunicación seguro (cifrado) entre dos equipos, el cliente y el servidor, después de una fase de autenticación. SSL actúa como una capa adicional que permite garantizar la seguridad de los datos y que se ubica entre la capa de la aplicación y la capa de transporte. A mediados de 2001, la patente SSL, que hasta ese momento había pertenecido a *Netscape*, fue adquirida por *IETF* (*Internet Engineering Task Force*) y adoptó el nombre de TLS (*Transport Layer Security*). (Martínez Rafael, 2009).

La seguridad de las transacciones a través de SSL se basa en el intercambio de claves entre un cliente y un servidor. Una transacción segura SSL se realiza de acuerdo al siguiente modelo que se puede visualizar en la figura 2.23:

- Primero, el cliente se conecta al servidor protegido por SSL y pide la autenticación. El cliente también envía la lista de los criptosistemas que soporta, clasificada en orden descendente por la longitud de la clave.
- El servidor que recibe la solicitud envía un certificado al cliente que contiene la clave pública del servidor firmado por una entidad de certificación (CA), y también el nombre del criptosistema que está más alto en la lista de compatibilidades (la longitud de la clave de cifrado - 40 o 128 bits - será la del criptosistema compartido que tiene el tamaño de clave de mayor longitud).

- El cliente verifica la validez del certificado, luego crea una clave secreta al azar, cifra esta clave con la clave pública del servidor y envía el resultado del servidor (clave de sesión).
- El servidor es capaz de descifrar la clave de sesión con su clave privada. De esta manera, hay dos entidades que comparten una clave que sólo ellos conocen. Las transacciones restantes pueden realizarse utilizando la clave de sesión, garantizando la integridad y la confidencialidad de los datos que se intercambian.

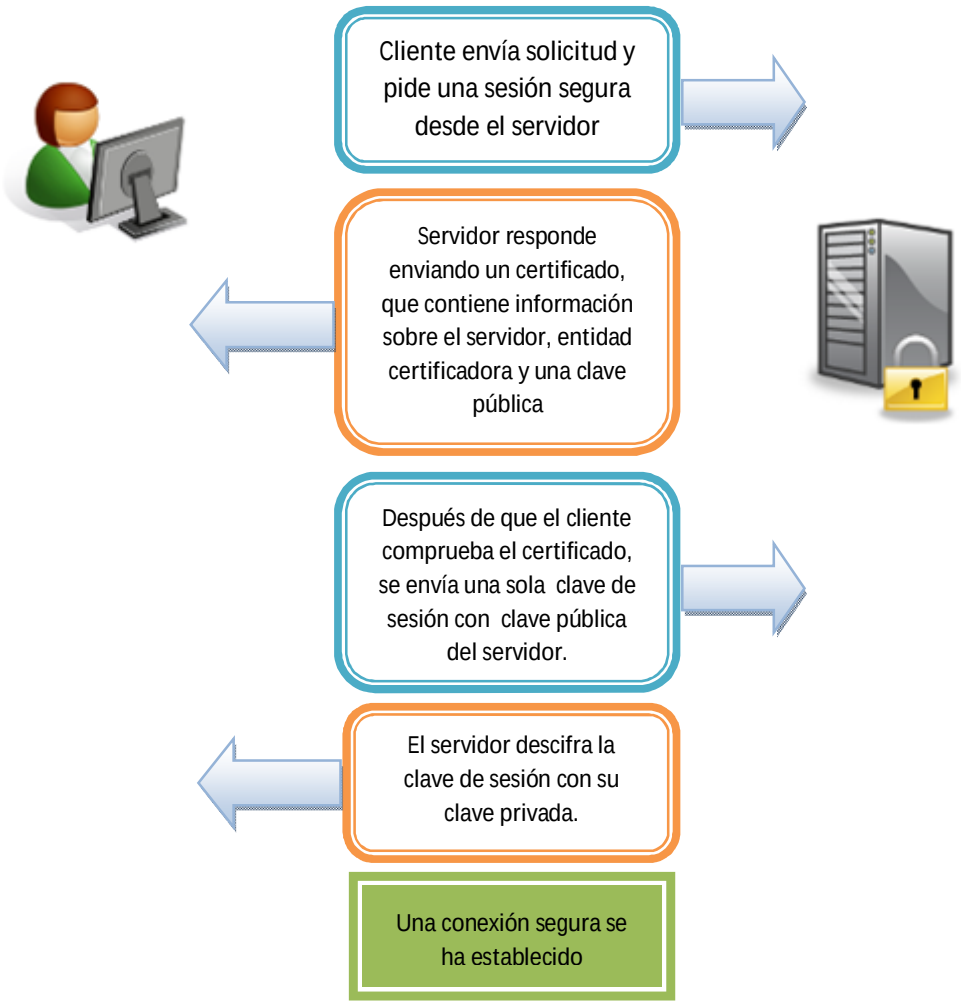


FIGURA 2.23 Modelo de la seguridad SSL.

CAPÍTULO 3

DISEÑO

Diseñar es una actividad abstracta que implica
programar, proyectar, traducir lo invisible
en visible, comunicar.
Jorge Frascara

Antes de iniciar el capítulo 3 es necesario mencionar que el nombre del sistema será *SIDIPAT* que significa Sistema distribuido del Departamento de Servicio de Patología. De aquí en adelante cuando se refiera al sistema se hará por su nombre.

3.1 DISEÑO DE LA BASE DE DATOS DISTRIBUIDA

Al planear el diseño de una base de datos distribuida se debe considerar la forma en la que será asignada la información entre los nodos que están involucrados, para asegurar un porcentaje alto en la eficiencia del acceso a la información. Por lo anterior se busca abordar dos cuestiones importantes. Primero la manera que será fragmentada la información, y la ubicación en los nodos de cada fragmento de información y segundo determinar si la información será replicada, es decir, señalar que datos tendrán múltiples copias y, de ser así, determinar cómo mantener la consistencia de toda la información. En el diseño también se debe contemplar los diferentes usuarios que pueden existir, como usuarios globales o locales.

En las siguientes secciones de éste capítulo se desglosa detalladamente los pasos que se llevan a cabo para el diseño de la base de datos distribuida que se requiere en el Departamento de Servicio de Anatomía Patologíca del Hospital Regional 1° de Octubre ISSSTE.

3.1.1 DISEÑO CONCEPTUAL

El diseño conceptual representa los objetos del mundo real, en este caso el funcionamiento del Departamento de Servicio de Anatomía Patológica. A través de un modelo de datos que contenga los objetos necesarios para tener un claro entendimiento del proceso del registro de solicitudes e informes que tiene dicho Departamento. El modelo debe permitir representar necesidades futuras. Lo anterior se fundamenta en la regla de los datos mínima:

"Todo lo que se requiere está allí, y todo lo que está allí se requiere"
(Peter Rob, Carlos Coronel, 2004)

Para poder construir un modelo que contemple en su totalidad todos los datos que se requieren para desarrollar adecuadamente el registro de solicitudes e informes, definiremos ciertas reglas de negocio que ayudaran a describir, crear y aplicar ciertas operaciones que se realizan, convirtiéndose en entidades, atributos y relaciones.

Las reglas de negocio en las cuales se sustenta el modelo son:

- Cada paciente puede estar asignado a una o más solicitudes.
- Cada solicitud debe ser asignada a uno y solamente un paciente.
- Cada servicio puede estar gestionando una o más solicitudes.
- Cada solicitud debe ser realizada uno y solamente un servicio.
- Cada solicitud debe estar asociada a una y solamente una solicitud anatomopatológica.
- Cada solicitud anatomopatológica debe estar relacionada a una y solamente una solicitud.
- Cada solicitud debe estar asociada a una y solamente una solicitud citológica.
- Cada solicitud citológica debe estar relacionada a una y solamente una solicitud.

- Cada solicitud anatomopatológica e informe anatomopatológico debe ser de uno y solamente un tipo anatomopatológico.
- Cada tipo anatomopatológico puede estar involucrado en una o más solicitudes anatomopatológicas e informes anatomopatológicos.
- Cada solicitud citológica e informe citológico debe ser de uno y solamente un tipo citológico.
- Cada tipo citológico puede estar involucrado en una o más solicitudes citológicas e informes citológicos.
- Cada solicitud debe estar designada a uno y solamente un informe.
- Cada informe debe estar adjudicado a una y solamente una solicitud.
- Cada solicitud debe ser de un y solamente un tipo.
- Cada tipo puede estar atribuido a una o más solicitudes.
- Cada informe debe ser de un y solamente un tipo.
- Cada tipo puede estar atribuido a uno o más informes.
- Cada solicitud citológica debe estar relacionada con uno y solamente un aspecto de cáncer, síntoma, procedencia de muestra, utensilio, producto, método de obtención, exploración y ginecobstétrico.
- Cada aspecto de cáncer, síntoma, procedencia de muestra, utensilio, producto, método de obtención, exploración y ginecobstétrico puede estar presente en una o más solicitudes citológicas.
- Cada informe debe estar asociado a uno y solamente un informe anatomopatológico.
- Cada informe anatomopatológico debe estar relacionada a uno y solamente un informe.
- Cada informe debe estar asociado a uno y solamente un informe citológico.
- Cada informe citológico debe estar relacionado a uno y solamente un informe.
- Cada informe citológico ó informe anatomopatológico debe ser revisado por uno y solamente un patólogo
- Cada patólogo puede ser responsable de revisar uno o más informes citológicos ó anatomopatológicos.
- Cada informe citológico debe estar asociado a uno y solamente un diagnostico citológico, un hallazgo, una muestra (característica).
- Cada diagnostico citológico, un hallazgo, muestra (característica) puede estar relacionado a uno o más informes citológicos.
- Cada informe citológico debe ser revisado por uno y solamente un citotecnólogo
- Cada citotecnólogo puede estar a cagar de uno o más informes citológicos.

Existen diferentes tipos de modelos conceptuales que se utilizan para el diseño de una base de datos, entre los más conocidos se encuentran: modelo E-R, modelo RM-T y modelos semánticos. El modelo E-R fue introducido por Peter Chen en 1976, permite describir las entidades que estarán involucradas en nuestra base de datos, así como las relaciones entre ellas, utilizando una notación grafica que hará posible tener un claro panorama del comportamiento de los datos de acuerdo a nuestras reglas de negocio. En la figura 3.1 se muestra el diagrama del modelo E-R que se utilizará para el diseño de la base de datos del Sistema para el Hospital Regional del ISSSTE 1°de Octubre.

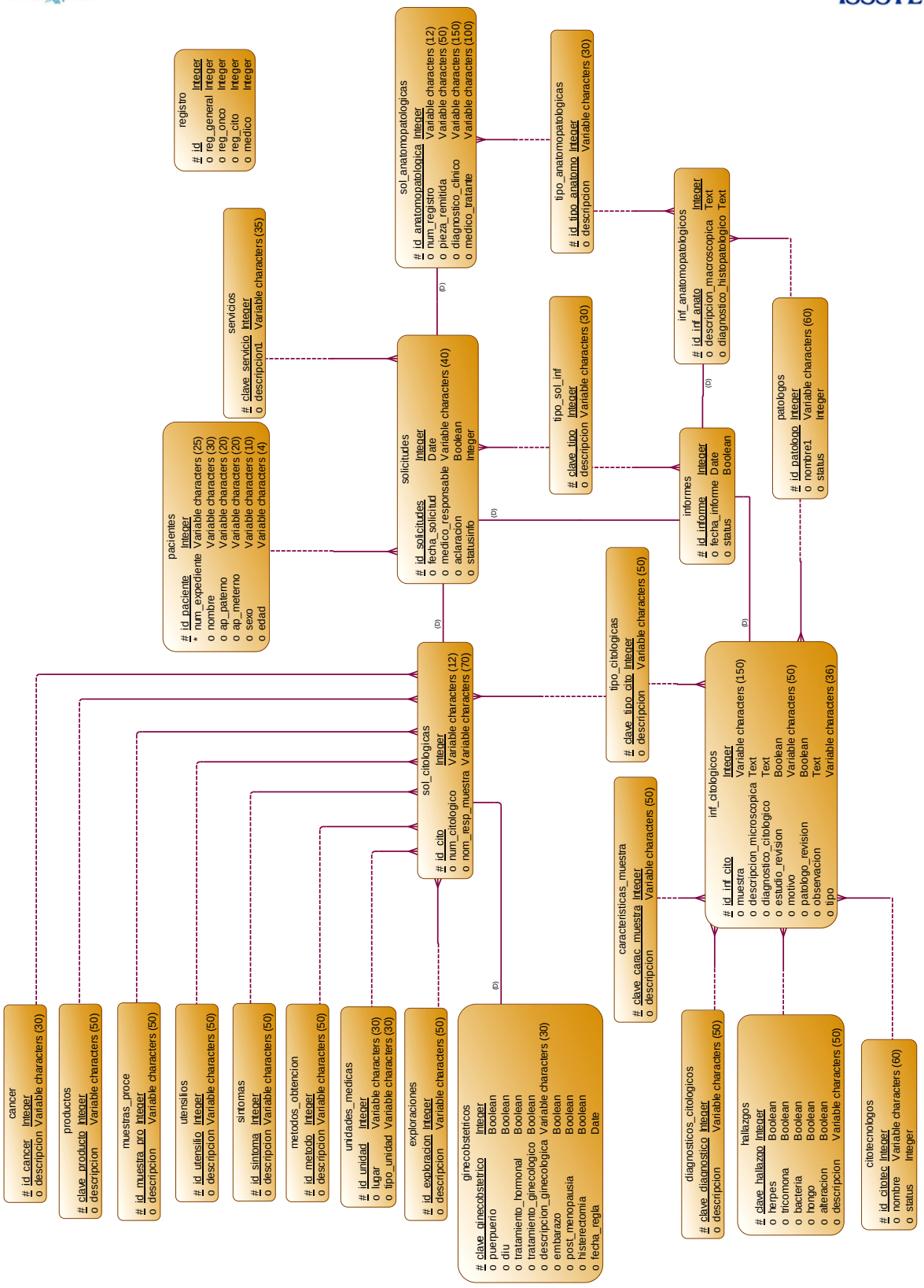


FIGURA 3.1 Diagrama Entidad – Relación de la base de datos del Departamento de Servicio de Anatomía Patológica.

3.1.1.1NORMALIZACIÓN

El siguiente paso durante el diseño de la base de datos es la normalización. La normalización nos ayudará a colocar los atributos en la entidad correspondiente, para minimizar la redundancia de los datos. La normalización es una parte muy importante en el proceso del diseño de una base de datos, sin embargo, se debe contemplar la demanda de información que el usuario requiere para tener un buen desempeño de la aplicación.

Para mostrar la función de la normalización se utilizará como ejemplo la entidad de **SOLICITUD** que se muestra en la figura 3.2, a la cual se le aplicarán las 3 formas normales.

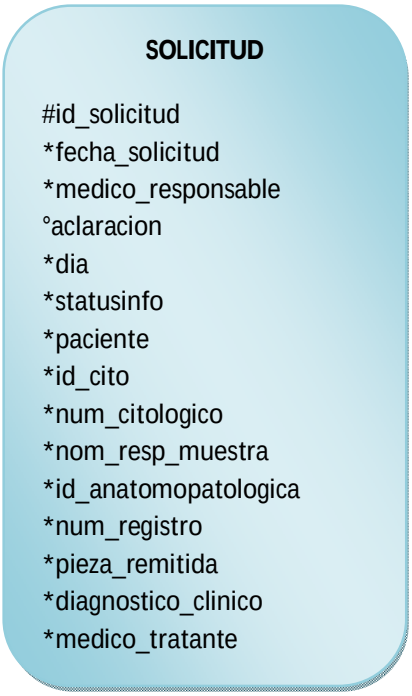


FIGURA 3.2 Diagrama de la entidad SOLICITUD.

- PRIMERA FORMA NORMAL (1 FN)

Se cumple la 1 FN si y sólo si todos los dominios son atómicos. Un dominio es atómico si los elementos del dominio son indivisibles, es decir, no debe contener un conjunto de valores. En la siguiente figura 3.3 se muestra un ejemplo donde se viola dicha forma normal.

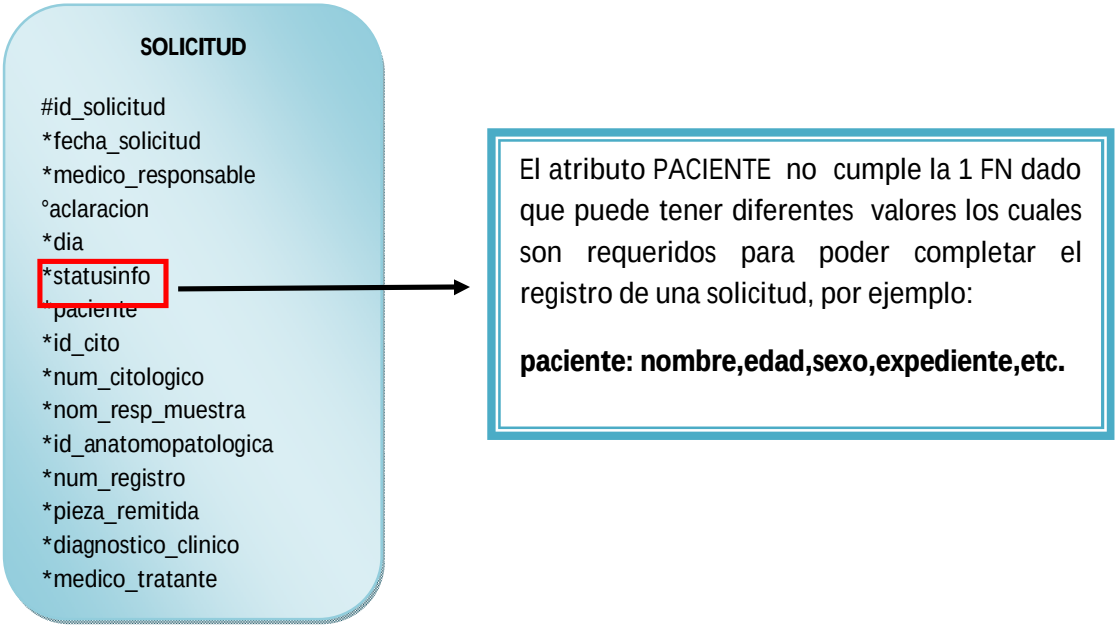


FIGURA 3.3 Ejemplo del incumplimiento de la 1 FN en la entidad **SOLICITUD**.

En la figura 3.4 se describe la solución a lo anterior, la cual consiste en separar en dos entidades, cada una con sus atributos correspondientes.

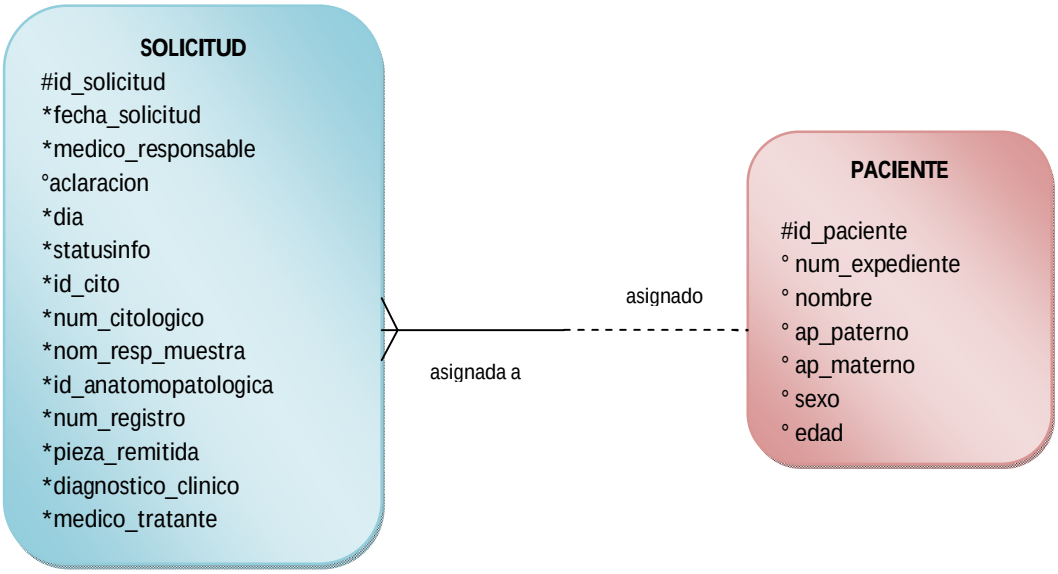


FIGURA 3.4 Solución a la 1FN de la entidad **SOLICITUD**.

- SEGUNDA FORMA NORMAL (2 FN) Y TERCER FORMA NORMAL (3 FN)

Es válida la 2 FN si cumple dos condiciones:

1. No violar la 1 FN
2. Los atributos de dicha entidad dependen completamente de la llave primaria.

La 3 FN es admitida si

1. Está en 2 FN
2. No existen dependencias transitivas entre los atributos, entendiendo como dependencias transitivas cuando existe más de una forma de llegar a referencias a un atributo de una relación

A continuación se muestra en la figura 3.5 las razones por las que no se cumplen la 2 FN y 3 FN en la entidad **SOLICITUD**.

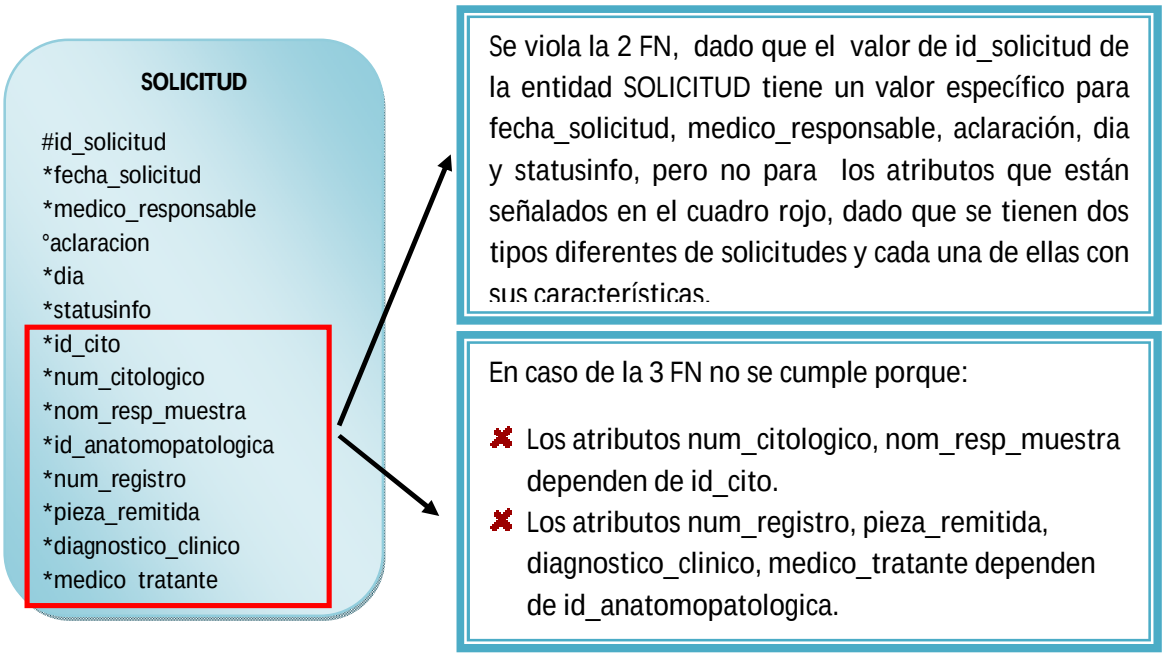


FIGURA 3.5 Ejemplo donde no es válida la 2 FN y 3 FN en la entidad **SOLICITUD**.

Al crear dos nuevas entidades llamadas **SOL_CITOLOGICA** y **SOL_ANATOMOPATOLOGICA** se eliminan las causas que ocasionan que no se cumplan estas formas normales. Esta solución se observa en la figura 3.6.

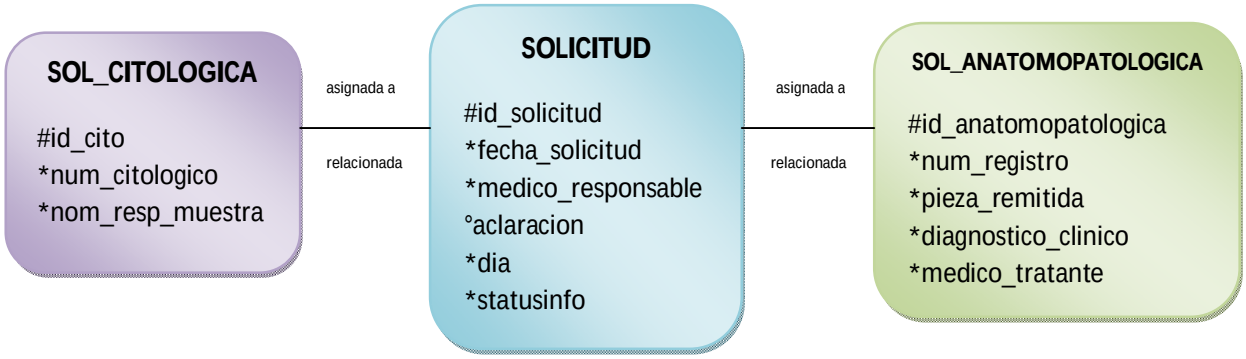
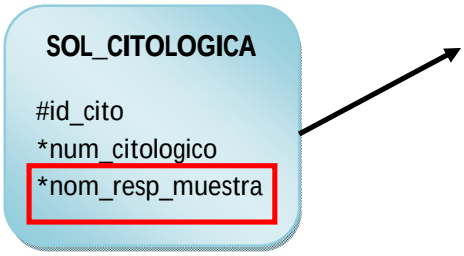


FIGURA 3.6 Solución a la 2 FN y 3 FN de la entidad **SOLICITUD**.

Para ejemplificar la situación en la que no se aplica la normalización, se realiza un análisis de todas las entidades considerando la información que cada una de ellas requiera para cumplir las necesidades que tenga el Departamento de Servicio de Anatomía Patológica. Una de las entidades que entra en este caso es SOL_CITOLOGICA.



El atributo nom_resp_muestra representa al médico que se encarga de tomar la muestra para ser analizada. El Departamento únicamente requiere el nombre del médico, por tanto, no es conveniente crear otra entidad para ubicar dicho atributo, dado que los nombres de los médicos cambian constantemente y no se requiere mayor información de ellos.

FIGURA 3.7 Ejemplo de la entidad **SOL_CITOLOGICA** donde no se aplica la normalización.

Dada la explicación en la figura 3.7 acerca de la normalización se ésta cuidando el desempeño del Sistema, al igual que se brinda al usuario información con un nivel bajo de redundancia e integridad.

3.1.1.2 *DICCIONARIO DE DATOS*

El diccionario de datos esta basado en el diagrama entidad – relación de la base de datos del Departamento de Servicio de Anatomía Patológica mostrado en la figura 3.1. Dicho diccionario contiene las características de los atributos de cada una de las entidades. El formato en el que se muestra la información se observa en la tabla 3.1.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
Nombre del atributo	INTEGER CHARACTER VARYING TEXT BOOLEAN	NOT NULL NULL	Pequeña explicación del atributo

TABLA 3.1 Formato que tiene el diccionario de datos para mostrar la información.

El diccionario completo se puede consultar en el apéndice B.

3.1.1.3 *NIVELES DE TRANSPARENCIA*

Durante el diseño de la base de datos distribuida del Departamento de Servicio de Anatomía Patológica se muestran los niveles requeridos de transparencia, los cuales se indican a continuación:

- Transparencia de distribución

Para mostrar todos los niveles de transparencia que contempla la transparencia distribuida, se utiliza la entidad de SOL_ANATOMOPATOLOGICAS. Los datos de dicha entidad están divididos en cinco nodos de la red Patologo1, Patologo2,..., Patologo5 como se muestra en la figura 3.8.

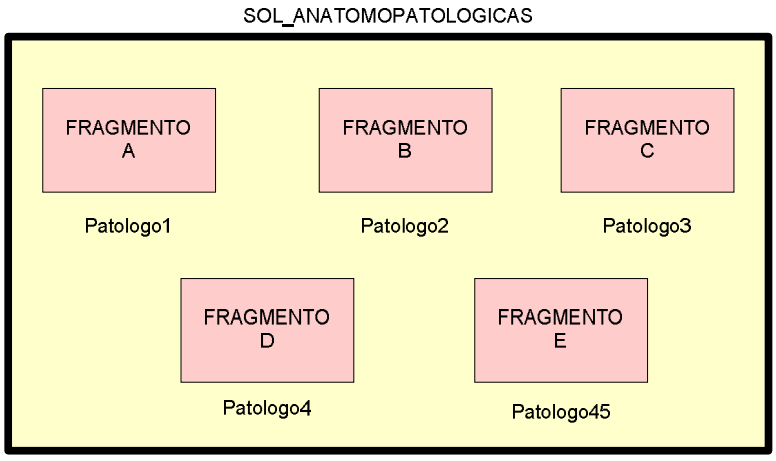


FIGURA 3.8 División de los datos de la entidad SOL_ANATOMOPATOLOGICAS en cinco nodos.

En la tabla 3.2 se observan algunas consultas que ayudan a obtener la inforcación de todas las solicitudes anatomopatológicas que tengan un informe asociado a ellas, sin importan en que fragmento se encuentren.

Transparencia de fragmentación	**SELECT * FROM SOL_ANATOMOPATOLOGICAS WHERE statusinfo=1**	
Transparencia de ubicación	SELECT * FROM **FRAGMENTO A** WHERE statusinfo=1 UNION SELECT * FROM **FRAGMENTO B** WHERE statusinfo=1 UNION SELECT * FROM **FRAGMENTO C** WHERE statusinfo=1	UNION SELECT * FROM **FRAGMENTO D** WHERE statusinfo=1 UNION SELECT * FROM **FRAGMENTO E** WHERE statusinfo=1
Transparencia de Ubicación local	SELECT * FROM **FRAGMENTO A NODE Patologo1** WHERE statusinfo=1 UNION SELECT * FROM **FRAGMENTO B NODE Patologo2** WHERE statusinfo=1 UNION SELECT * FROM **FRAGMENTO C NODE Patologo3** WHERE statusinfo=1	UNION SELECT * FROM **FRAGMENTO D NODE Patologo4** WHERE statusinfo=1 UNION SELECT * FROM **FRAGMENTO E NODE Patologo5** WHERE statusinfo=1

TABLA 3.2 Consultas que ejemplifican la transparencia de distribución.

Para la implementación de la base de datos distribuida, es fundamental soportar transparencia de ubicación y transparencia de fragmentación para poder realizar las consultas distribuidas que requiere el Departamento.

- Transparencia de transacción

Para esta transparencia se debe tener en cuenta dos cosas:

Las transacciones distribuidas:

- Hacen referencia a distintos nodos (locales y remotos).
- Continen solicitudes en donde cada una puede accesar a un sitio remoto a la vez.

Para las solicitudes distribuidas:

- Se requiere dividir una tabla en varios fragmentos.
- Se puede hacer referencia a varios fragmentos con una sola solicitud, es decir utilizar la transparencia de fragmentación.

Por lo anterior se utiliza el protocolo commit de dos fases (2PC) ilustrado en la figura 3.9, lo que ayudará a controlar las operaciones que se lleven a cabo entre dos nodos. Este protocolo realiza el siguiente proceso:

1. Fase de preparación: El nodo **RESULTADO** envía un mensajes a los nodos patologo1 … 5, estos reciben el mensaje y realizan la escritura de la transacción, para finalizar su trabajo envían un mensaje al nodo Resultado para que él se asegure que están listos para hacer commit o abortar la transacción.

2. Fase de commit final: El nodo **RESULTADO** envía otro mensaje que lleva un commit a los demás nodos, cada uno de estos nodos realiza la actualización de la base de datos y envía al nodo Resultado un mensaje informando si pudieron realizar esa actualización.

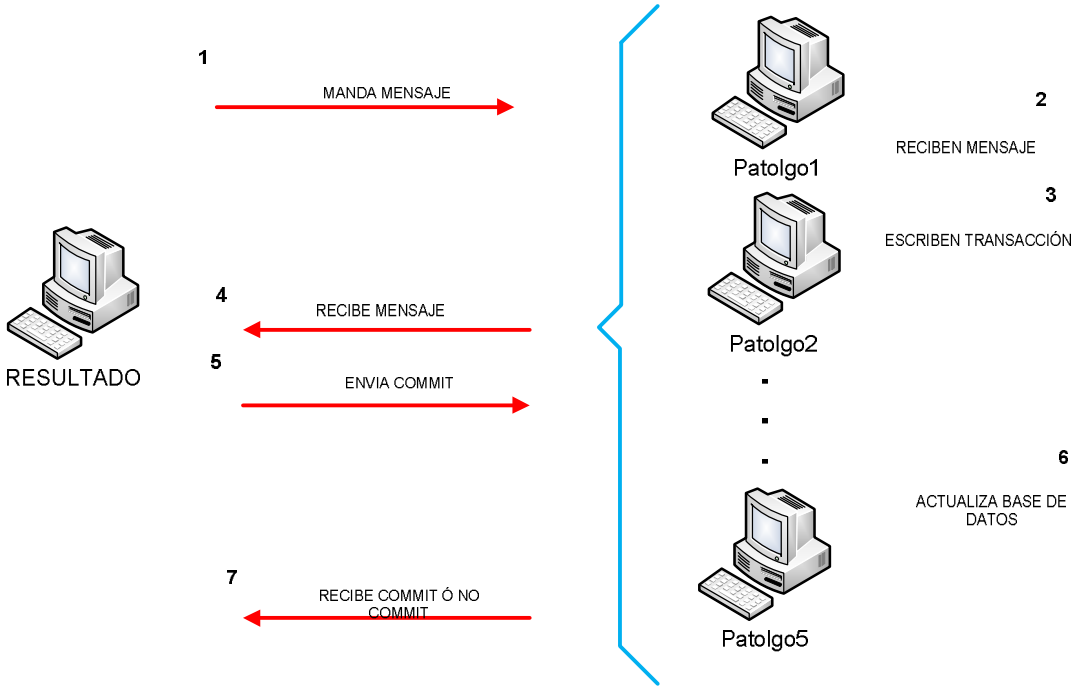


FIGURA 3.9 Diagrama del protocolo commit de dos fases.

3.1.1.4 ***FRAGMENTACIÓN Y REPLICACIÓN DE DATOS***

Al tomar las ventajas y desventajas que ofrece la fragmentación y replicación, se analiza la información requerida por cada nodo que conforma la red de comunicación inalámbrica.

- **NODO RECEPCIÓN**

En este nodo se realizará el registro de las solicitudes y biopsias que mandan los diferentes Departamentos de Servicios para su estudio. Por esta razón se establece la creación de algunos

fragmentos que serán replicados, con la finalidad de cumplir con las especificaciones necesarias para el buen funcionamiento de éste nodo, y así el personal administrativo asignado logrará entregar a tiempo el trabajo.

En primer lugar, se fragmentará horizontalmente la información que se encuentra en las entidades SOLICITUDES, SOL_CITOLOGICAS, SOL_ANATOMOPATOLOGICAS; para cumplir con los requisitos especificados por el jefe del Departamento de Servicio de Anatomía Patológica.

A continuación se hace requerencia al requerimiento que justifica la fragmentación anteriormente mencionada:

"[...]estudios de biopsias como son piezas anatómicas y citologías[...]
del tipo de solicitud que llegue, las cuales pueden ser cervico vaginales,
de diversos que son líquidos y células de descamación, generales que son
pequeñas muestras de diferentes órganos, y oncológicas[...]"

La primera fragmentación horizontal primaria se realizará en la entidad de SOL_ANATOMOPATOLOGICAS, con el propósito de obtener las solicitudes anatomopatológicas generales y oncológicas. La representación de dicha fragmentación se encuentra en la figura 3.10.

$$TIPO_SOL_ANATOMO_i = \sigma_{id_tipo_anatomo=i}(SOL_ANATOMOPATOLOGICAS)$$
Donde i={1 (general), 2 (oncológica) }

FIGURA 3.10 Representación de la fragmentación horizontal primaria de la entidad SOL_ANATOMOPATOLOGICAS.

En las tablas 3.3 y 3.4 se ejemplifican los fragmentos resultantes:

Fragmento general donde id_tipo_anatomo es igual a 1

id_anatomopatologicas	id_tipo_anatomo	num_registro	pieza_remitida	diagnostico_clinico	medico_tratante
1	1	Q-2808-09	Laminillas	Adenocarcinoma	Dra. Claudia Cano

TABLA 3.3 Ejemplo del fragmento de las solicitudes anatomopatológicas generales.

Fragmento oncológica donde id_tipo_anatomo es igual a 2

id_anatomopatologicas	id_tipo_anatomo	num_registro	pieza_remitida	diagnostico_clinico	medico_tratante
3	2	Q-O-0348-07	Estomago	Adenocarcinoma	Dr. Eduardo Vargas

TABLA 3.4 Ejemplo del fragmento de las solicitudes anatomopatológicas oncológicas.

Posteriormente los fragmentos obtenidos serán utilizados para determinar las solicitudes anatomopatolólogicas, teniendo ahora una fragmentación horizontal derivada indicada en la figura 3.11:

$$SOLICITUD_ANATOMO_i = SOLICITUDES_{id_anatomopatologicas=id_anatomopatologicas} \ltimes TIPO_SOL_ANATOMO_i$$

FIGURA 3.11 Representación de la fragmentación derivada de las solicitudes anatomopatológicas.

También se selecciona los diferentes tipos de solicitudes citológicas, aplicando distintos tipos de fragmentación.

Se lleva a cabo una fragmentación vertical para lograr separar las características de las solicitudes citológicas de diversos y cervicovaginales.

Las características de solicitud de diversos se indican en la figura 3.12.

$$SOL_CITO_DIVERSO: \Pi_{id_cito,id_muestra_pro,id_producto,id_metodo,id_unidad,clave_tipo_cito,nom_resp_medico,num_citologico}(SOL_CITOLOGICAS)$$

FIGURA 3.12 Fragmentación vertical de las solicitudes citológicas de diversos.

En la figura 3.13 se muestra un ejemplo de la fragmentación anterior:

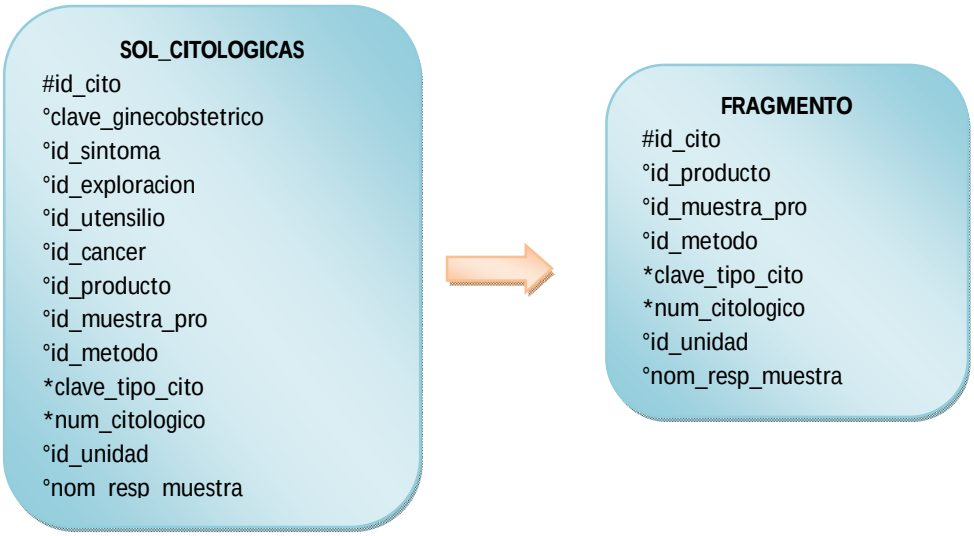


FIGURA 3.13 Ejemplo de la fragmentación vertical en las solicitudes citológicas de diversos.

Se observa en la figura 3.14 las características de solicitud cervico vaginal

$$SOL_CITO_CERVICO:\Pi_{id_cito,id_sintomas,id_exploracion,id_utensilio,id_cancer,id_ginecobtetricos,nom_resp_muestra,clave_tipo_cito,num_citologico}(SOL_CITOLOGICAS)$$

FIGURA 3.14 Características de las solicitudes citológicas cervico vaginales.

A continuación se muestra la figura 3.15 que contienen un ejemplo de la fragmentación anterior

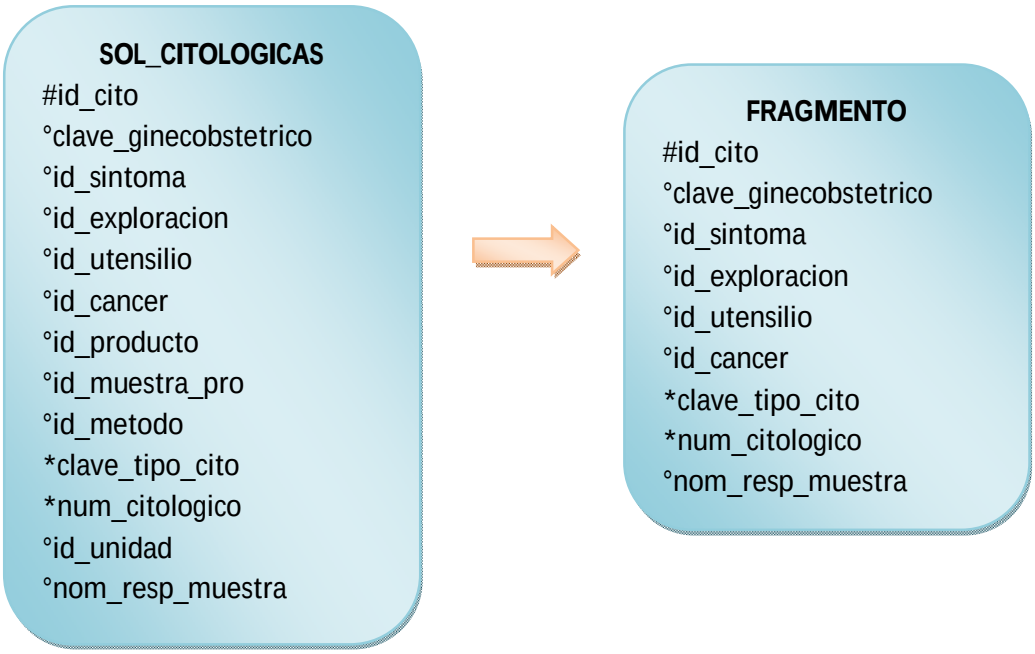


FIGURA 3.15 Ejemplo de la fragmentación vertical en las solicitudes citológicas cervico vaginales.

Hasta este paso sólo se han obtenido las características, pero falta definir los dos tipos de solicitudes citológicas, para ello se efectuará una fragmentación horizontal primaria y posteriormente una derivada, dichas fragmentaciones se indican en las figuras 3.16 y 3.17.

$$\ltimes$$

$$\textbf{A)}TIPO_DIVERSO_i = \sigma_{clave_tipo_cito=i}(SOL_CITO_DIVERSO)$$

Donde i={1 (Diversos)}

$$\ltimes$$

$$\textbf{B)}SOLICITUD_CITODIV_i = SOLICITUDES_{id_cito = id_cito}\ TIPO_DIVERSO_i$$

FIGURA 3.16 Fragmentación horizontal primaria (A) y fragmentación derivada (B) de la solicitud citológica de diversos.

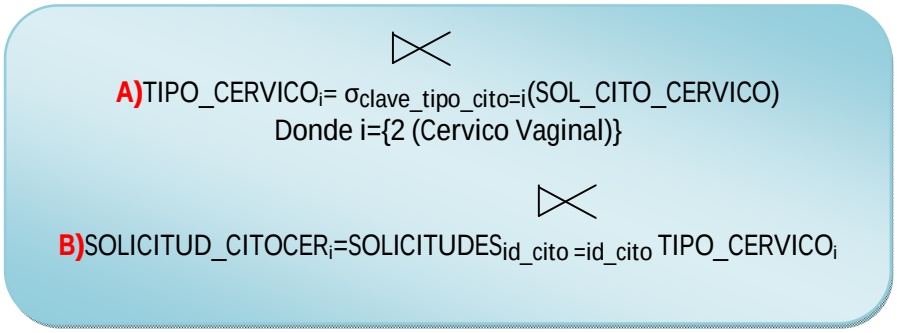


FIGURA 3.17 Fragmentación horizontal primaria (A) y fragmentación derivada (B) de la solicitud citológica cervico vaginal.

Hasta este punto la fragmentación aplicada a las entidades SOLICITUDES, SOL_CITOLOGICAS, SOL_ANATOMOPATOLOGICAS bastará para tener cubiertos todos los requerimientos de este nodo de la red. Se empleará replicación dado que tenemos catálogos relacionados con dichas entidades, como ya se ha descrito la replicación se aplicará a las entidades o catálogos que se utilizan con mayor frecuencia en operaciones de solo lectura o tengan el mínimo uso de las operaciones de escritura, actualización y borrado de datos. Por lo anterior las entidades que serán copiadas a esté nodo se muestran en la tabla 3.5:

TABLA ASOCIADA	CATÁLOGOS ó TABLAS	REPLICACIÓN
SOLICITUDES	INFORMES	SI
	SERVICIOS	SI
	TIPO_SOL_INF	SI
	PACIENTES	SI
	SOL_ANATOMOPATOLOGICAS	NO
	SOL_CITOLOGICAS	NO
SOL_CITOLOGICAS	METODOS_OBTENCION	SI
	MUESTRAS_PROCE	SI
	GINECOBSTETRICOS	SI
	PRODUCTOS	SI
	UTENSILIOS	SI
	EXPLORACIONES	SI
	SINTOMAS	SI
	CANCER	SI
	TIPO_CITOLOGICAS	SI
	SOLICITUDES	NO
SOL_ANATOMOPATOLOGICAS	TIPOS_ANATOMOPATOLOGICAS	SI
	SOLICITUDES	NO

TABLA 3.5 Entidades que serán replicadas al nodo **RECEPCIÓN**.

NODO CITOTECNÓLOGO

Este nodo tiene como función generar los informes de resultados de las muestras citológicas tanto de diversos como cervico vaginales, por esta razón, se fragmentarán las tablas INF_CITOLOGICOS e INFORMES.

Cada informe contempla ciertas características, por ello se llevará a cabo una fragmentación vertical.

En la figura 3.18 se observa el fragmento del informe citológico de diversos y en la figura 3.19 se ejemplica dicho framento.

$$INF_CITO_DIVERSO:\Pi_{id_inf_cito,muestra,descripción_microscopica,diagnostico_citologico,\ id_patologo,id_citotec,\ clave_tipo_cito}\ (INF_CITOLOGICOS)$$

FIGURA 3.18 Fragmentación vertical de los informes citológicos de diversos.

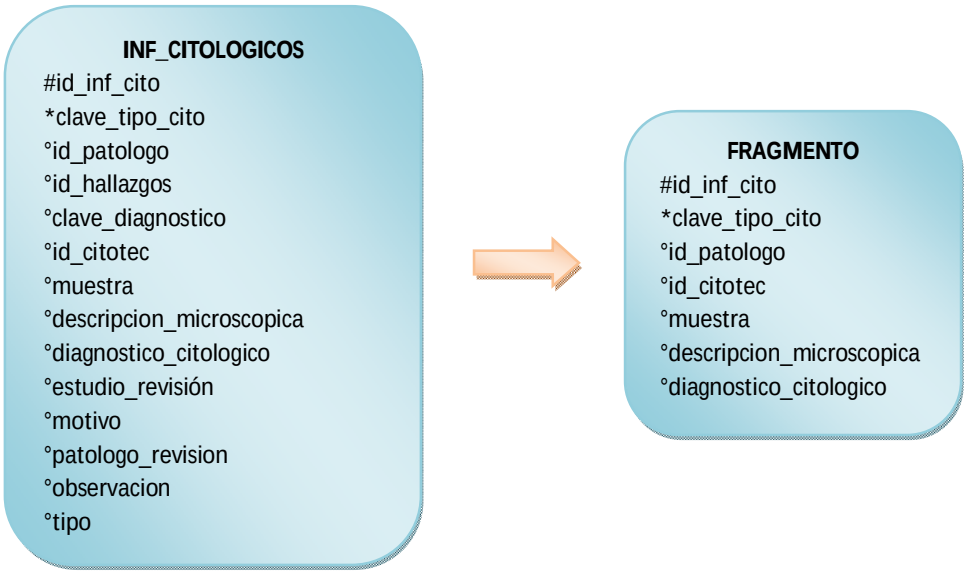


FIGURA 3.19 Ejemplo de la fragmentación vertical en los informes citológicos de diversos.

El fragmento del informe citológico cervico vaginal se encuentra representado en l a figura 3.20, así como un ejemplo de dicho fragmento se observa en la figura 3.21.

$$INF_CITO_CERVICO:\Pi_{id_inf_cito,id_patologo,id_hallazgo,clave_diagnostico,estudio_revision,\ motivo,patólogo_revision,observación,id_citotec,clave_tipo_cito,tipo}\ (INF_CITOLOGICOS)$$

FIGURA 3.20 Fragmentación vertical de los informes citológicos cervico vaginales.

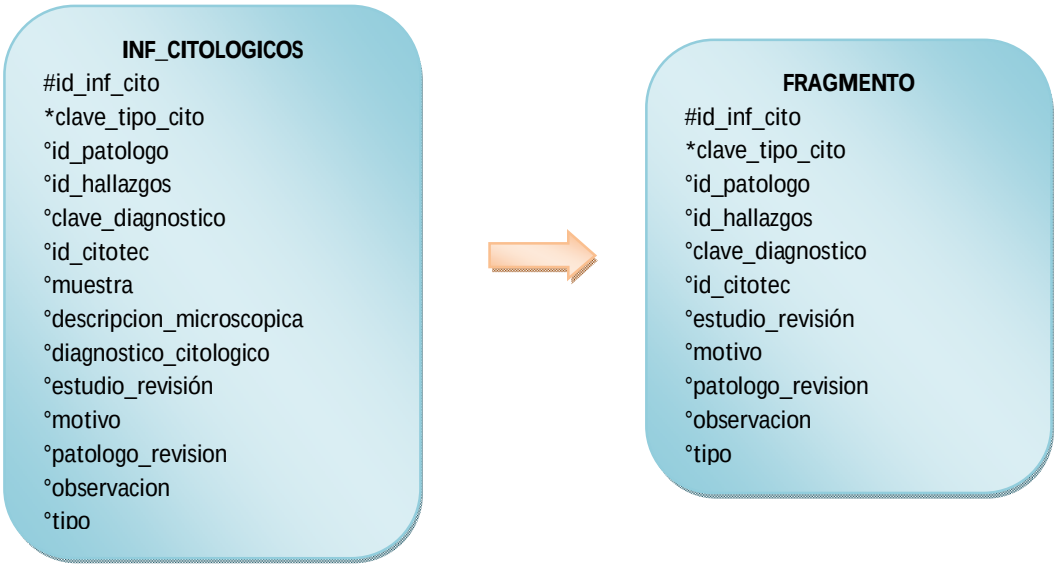


FIGURA 3.21 Ejemplo de la fragmentación vertical en los informes citológicos cervico vaginales.

Para poder dividir cada uno de los informes en su correspondiente categoría, de diversos o cervico vaginales, se fragmenta de manera primaria y derivada. Los fragmentos TIPO_DIVERSO_INF primario y INFORME _CITODIV derivado resultantes del fragmento mostrado en la figura 3.18 se muestran en la figura 3.22.

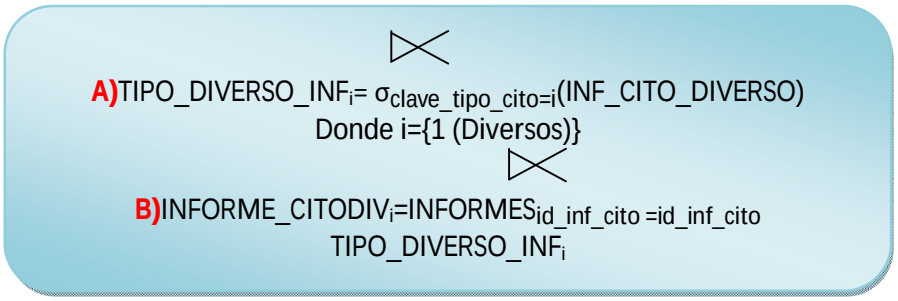


FIGURA 3.22 Fragmentación primaria (A) y Fragmentación derivada (B) de los informes citológicos de diversos.

Los fragmentos TIPO_CERVICO_INF primario y INFORME _CITOCER derivado resultantes del fragmento mostrado en la figura 3.20 se muestran en la figura 3.23.

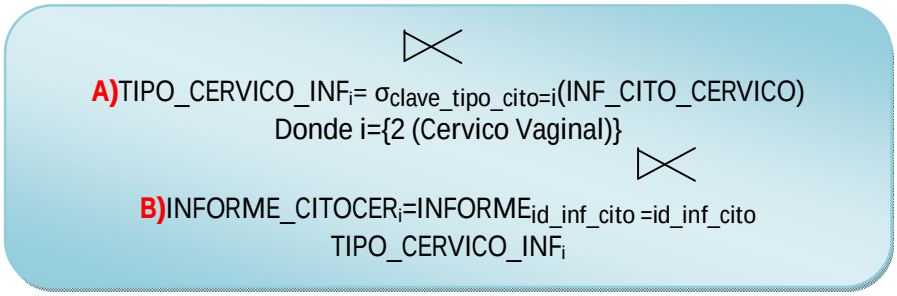


FIGURA 3.23 Fragmentación primaria (A) y Fragmentación derivada (B) de los informes citológicos cervico vaginales.

En el nodo **CITOTECNÓLOGO** se aplica la replicación de algunos catálogos y entidades que no son muy utilizadas para las operaciones de escritura, actualización ó borrado,lo cual ayuda a reducir el tiempo de acceso. Lo anterior se muestra en la tabla 3.6.

TABLA ASOCIADA	CATÁLOGOS ó TABLAS	REPLICACIÓN
INFORMES	SOLICITUDES	SI
	TIPO_SOL_INF	SI
	INF_CITOLOGICOS	NO
INF_CITOLOGICOS	TIPO_CITOLOGICAS	SI
	CITOTECNOLOGOS	SI
	DIAGNOSTICOS_CITOLOGICOS	SI
	HALLAZGOS	SI
	PATOLOGOS	SI

TABLA 3.6 Entidades que serán replicadas al nodo **CITOTECNÓLOGO**.

- **NODO ONCÓPATOLOGO Y NODO PATOLOGO (1,2,3,4,5)**

El personal que labora en estos nodos de la red, está encargado de generar los informes de resultados de las piezas anatómicas. Existe la restricción de que en los nodos **PATÓLOGO 1,2,3,4,5**, únicamente se crearán los informes de las piezas anatómicas generales. Dado que el nodo **ONCÓPATOLOGO** estará encargado de las piezas anatómicas de oncología, se determina que no es útil efectuar una fragmentación de las tablas INFORMES, INF_ANATOMOPATOLOGICOS, ya que ambos informes tienen la misma estructura final. Por lo anterior únicamente se aplicará la replicación de entidades en el nodo **PATÓLOGO 3** y los demás nodos utilizaran la información de dicho nodo, lo cual se muestra en la tabla 3.7.

TABLA ASOCIADA	CATÁLOGOS ó TABLAS	REPLICACIÓN
INFORMES	SOLICITUDES	SI
	TIPO_SOL_INF	SI
	INF_ANATOMOPATOLOGICOS	SI
INF_ANATOMOPATOLOGICOS	INFORMES	SI
	PATOLOGOS	SI

TABLA 3.7 Entidades que serán replicadas al nodo **PATÓLOGO 3**.

- **NODO RESULTADOS**

En este nodo no se aplica ni fragmentación ni replicación, ya que únicamente requiere obtener la información de los nodos **CITOTECNÓLOGO, ONCÓPATOLOGO** y **PATÓLOGO 1,2,3,4,5** para obtener los resultados de los informes y poder realizar la impresión de estos. Los informes impresos posteriormente serán llevados al Archivo de la Unidad Hospitalaria. La información se obtiene a través de transacciones y solicitudes distribuidas.

NODO JEFATURA

El nodo principal o central del sistema, tendrá el control de toda la base de datos, por esta razón se contará con todos los permisos de escritura, lectura, actualización y borrado. El nodo poseerá toda la base de datos, por ello se efectuará una replicación total a partir del servidor que aloja a la base de datos o con la unión de todos los fragmentos y replicaciones parciales que se encuentren en los diferentes nodos.

La tabla 3.8 muestra el esquema de fragmentación y ubicación del sistema del Departamento de Servicio de Anatomía Patologica.

NODO	NOMBRE_FRAGMENTO	TIPO_FRAGMENTACIÓN	CONDICIÓN	REPLICA
RECEPCIÓN	$TIPO_SOL_ANATOMO_i$	HORIZONTAL PRIMARIA	i={1 (general), 2 (oncológica) }	NO
	$SOLICITUD_ANATOMO_i$	HORIZONTAL DERIVADA	id_anatomopatologicas=id_anatomopatologicas	NO
	SOL_CITO_DIVERSO	VERTICAL	id_cito,id_muestra_pro,id_producto,id_metodo,id_unidad,clave_tipo_cito,nom_resp_medico, num_citologico	NO
	SOL_CITO_CERVICO	VERTICAL	id_cito,id_sintomas,id_exploracion,id_utensilio,id_cancer,id_ginecobtetricos,nom_resp_muestra,clave_tipo_cito,num_citologico	NO
	$TIPO_DIVERSO_i$	MIXTA (VH)	i={1 (Diversos)}	NO
	$SOLICITUD_CITODIV_i$	HORIZONTAL DERIVADA	id_cito =id_cito	NO
	$TIPO_CERVICO_i$	MIXTA (VH)	i={2 (Cervico Vaginal)}	NO
	$SOLICITUD_CITOCER_i$	HORIZONTAL DERIVADA	id_cito =id_cito	NO
CITOTECNÓLOGO	INF_CITO_DIVERSO	VERTICAL	id_inf_cito,muestra,descripción_microscopica,diagnostico_citologico, id_patologo,id_citotec, clave_tipo_cito	SI
	INF_CITO_CERVICO	VERTICAL	id_inf_cito,id_patologo,id_hallazgo,clave_diagnostico,estudio_revision,motivo,patólogo_revision,observación,id_citotec,clave_tipo_cito,tipo	SI
	$TIPO_DIVERSO_INF_i$	MIXTA (VH)	i={1 (Diversos)}	SI
	$INFORME_CITODIV_i$	HORIZONTAL DERIVADA	id_inf_cito =id_inf_cito	SI
	$TIPO_CERVICO_INF_i$	MIXTA (VH)	i={2 (Cervico Vaginal)}	SI
	$INFORME_CITOCER_i$	HORIZONTAL DERIVADA	id_inf_cito =id_inf_cito	SI
ONCÓPATOLOGO Y PATÓLOGO 1,2,3,4,5	NO HABRA FRAGMENTACIÓN, SE REPLICARAN CIERTAS ENTIDADES Y CATALOGOS EN EL NODO patólogo3			
RESULTADOS	NO HABRA NI REPLICACIÓN NI FRAGMENTACIÓN			
JEFATURA	CONTENDRA TODOS LOS FRAGMENTOS DE LOS NODOS ANTERIORES Ó HABRA UNA REPLICACIÓN TOTAL			

TABLA 3.8 Esquema de fragmentación y ubicación del sistema.

3.1.1.5 DISTRIBUCIÓN DE LOS DATOS

La figura 3.24 muestra la manera en la que se distribuye la información en todos los nodos correspondiente a la replicación y fragmentación de los datos.

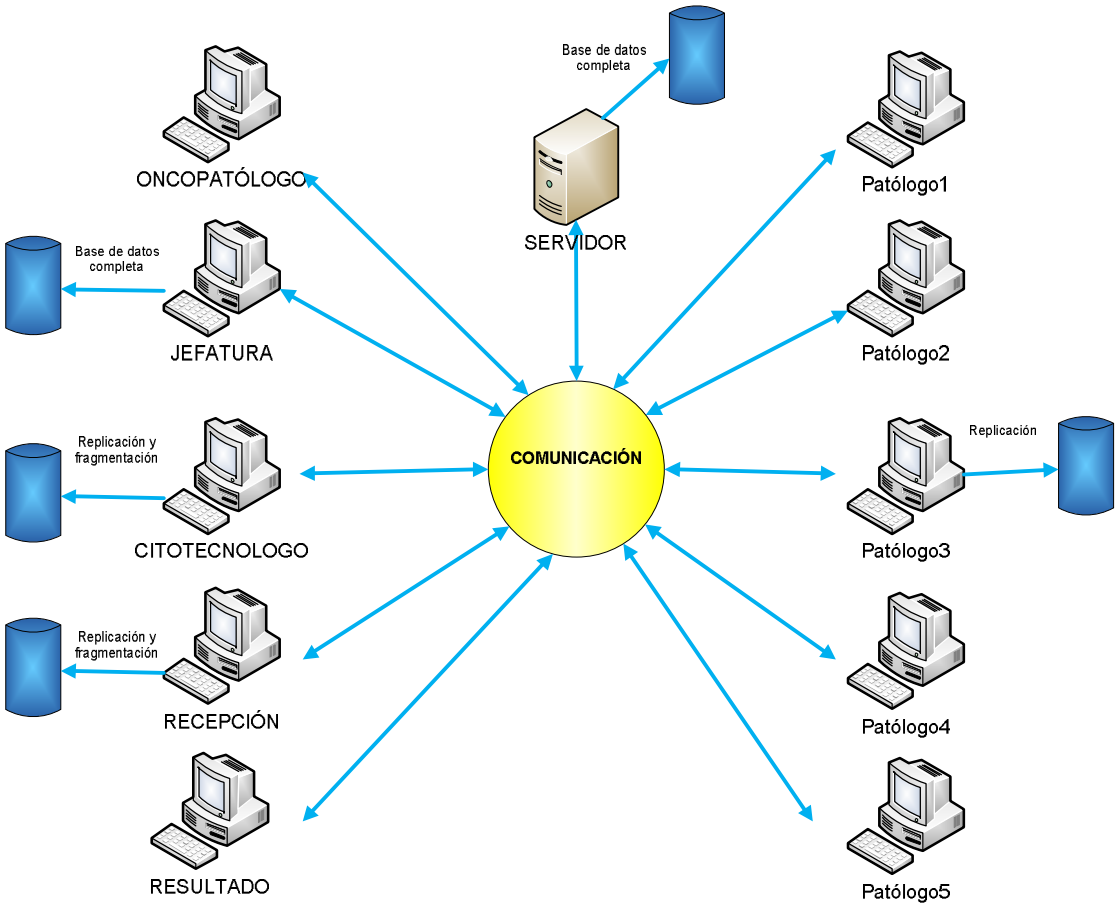


FIGURA 3.24 Diagrama de la distribución de los datos fragmentados y replicados.

3.1.2 DISEÑO LÓGICO Y FÍSICO

Después de la elección del manejador de bases de datos (veáse capítulo 1) se puede transformar el diseño conceptual a un diseño lógico para transformar las entidades en tablas y contemplar los índices, vistas, transacciones, etc. Al realizar esta transformación de diseño se tiene que contemplar las características de almacenamiento y acceso a la base de datos.

Como primer punto en la tabla 3.9 se define el espacio posible que podrá ocupar la base de datos en cinco años.

Numero de registros	Tamaño estimado KB	Objeto
100	4	Tabla cancer
300	10	Tabla característica_muestra
100	6	Tabla citotecnologos
200	6	Tabla diagnosticos_citologicos
300	10	Tabla exploraciones
27,000	3,000	Tabla ginecobstetricos
27,000	2,348	Tabla hallazgos
43,000	28,668	Tabla inf_anatomopatologicos
	172,000	Tipos de datos largos
	822	Índice inf_anatomopatologicos_pk
	822	Índice pat_inf_anato_fk
27,000	27,000	Tabla inf_citologicos
	162,000	Tipos de datos largos
	518	Índice caracteristicas_fk
	518	Índice diagnostico_fk
	518	Índice hallazgo_inf_cito_fk
	518	Índice inf_citologicos_pk
	518	Índice info_cito_citotec_fk
	518	Índice pat_inf_cito_fk
70,000	6,364	Tabla informes
	1,338	Índice inf_inf_anato_fk
	1,338	Índice inf_inf_cito_fk
	1,338	Índice inf_tipo_sol_inf_fk
	1,338	Índice informes_pk
	1,338	Índice sol_inf_fk
400	12	Tabla metodos_obtencion
400	12	Tabla muestra_proce
20,000	1,740	Tabla pacientes
300	14	Tabla patologos
400	12	Tabla productos
1	2	Tabla registro
1,000	30	Tabla servicios

300	10	Tabla síntomas
43,000	3,186	Tabla sol_anatomopatologicas
	822	Índice sol_anato_tipo_fk
	822	Índice sol_anatomopatologicas_pk
27,000	4,154	Tabla sol_citologicas
	518	Índice contenido_en_fk
	518	Índice sol_cito_can_fk
	518	Índice sol_cito_explo_fk
	518	Índice sol_cito_gineco_fk
	518	Índice sol_cito_muestra_fk
	518	Índice sol_cito_produc_fk
	518	Índice sol_cito_sin_fk
	518	Índice sol_cito_tipo_fk
	518	Índice sol_cito_uten_fk
	518	Índice sol_citologicas_pk
	518	Índice unidad_med_sol_cito_fk
70,000	8,750	Tabla solicitudes
	1,338	Índice paciente_fk
	1,338	Índice sol_servicio_fk
	1,338	Índice sol_sol_anato_fk
	1,338	Índice sol_sol_cito_fk
	1,338	Índice sol_tipo_sol_inf_fk
	1,338	Índice solicitudes_pk
4	2	Tabla tipo_anatomopatologicas
4	2	Tabla tipo_citologicas
4	2	Tabla tipo_sol_inf
27,000	1,104	Tabla unidades_medicas
400	12	Tabla utensilios
	447,790	**Total de espacio estimado**

TABLA 3.9 Tamaño aproximado de la base de datos en 5 años.

Posteriormente se creará la base de datos contemplando que primero se debe instalar el manejador de base de datos (véase Apéndice C). A continuación se crearan los objetos de la base de datos necesarios para el sistema del Departamento a través de los siguientes pasos:

1. Creación de un tablespace llamada patología

Código SQL:

```
CREATE TABLESPACE patologia
OWNER postgres
LOCATION 'C:/Proyecto';
```

Para crear el tablespace con la interface grafica pgAdmin del manejador véase las figuras 3.25, 3.26, 3.27 y 3.28.

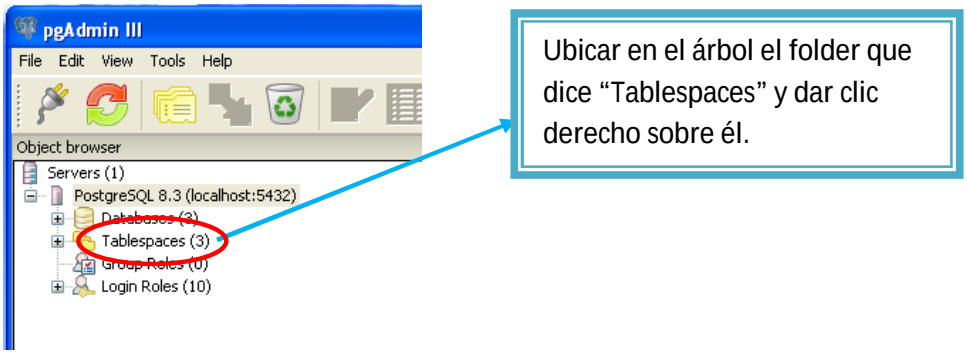


FIGURA 3.25 Selección de la pestaña "Tablespaces".

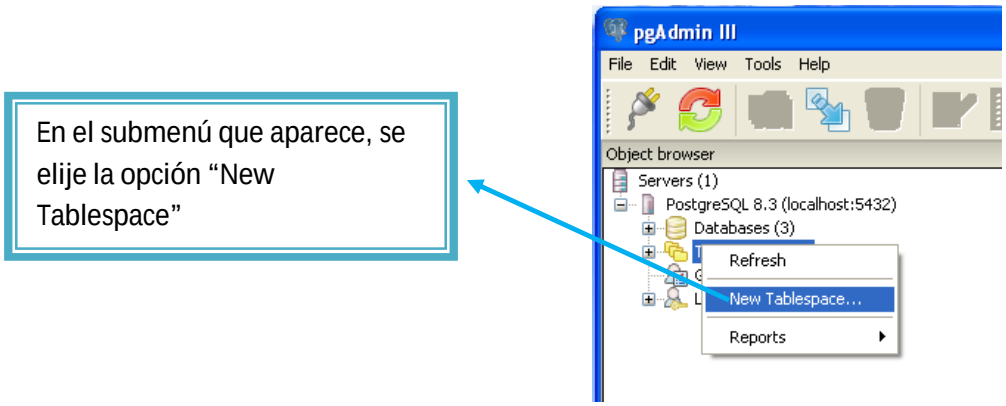


FIGURA 3.26 Creación de un nuevo tablespace.

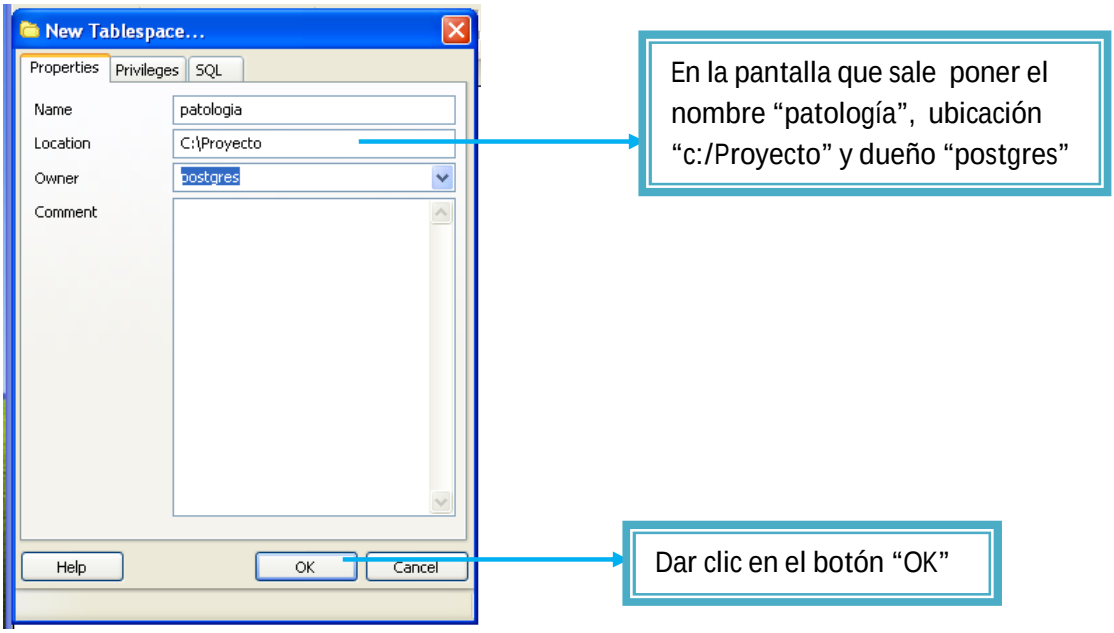


FIGURA 3.27 Configuración del tablespace.

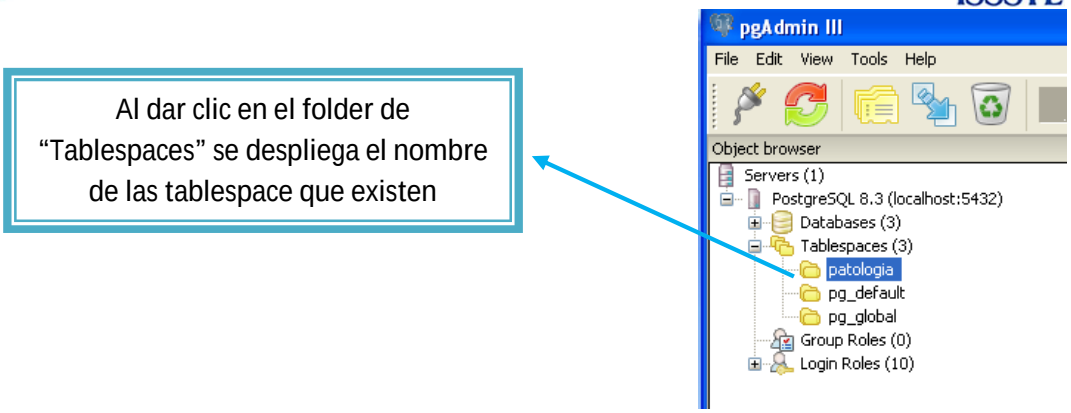


FIGURA 3.28 Visualización de tablespaces existentes.

2. Creación de la base de datos llamada "Patologia"

Código SQL

```
CREATE DATABASE "Patologia"
WITH ENCODING='UTF8'
 OWNER=postgres
TABLESPACE=patologia;
```

En las figuras 3.29, 3.30, 3.31 y 3.32 se muestra la forma de realizarlo con la interface grafica pgAdmin

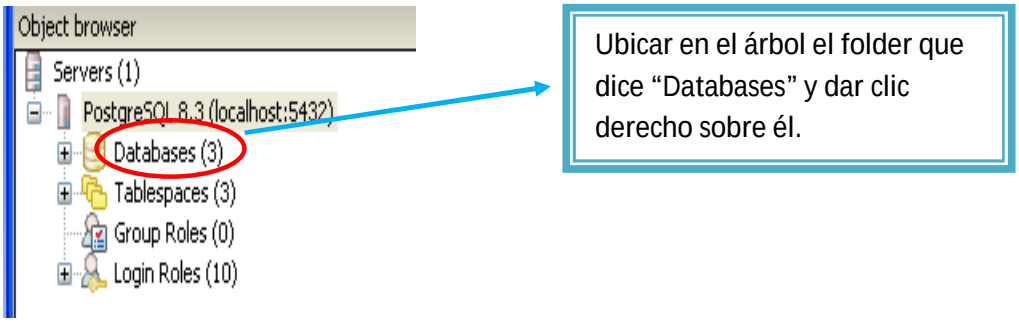


FIGURA 3.29 Selección de la pestaña "Databases".

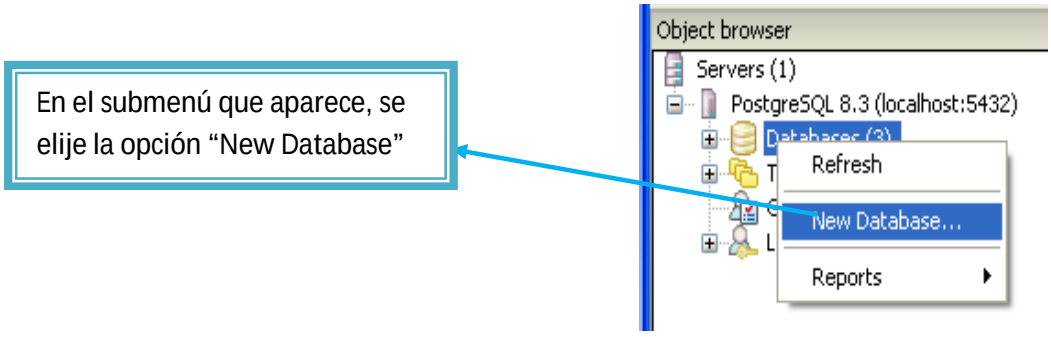


FIGURA 3.30 Creación de una nueva base de datos.

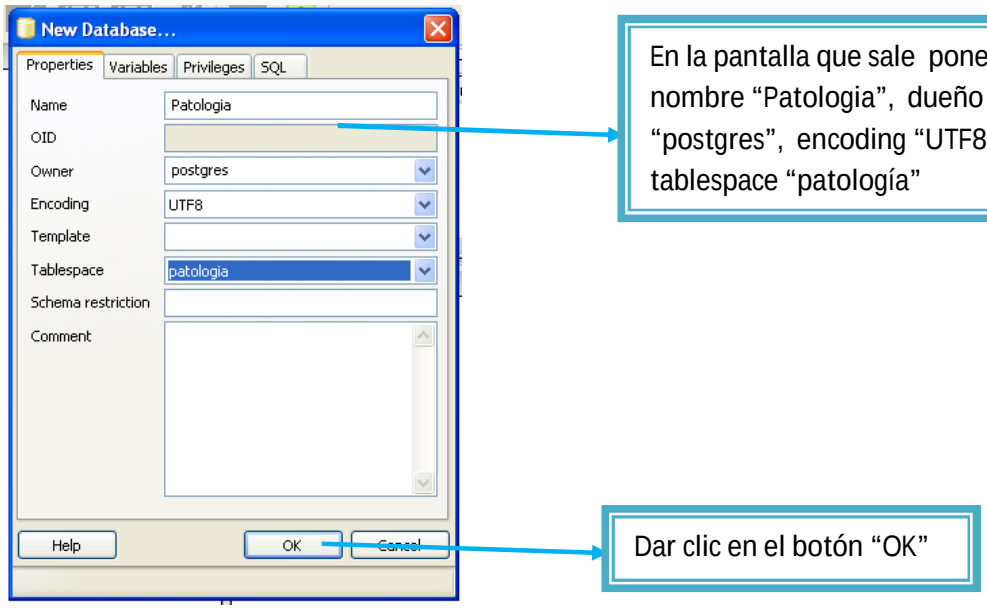


FIGURA 3.31 Configuración de la base de datos.

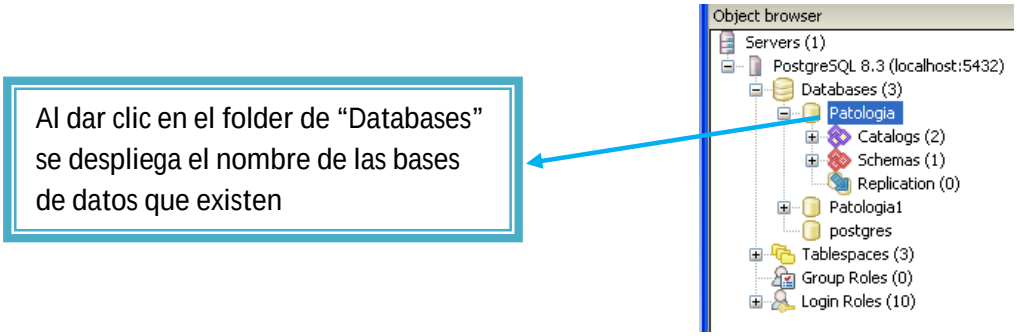


FIGURA 3.32 Visualización de las bases de datos existentes.

3. Se ejecuta el script de secuencias que serán útiles para las claves principales de las tablas para que su valor sea autoincremental.

La forma de ejecutar un script en la interface pgAdmin se muestra en la figura 3.33 y 3.34.

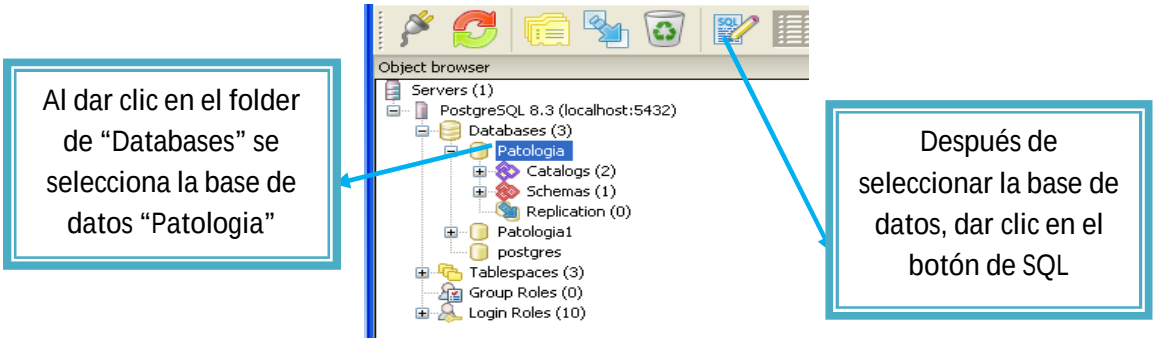


FIGURA 3.33 Selección de la opción SQL.

Se abre una nueva ventana

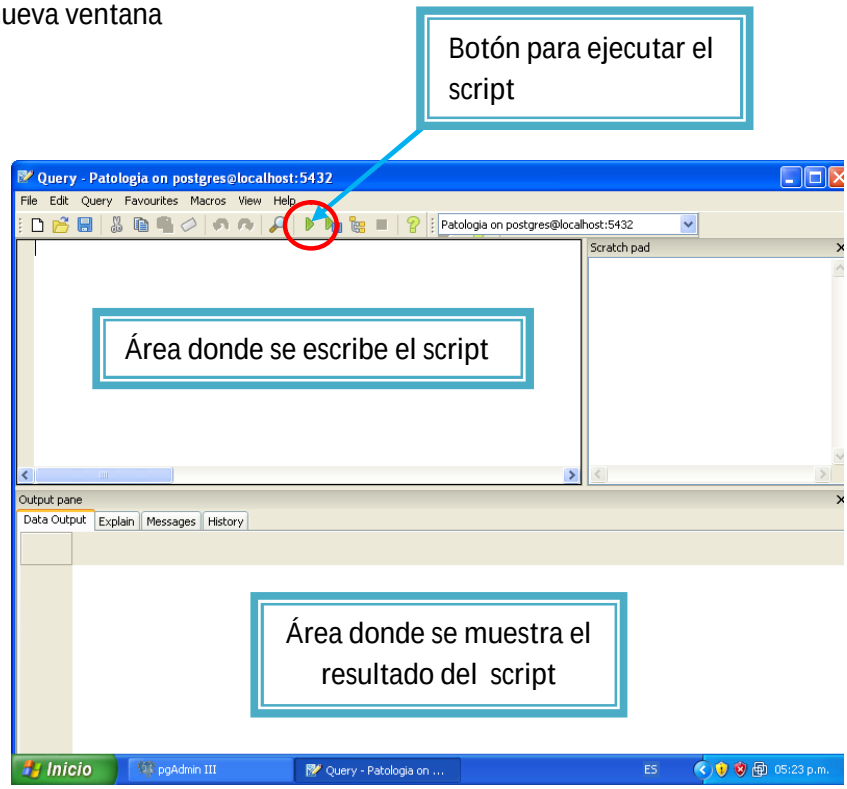


FIGURA 3.34 Ventana especial para correr los scripts.

A continuación se muestra el script para crear las secuencias de las claves principales

CREATE SEQUENCE anato_seq;
CREATE SEQUENCE cac_muestra_seq;
CREATE SEQUENCE cancer_seq;
CREATE SEQUENCE cito_seq;
CREATE SEQUENCE citolo_seq;
CREATE SEQUENCE diag_seq;
CREATE SEQUENCE explo_seq;
CREATE SEQUENCE gineco_seq;
CREATE SEQUENCE hallazgo_seq;
CREATE SEQUENCE inf_anato_seq;
CREATE SEQUENCE inf_cito_seq;
CREATE SEQUENCE informe_seq;
CREATE SEQUENCE metodo_seq;
CREATE SEQUENCE muestra_pro_seq;
CREATE SEQUENCE paciente;
CREATE SEQUENCE pato_seq;
CREATE SEQUENCE prod_seq;
CREATE SEQUENCE servicio_seq;
CREATE SEQUENCE sintomas_seq;
CREATE SEQUENCE soli_seq;
CREATE SEQUENCE tipo_seq;
CREATE SEQUENCE tipoana_seq;
CREATE SEQUENCE tipocito_seq;
CREATE SEQUENCE unidad_seq;
CREATE SEQUENCE uten_seq;

4. Por último se crean las tablas con sus respectivos atributos, posteriormente se ejecutan los insert en las tablas que actuarán como catálogos.

- A continuación se visualiza un ejemplo del script que se utiliza para la creación de una de las tablas más importantes de la base de datos del sistema del Departamento.

- **TABLA SOLICITUDES**

```
CREATE TABLE solicitudes
(
 id_solicitudes integer NOT NULL DEFAULT nextval('soli_seq'::regclass),
 clave_servicio integer,
 id_cito integer,
 id_anatomopatologica integer,
 clave_tipo integer,
 fecha_solicitud date,
 medico_responsable character varying(40),
 aclaracion boolean,
 id_paciente integer NOT NULL,
 dia character varying(20),
 statusinfo integer DEFAULT 0,
 CONSTRAINT pk_solicitudes PRIMARY KEY (id_solicitudes),
 CONSTRAINT fk_solicitu_paciente_paciente FOREIGN KEY (id_paciente)
    REFERENCES pacientes (id_paciente) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
 CONSTRAINT fk_solicitu_sol_servi_servicio FOREIGN KEY (clave_servicio)
    REFERENCES servicios (clave_servicio) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
 CONSTRAINT fk_solicitu_sol_sol_a_sol_anat FOREIGN KEY (id_anatomopatologica)
    REFERENCES sol_anatomopatologicas (id_anatomopatologica) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
 CONSTRAINT fk_solicitu_sol_sol_c_sol_cito FOREIGN KEY (id_cito)
    REFERENCES sol_citologicas (id_cito) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
 CONSTRAINT fk_solicitu_sol_tipo__tipo_sol FOREIGN KEY (clave_tipo)
    REFERENCES tipo_sol_inf (clave_tipo) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITH (OIDS=FALSE);
ALTER TABLE solicitudes OWNER TO postgres
```

- Posteriormente se muestra un ejemplo del script que contiene los insert de alguna de las tablas que fungen como catálogos

```
INSERT INTO cancer(descripcion) VALUES ('Primera vez');
INSERT INTO cancer(descripcion) VALUES ('Un año o menos');
INSERT INTO cancer(descripcion) VALUES ('2 o 3 años');
INSERT INTO cancer(descripcion) VALUES ('Más de 3 años');
INSERT INTO caracteristicas_muestra(descripcion) VALUES ('Presencia de células endocervicales');
INSERT INTO caracteristicas_muestra(descripcion) VALUES ('Metaplasia epidermoide');
INSERT INTO caracteristicas_muestra(descripcion) VALUES ('Inadecuada para el diagnóstico');
```

5. Se realiza un respaldo de la base de datos creada, él cual será utilizado en la etapa de implementación

La forma de realizar el respaldo por medio de pgAdmin se muestra en las figuras 3.35, 3.36 y 3.37.

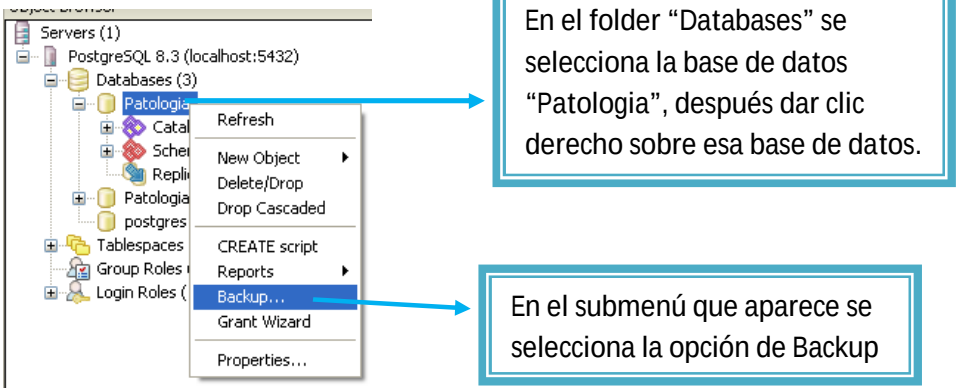


FIGURA 3.35 Selección de la opción para realizar el respaldo de la base de datos “Patologia”.

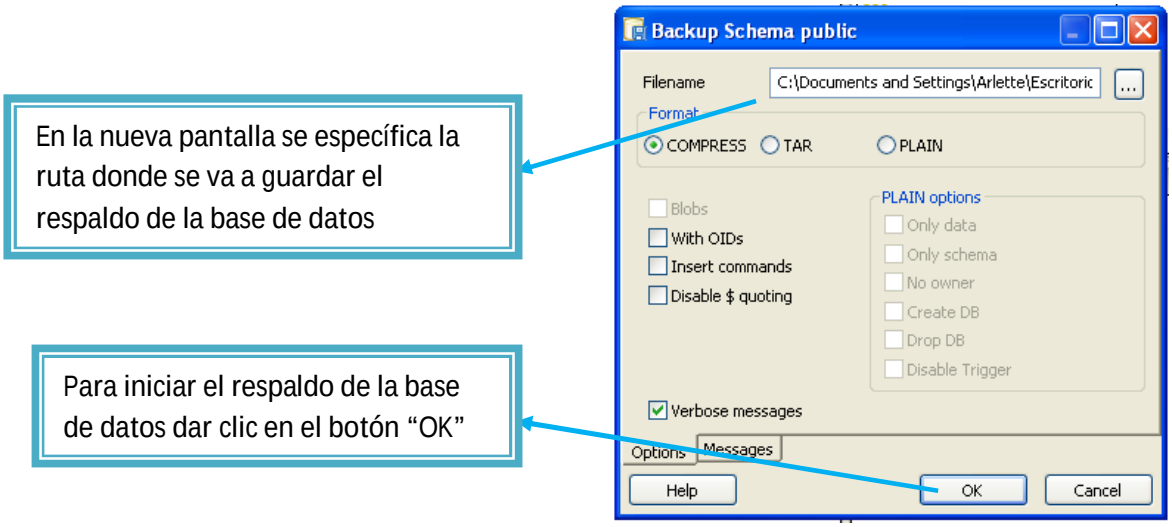


FIGURA 3.36 Configuración del respaldo de la base de datos “Patologia”.

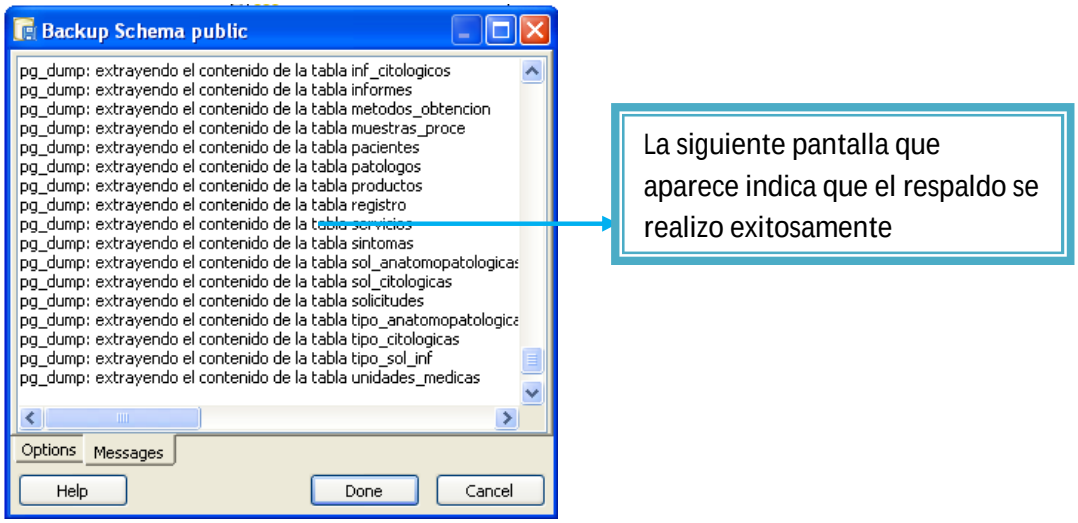


FIGURA 3.37 Creación del repaldo de la base de datos “Patologia”.

3.2 DISEÑO DE LA APLICACIÓN

El diseño de la aplicación se realiza con el lenguaje de programación JAVA por las múltiples librerías que ofrece para poder realizar una aplicación eficiente, por otro lado nos permite la conexión con varios manejadores de bases de datos.

Después de definir el diseño de la base de datos se explica la construcción de la aplicación, lo primero que se muestra es la conectividad que tendrá la aplicación con el manejador de bases de datos PostgreSQL. Se requiere como primer punto tener una pantalla (véase la figura 3.38) que valide el usuario y el password necesarios para conectarse al manejador.

FIGURA 3.38 Pantalla principal para la conexión con el manejador de la base de datos.

1. La función valida que el password tenga una longitud de 8 caracteres se muestra a continuación:

```
char passArray[]= pass.getPassword();
if(passArray.length > 8){
        System.out.println("El password tiene que ser de 8 carácteres");
}
else{
    for(j=0;j<passArray.length;j++){
            char c = passArray[j];
            if(!Character.isLetterOrDigit(c)) {
                System.out.println("Tiene algun caracter que no es letra ni numero");
            }
    }
}
```

Valida la longitud y que solo acepte caracteres

2. Después se valida el usuario para realizar la conexión con el manejador con el propósito de tener acceso a la base de datos "Patologia", cuando se logra la conexión se abre la ventana a la cual cada usuario tendrá acceso. Lo anterior se muestra en el siguiente código:

```
if (usuario.equals("DrVazquez") ){
      condicion = conexionmetodo();
      if (condicion == false){
            Ventana.setVisible(false);
            MenuJefatura intro2 = new MenuJefatura(cf, usuario);
      }
      else{
            System.out.println("No hubo conexión,verifique su password o usuario");
      }
}
```

1. Valida al usuario
2. Función que realiza la conexión
3. Abre la nueva ventana

La función que realiza la conexión utiliza la librería de JDBC para postgreSQL con el siguiente código

- URL que utiliza el Driver de JDBC para saber en dónde está localizada la base de datos que se ocupará.

  url="jdbc:postgresql://"+ip+"/Patologia";

  Donde:

  jdbc:postgresql = tipo de driver

  ip = dirección del servidor donde se encuentra la base de datos

  Patologia= base de datos

- Función para cargar el driver a utilizar

```
try{
  //Carga del driver
  Class.forName("org.postgresql.Driver");
}catch(ClassNotFoundException e){
  e.printStackTrace();
}
```

- Crear la conexión con la base de datos incluyendo la url, el usuario y el password.

```
con =  DriverManager.getConnection(url,usuario,password);
```

El sistema ***SIDIPAT*** contará con las siguientes opciones de funcionalidad mostradas en la figura 3.39.

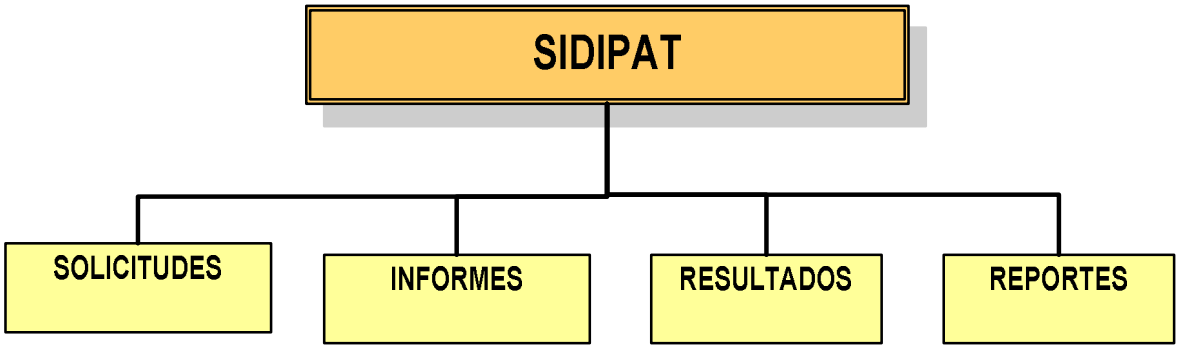


FIGURA 3.39 Opciones de funcionalidad del sistema ***SIDIPAT***.

Cada una de las funcionalidades tiene una pantalla relacionada que a continuación será descrita.

- **PANTALLAS DE SOLICITUDES**

La pantalla de **SOLICITUDES** cuanta con ciertas funciones que ayudan a realizar el registro de las solicitudes, dichas funciones se muestran en la figura 3.40.

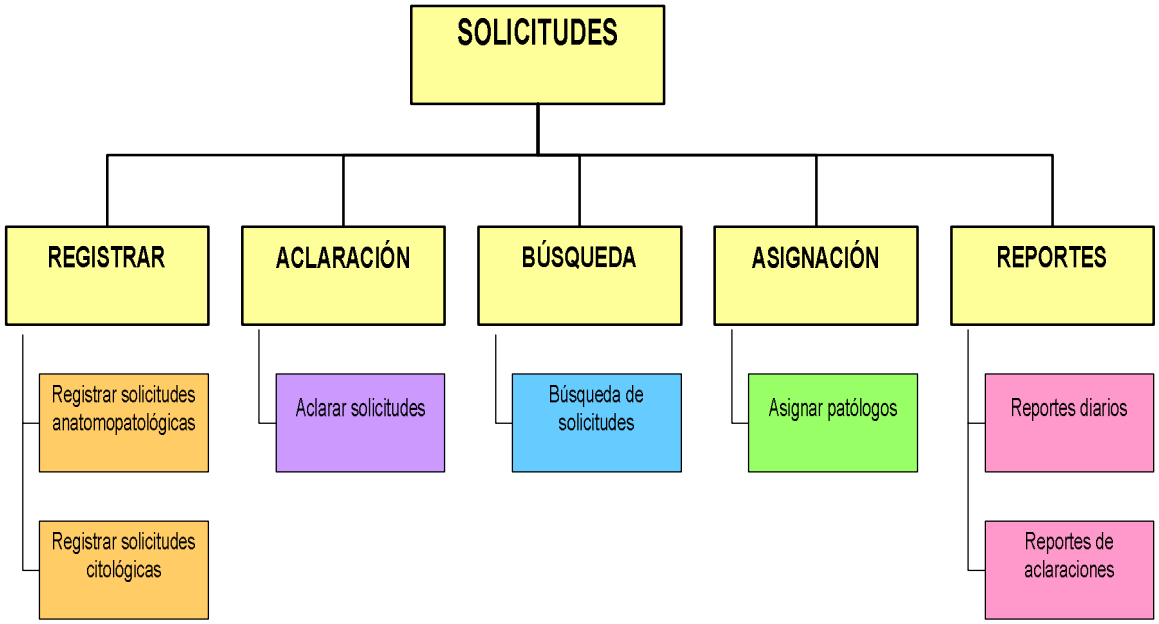


FIGURA 3.40 Diagrama jerárquico funcional de la pantalla **SOLICITUDES**.

La pantalla que contiene las funciones descritas en el diagrama jerárquico funcional de la figura 3.40 se observa en la figura 3.41:

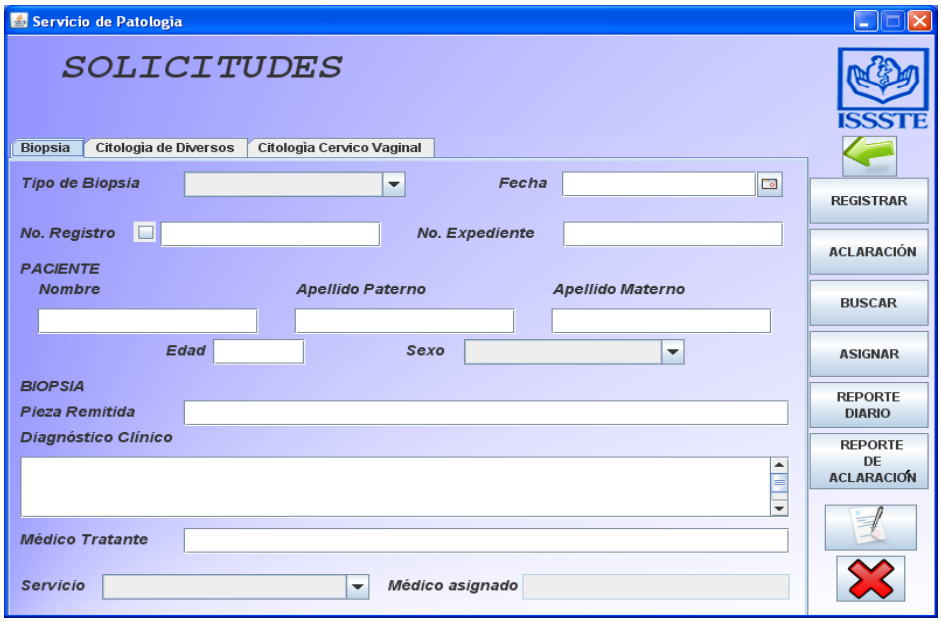


FIGURA 3.41 Pantalla de **SOLICITUDES**.

A continuación se presenta el diagrama de flujo de datos de cada una de las funciones de solicitudes.

- **REGISTRAR**

Esta función se codifica para realizar el registro de las solicitudes en la base de datos, obteniendo la información de los formularios. En la figura 3.42 se muestra el diagrama de flujo de datos de dicha función.

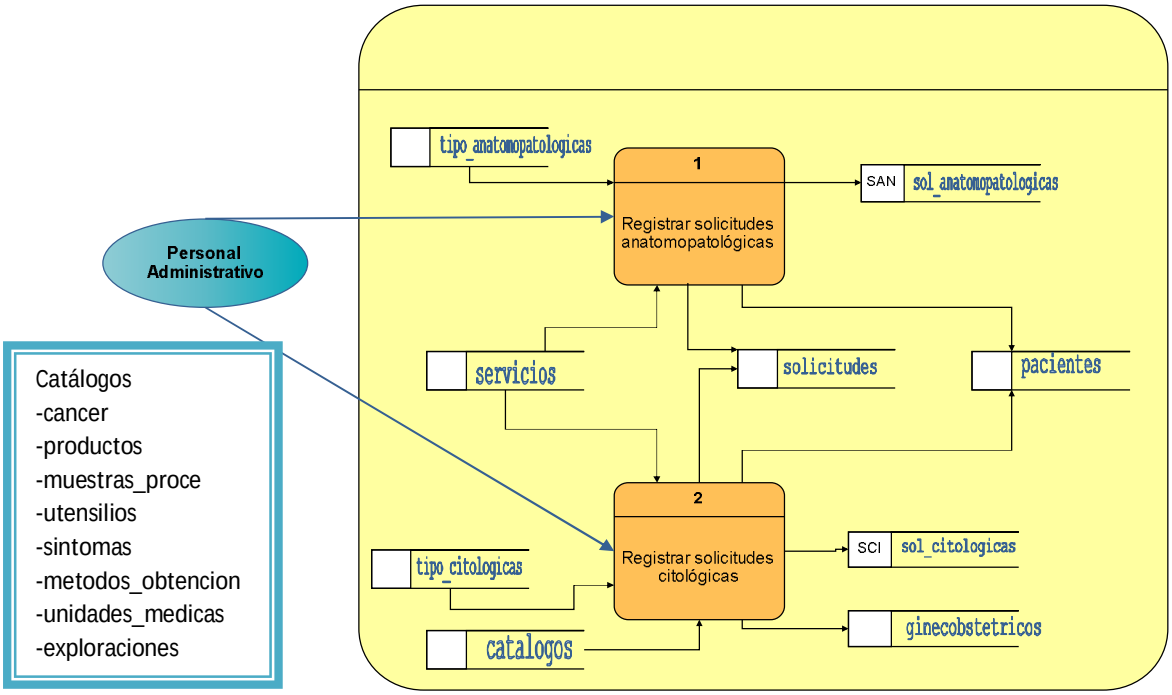


FIGURA 3.42 Diagrama de flujo de datos de la función **REGISTRAR**.

- **ACLARACIÓN**

En la figura 3.43 se muestra el diagrama de flujo de datos de esta función que servirá para marcar una solicitud que requiere ser aclarada.

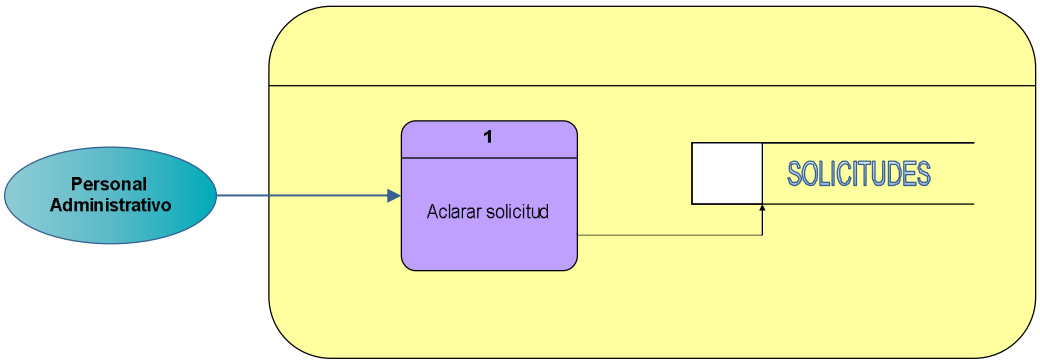


FIGURA 3.43 Diagrama de flujo de datos de la función **ACLARACIÓN.**

- **BÚSQUEDA**

Los métodos que se codifican para esta función sirven para encontrar las solicitudes que ya fueron registradas de acuerdo a ciertas condiciones. Esta función tiene una pantalla especial mostrada en la figura 3.44.

FIGURA 3.44 Pantalla de la función **BÚSQUEDA**.

La pantalla de búsqueda tiene una tabla dinámica donde se muestran los resultados encontrados, las columnas de la tabla varían de acuerdo al tipo de estudio que se seleccione, dado que cada estudio tiene información específica.

- **panel.remove(tb1);** con esta línea de código se remueve la tabla existente.
- Se crea una nueva tabla asignándole las columnas y filas que tendrá cada tipo de solicitud, por último se agregar al panel para que sea repintada.

```
tb1= new JTable(filas,columnas);
tb1.setFillsViewportHeight(true);
 panel.setViewportView(tb1);
 panel2.add(panel);
```

El diagrama de flujo de datos para la función **BÚSQUEDA** se muestra en la figura 3.45.

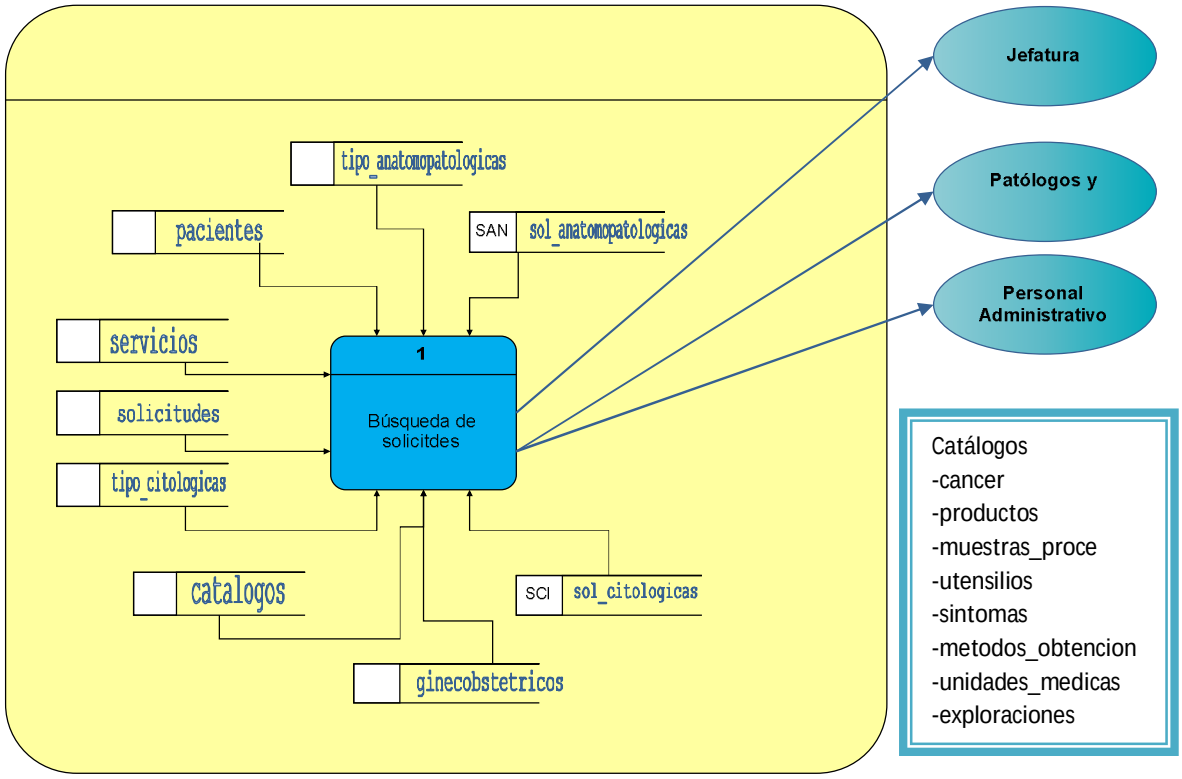


FIGURA 3.45 Diagrama de flujo de datos de la función **BÚSQUEDA.**

- **ASIGNACIÓN**

Con esta función se asigna a cada patólogo una solicitud, el método lo hará de manera equitativa y aleatoria. En la figura 3.46 se observa el diagrama de flujo de dicha función.

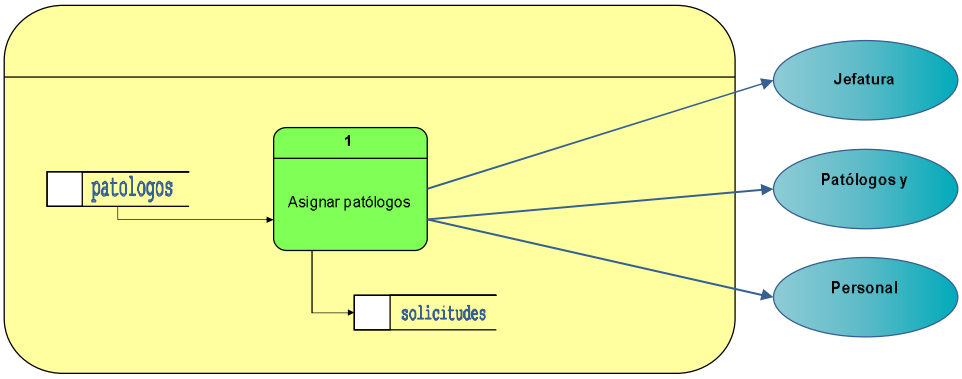


FIGURA 3.46 Diagrama de flujo de datos de la función **ASIGNACIÓN.**

- Consultas requeridas para obtener la información necesaria para realizar la asignación de solicitud a cada patólogo.

```
//Obtiene el número del último patólogo asignado
querry1="select medico from registro where id =1";

//Número total de patólogos que están activos
querry2="select count(*) as cuenta from patologos where status =1";

//Nombre de los patólogos que están activos
 querry3="select nombre1 from patologos where status =1 ";

//Actualización de la tabla solicitudes
querry4="UPDATE solicitudes set medico_responsable= ? where id_solicitudes="+id_sol;

//Actualización de la tabla registro, guarda el valor del último medico asignado
querry5="UPDATE registro set medico=? where id =1";
```

- Para ejecutar las consultas a través de código JAVA se realiza el siguiente procedimiento:

1. Guarda en una variable tipo entero el valor del último patólogo asignado.

```
s1=cf.createStatement();
 rs1=s1.executeQuery(querry1);
 while(rs1.next()){
        id=rs1.getInt("medico");
}
```

2. Guarda el valor del total de patólogos activos en una variable tipo entero.

```
s2=cf.createStatement();
rs2=s2.executeQuery(querry2);
while(rs2.next()){
        num=rs2.getInt("cuenta");
}
```

3. Guarda el nombre de los patólogos activos en una lista de arreglos de tipo String.

```
s3=cf.createStatement();
rs3=s3.executeQuery(querry3);
for(int l=0;l<num;l++){
    while(rs3.next()){
        pat.add(rs3.getString("nombre1"));
    }
}
```

4. Es un ciclo que ejecuta la actualización de la tabla **SOLICITUDES** para colocar el nombre del patólogo responsable. Al terminar de actualizar dicha tabla guarda en la tabla **REGISTRO** el valor del patólogo que acaba de ser asignado.

```
ps1=cf.prepareStatement(querry4);
ps1.setString(1,pat.get(id));
ps1.executeUpdate();
medresp.setText(pat.get(id));
ps2=cf.prepareStatement(querry5);
int w=id +1;
ps2.setInt(1,w);
ps2.executeUpdate();
```

- **REPORTES**

Esta función genera el reporte de aclaraciones y el reporte diario. El diagrama de flujo de datos se muestra en la figura 3.47.

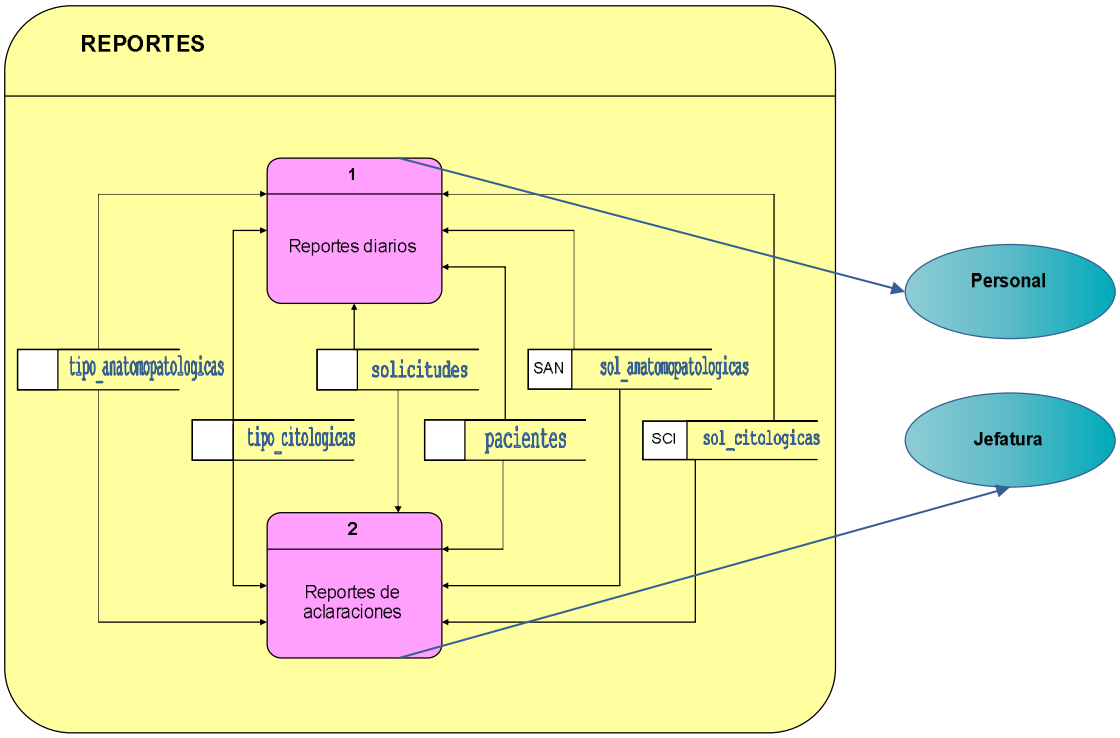


FIGURA 3.47 Diagrama de flujo de datos de la función **REPORTES**.

El diseño de las hojas de dichos reportes se muestra en la figura 3.48 y 3.49:

- **Reporte Diario**

Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado

HOSPITAL REGIONAL 1° DE OCTUBRE

Subdirección General Médica

Reporte Diario de Solicitudes

Fecha_Solicitud	Nombre del Paciente	Expediente	No. Registro	Médico Asignado
2010-08-12	Armando Ruiz Malcon	ASDE134678-4	Q-8-10	Dr. Ulises Popoca Barriga

FIGURA 3.48 Hoja para el reporte diario de solicitudes.

- **Reporte de aclaraciones**

Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado

HOSPITAL REGIONAL 1° DE OCTUBRE

Subdirección General Médica

Aclaraciones

Reporte Diario de Solicitudes

Fecha_Solicitud	Nombre del Paciente	Expediente	No. Registro	Médico Asignado

FIGURA 3.49 Hoja para el reporte de aclaraciones de las solicitudes.

- **PANTALLAS DE INFORMES**

Para ingresar a informes primero se debe interactuar con una pantalla que hará el filtro de las solicitudes que no tengas un informe relacionado. La pantalla mostrará la información de la solicitud a la cual se desea realizar su informe.

Su funcionalidad se describe en la figura 3.50 que se muestra a continuación:

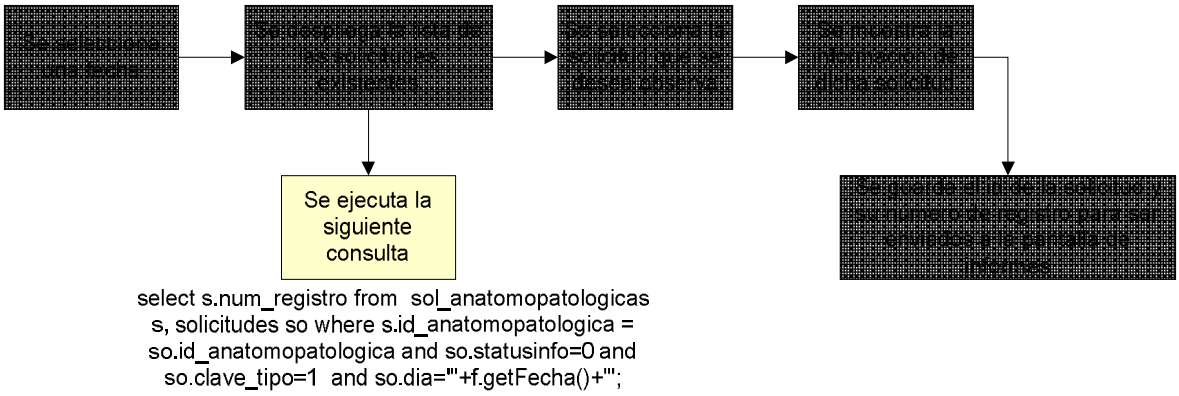


FIGURA 3.50 Procedimiento para realizar el filtro de solicitudes.

Se crearan dos tipos de pantallas dependiendo del tipo de informe que se realice, la única diferencia es el número de pestañas, dado que el informe citológico tiene un formato diferente para las citologías de diversos y las cervico vaginales. La funcionalidad de ambas pantallas es la misma únicamente cambia el diseño. A continuación se muestras las pantallas a utilizar en las figuras 3.51 y 3.52.

- PANTALLA DE FILTRO DE SOLICITUDES PARA INFORMES ANATOMOPATOLÓGICOS

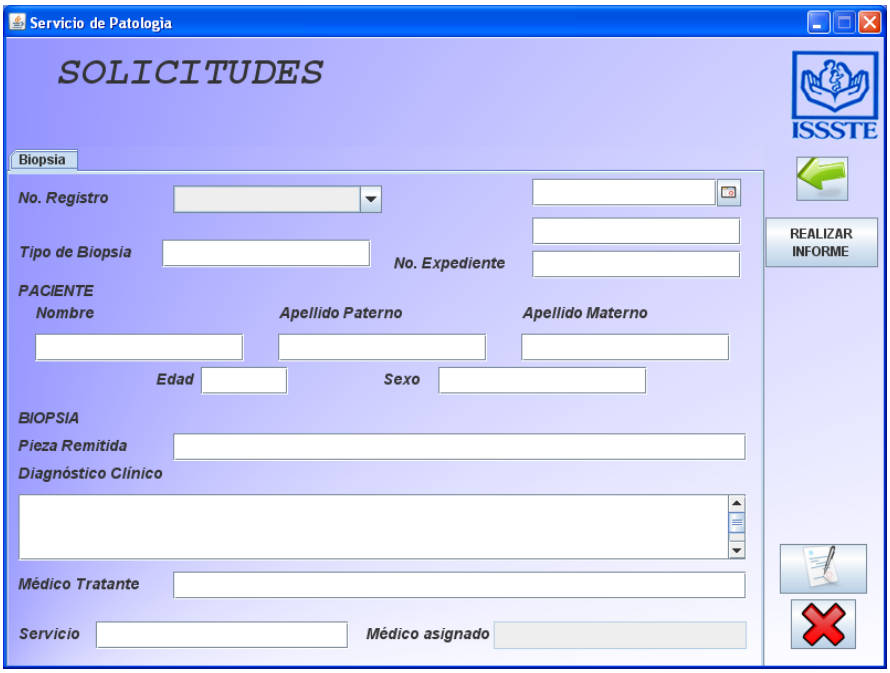


FIGURA 3.51 Pantalla que sirva para realizar el filtro de las solicitudes para informes anatomopatológicos.

- PANTALLA DE FILTRO DE SOLICITUDES PARA INFORMES CITOLÓGICOS

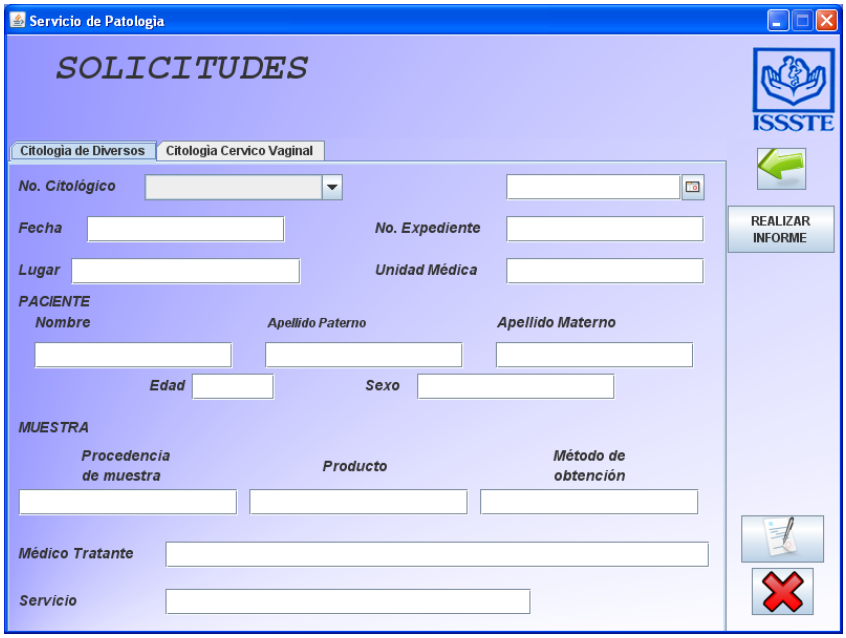


FIGURA 3.52 Pantalla que sirva para realizar el filtro de las solicitudes para informes anatomopatológicos

La pantalla **INFORMES** también tiene dos diseños de pantalla dependiendo del tipo de informe que se realizará (véase las figuras 3.53 y 3.54), la funcionalidad no se afecta, ambas pantallas realizan las acciones programadas sin ningún problema.

- PANTALLA INFORMES ANATOMOPATOLÓGICOS

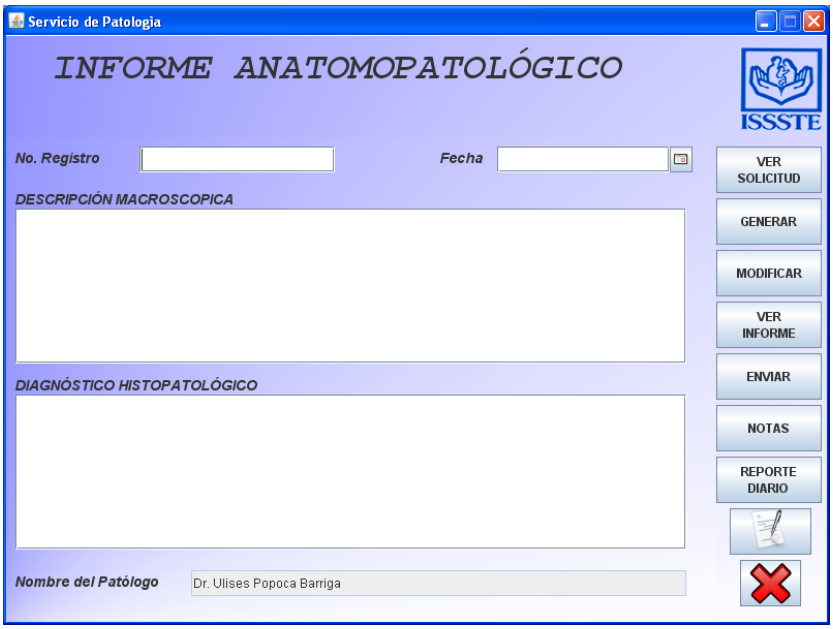


FIGURA 3.53 Pantalla **INFORMES** de tipo anatomopatológicos.

- PANTALLA INFORMES CITOLÓGICOS

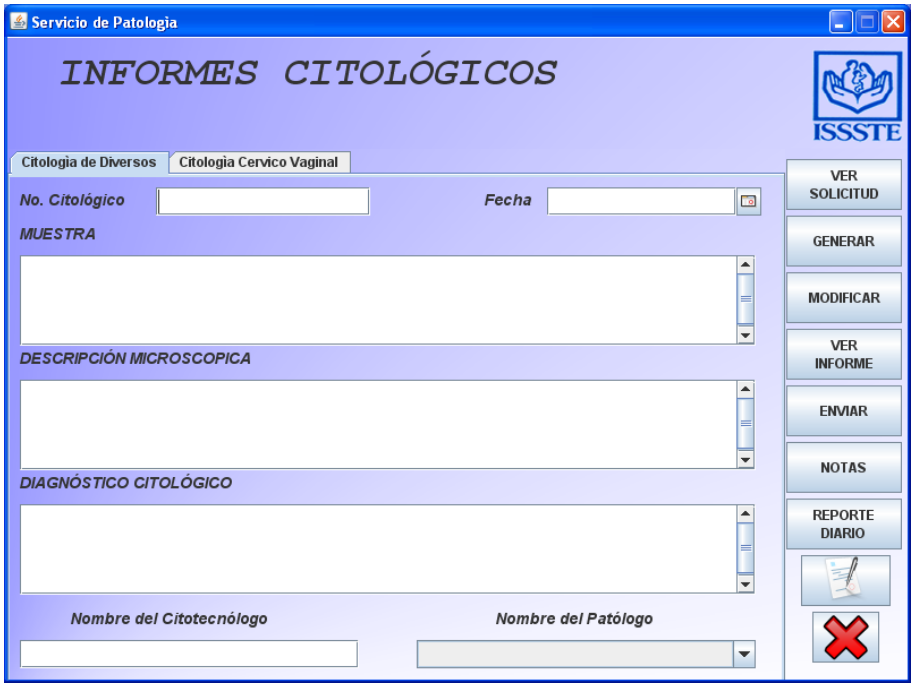


FIGURA 3.54 Pantalla **INFORMES** de tipo citológicos.

Las acciones que ejecutarán las pantallas anteriores se describen en la figura 3.55:

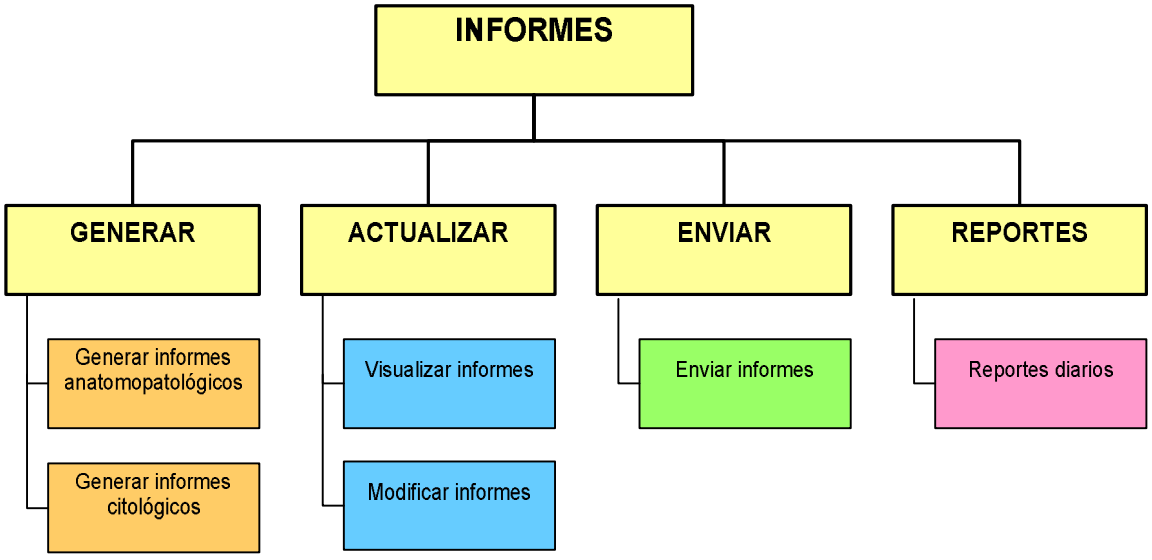


FIGURA 3.55 Diagrama jerárquico funcional de la pantalla **INFORMES**.

- **GENERAR**

Tiene la funcionalidad de guardar los datos de los informes en la base de datos. A continuación se muestra el diagrama de flujo de datos en la figura 3.56.

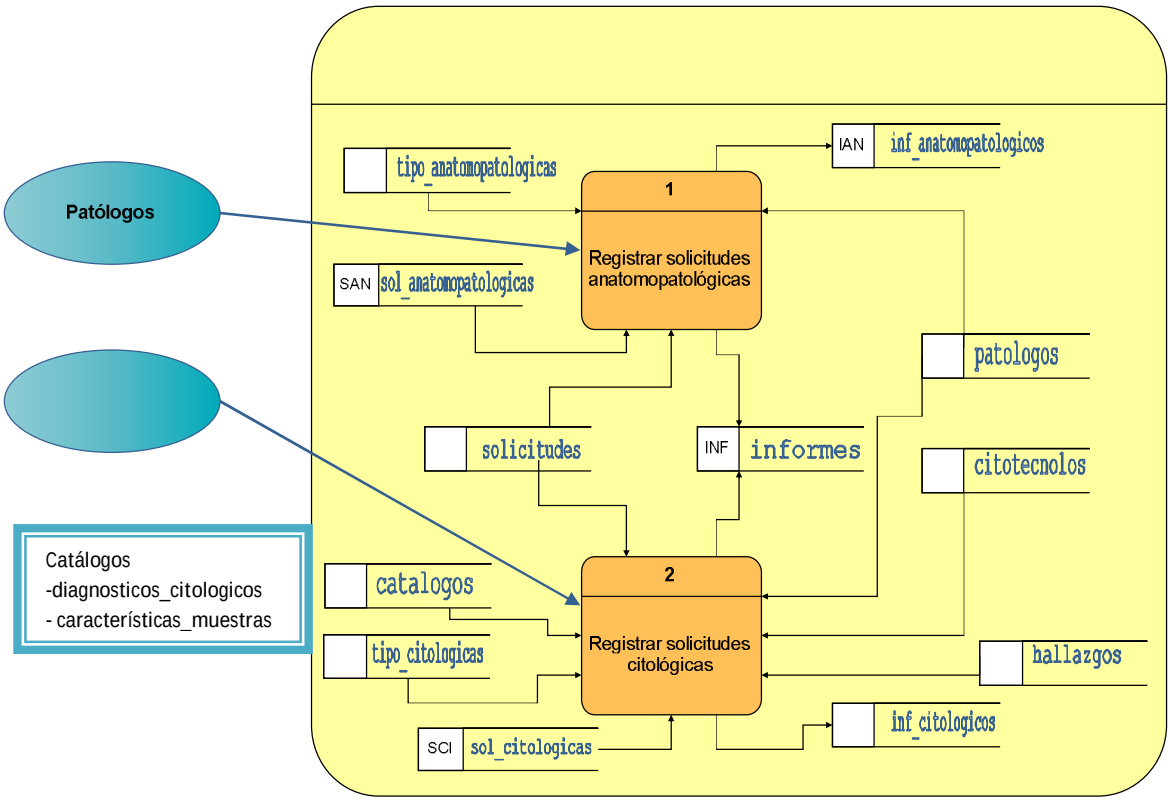


FIGURA 3.56 Diagrama de flujo de datos de la función **GENERAR**.

- **ACTUALIZAR**

La función que realiza es de obtener los datos de la base de datos de un informe que ya este creado y mostrarlo al usuario por si desea realizar alguna modificación. Si se realiza algún cambio lo guarda en la base de datos. La acción de modificar únicamente es válida si el informe tiene un estado de pendiente.

El diagrama de flujo de datos de dicha función se muestra en la figura 3.57.

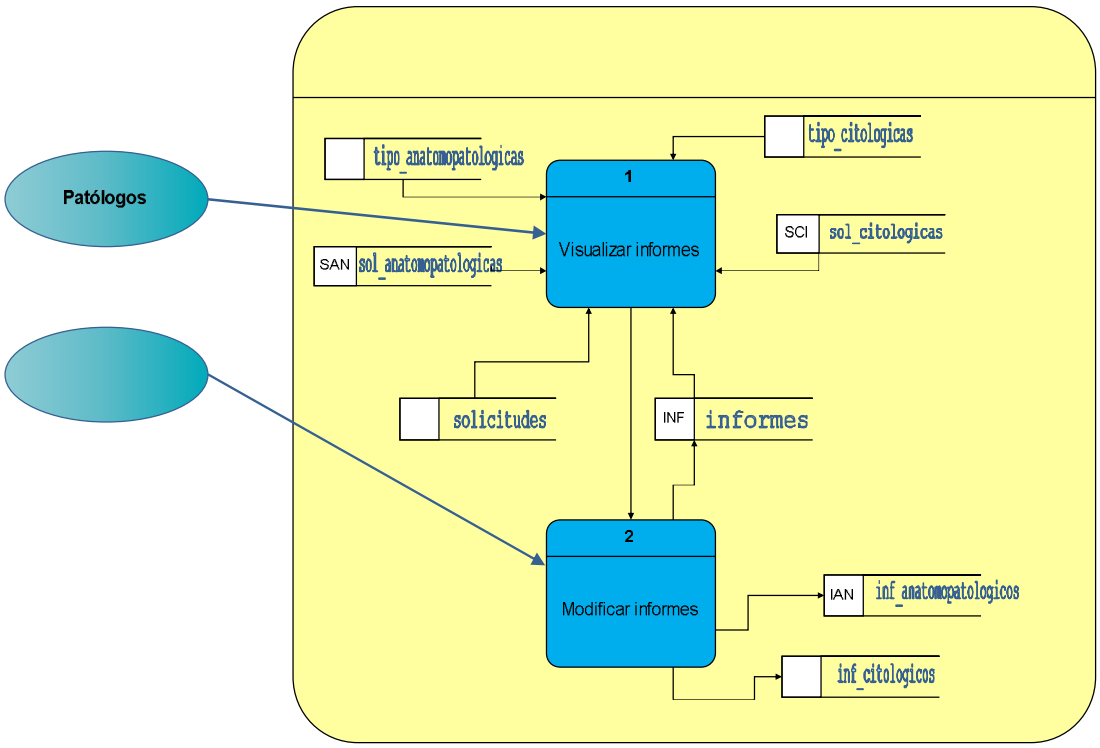


FIGURA 3.57 Diagrama de flujo de datos de la función **ACTUALIZAR**.

- **ENVIAR**

Cambia el estado del informe de pendiente a finalizado, esto servirá para llevar un control de la productividad que se tiene en el Departamento de Servicio. Al cambiar el estado de un informe a finalizado los patólogos y citotecnólogos no podrán realizar ninguna modificación, se pedirá autorización al administrador, quien podrá cambiar el estado del informe. En la figura 3.58 se observa el diagrama de flujo de datos de la función **ENVIAR**.

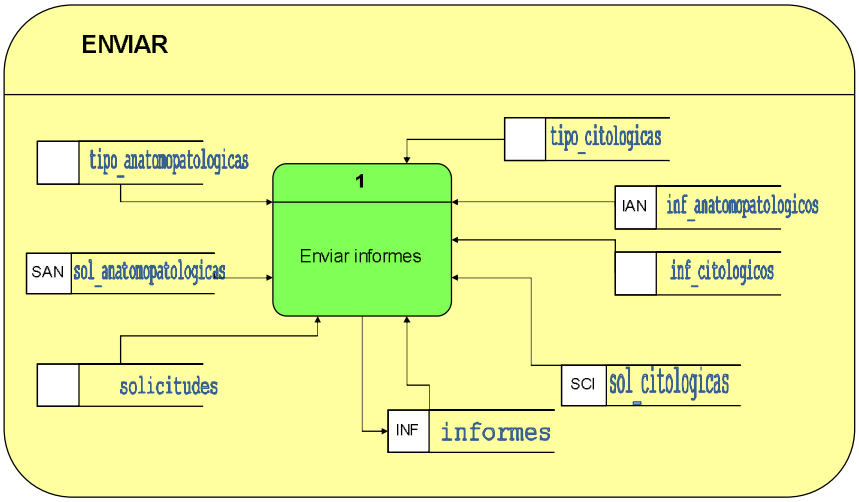


FIGURA 3.58 Diagrama de flujo de datos de la función **ENVIAR**.

REPORTES

Genera un concentrado diario de los informes que fueron realizados sin importar el estado que tengan. Para entender mejor su funcionamiento se crea un diagrama de flujo de datos mostrado en la figura 3.59.

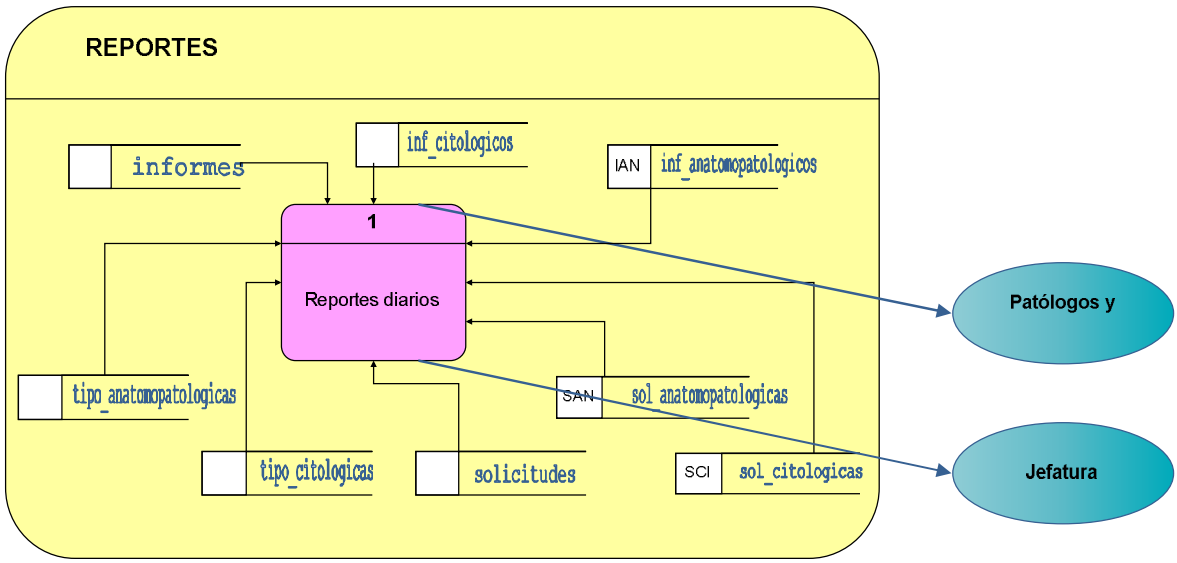


FIGURA 3.59 Diagrama de flujo de datos de la función **REPORTES**.

El formato de dicho informe se muestra en la figura 3.60:

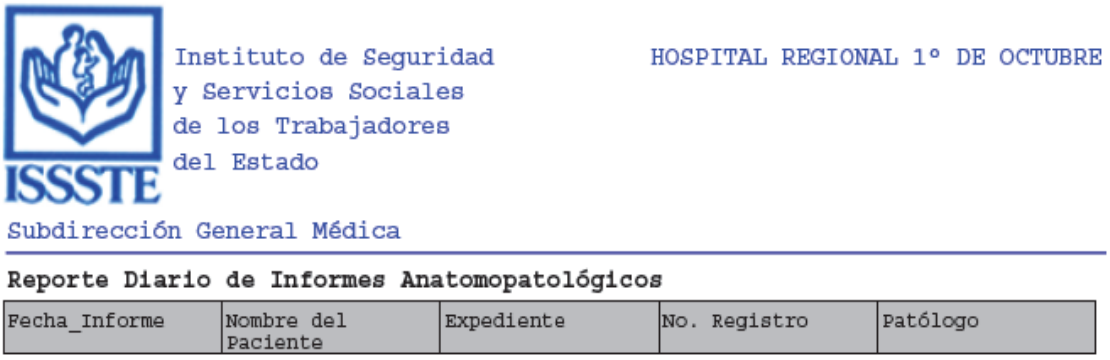


FIGURA 3.60 Formato de la hoja de reportes de informes.

La pantalla **INFORMES** tiene un botón **NOTAS** que abre una nueva pantalla llamada **NOTAS,** en la que el médico puede realizar sus anotaciones personales con la capacidad de abrir cualquier archivo .txt y guardar en el mismo formato. Dicha pantalla **NOTAS** se muestra en la figura 3.61, así como los menus que contempla se observan el la figura 3.62 y 3.63.

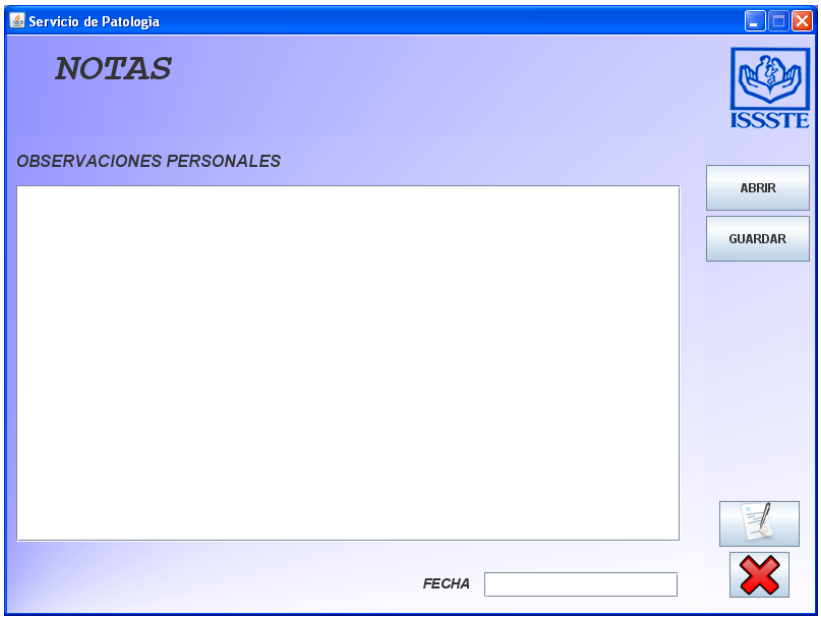


FIGURA 3.61 Pantalla **NOTAS**.

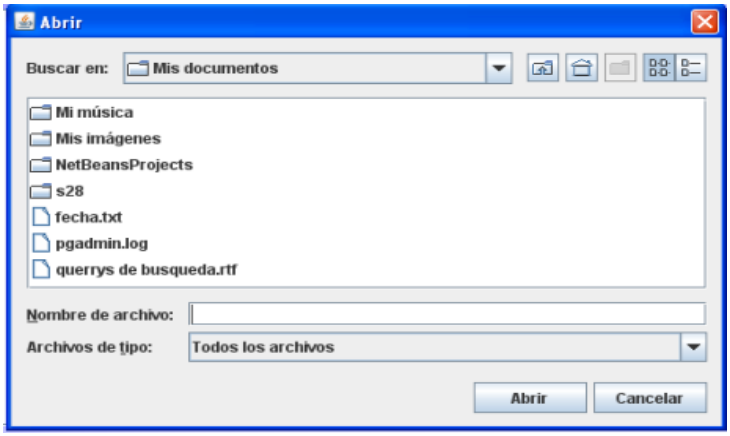


FIGURA 3.62 Menú Abrir de la pantalla **NOTAS**.

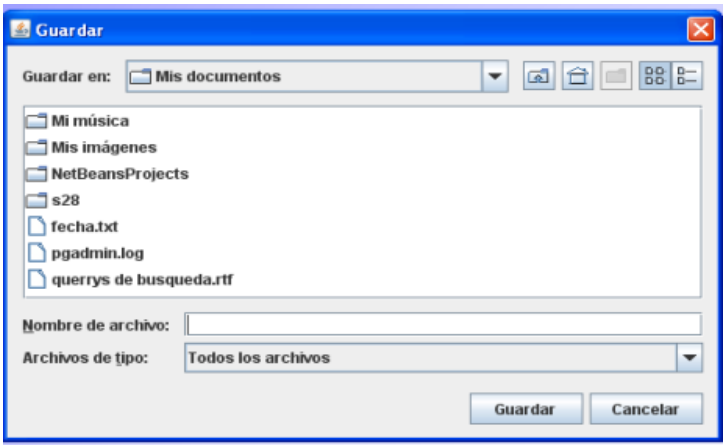


FIGURA 3.63 Menú Guardar de la pantalla **NOTAS**.

RESULTADOS

La funcionalidad que tiene la pantalla **RESULTADOS** es realizar una búsqueda de los informes que tienen un estado finalizado para poder obtener su informe de resultado. Esta pantalla también utiliza una tabla dinámica que cambiará de acuerdo al tipo de estudio que se seleccione para generar el informe en su formato correspondiente. Dicha pantalla se muestra en la figura 3.64.

FIGURA 3.64 Pantalla **RESULTADOS.**

El diagrama de flujo de datos que corresponde a la función **BUSCAR** se muestra en la figura 3.66.

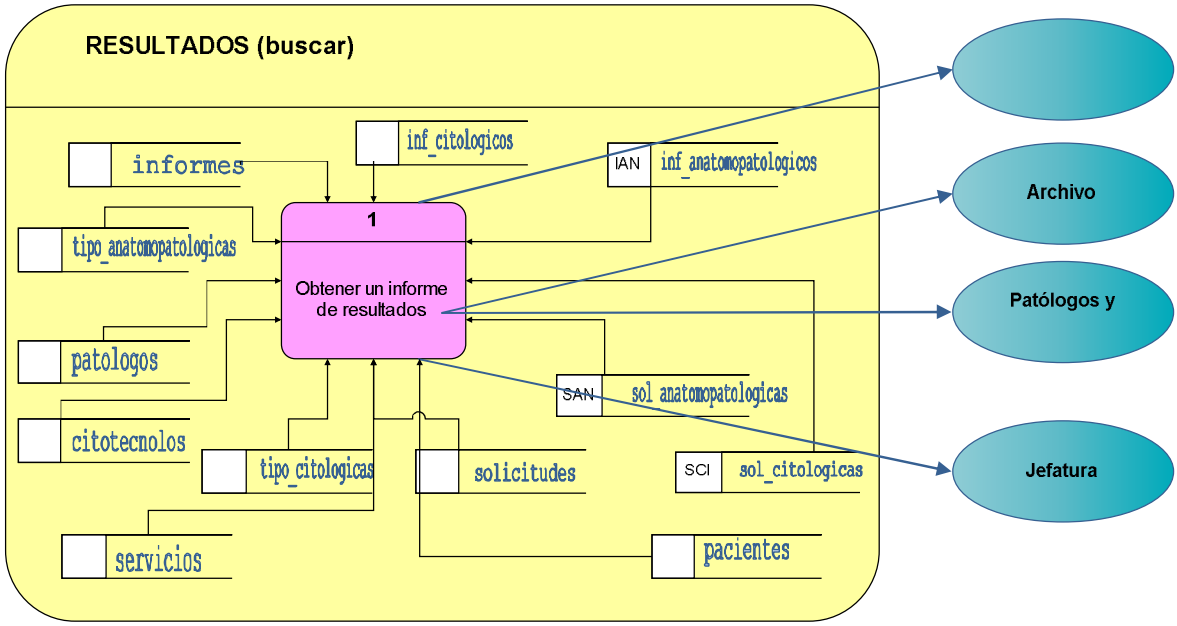
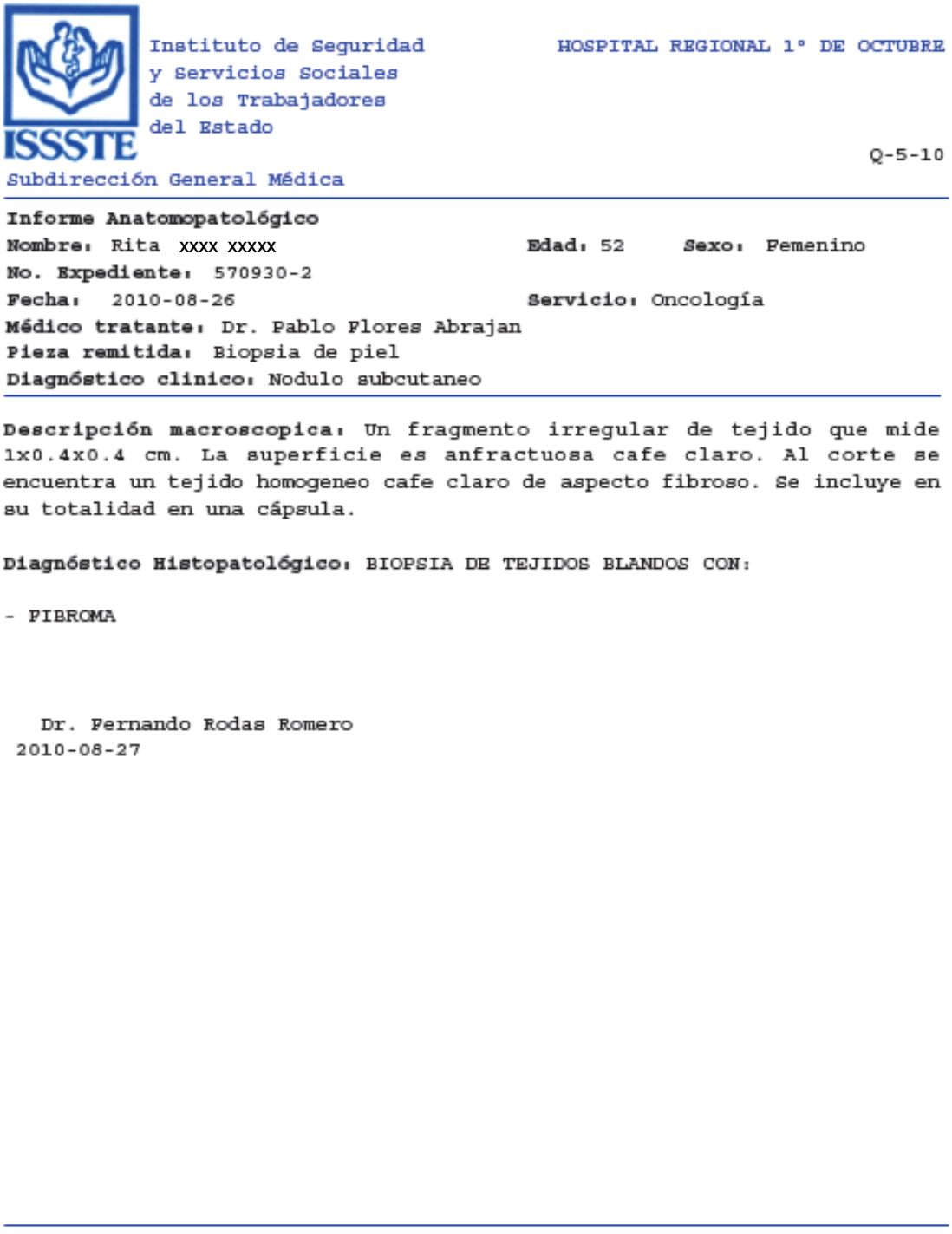


FIGURA 3.65 Diagrama de flujo de datos de la función **BUSCAR**.

Los formatos de los informes de resultados se muestran en las figuras 3.67, 3.68 y 3.69.

ANATOMOPATOLÓGICOS (GENERALES Y ONCOLÓGICOS)



Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado

ISSSTE

Subdirección General Médica

HOSPITAL REGIONAL 1° DE OCTUBRE

Q-5-10

Informe Anatomopatológico

Nombre: Rita xxxx xxxxx **Edad:** 52 **Sexo:** Pemenino

No. Expediente: 570930-2

Fecha: 2010-08-26 **Servicio:** Oncología

Médico tratante: Dr. Pablo Flores Abrajan

Pieza remitida: Biopsia de piel

Diagnóstico clinico: Nodulo subcutaneo

Descripción macroscopica: Un fragmento irregular de tejido que mide 1x0.4x0.4 cm. La superficie es anfractuosa cafe claro. Al corte se encuentra un tejido homogeneo cafe claro de aspecto fibroso. Se incluye en su totalidad en una cápsula.

Diagnóstico Histopatológico: BIOPSIA DE TEJIDOS BLANDOS CON:

- FIBROMA

Dr. Fernando Rodas Romero
2010-08-27

FIGURA 3.66 Ejemplo de un informe de resultado de tipo anatomopatológicos (generales y oncológicos).

CITOLOGICOS DE DIVERSOS

Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado

ISSSTE

Subdirección General Médica

HOSPITAL REGIONAL 1° DE OCTUBRE

C-9-10

Informe Citológico de Diversos

Nombre: Francisco XXXX XXXXX **Edad:** 56 **Sexo:** Masculino

No. Expediente: UVAS610401-1

Fecha: 2010-08-26 **Servicio:** Oncología

Médico tratante: Dr. Perez

Lugar: D.F **Unidad Médica:** 1° de octubre

Procedencia de muestras: Cavidad oral

Producto: Esputo

Método de obtención: Frotis

Muestra: Se reciben 4 laminillas de frotis de esputo procesadas con la tenica Papanicolaou

Descripción Microscopica: En los extendidos estudiados en un fondo inflamatorio a base de polimorfo nucleares se identifican estructuras alargadas septadas en moderada cantidad. Se observa abundante detritus celular.

Diagnóstico Citológico: Protis de cavidad oral con
- Candidiasis
- Inflamación aguda

Citotecnologo1 Dr. Fernando Rodas Romero

2010-08-26

FIGURA 3.67 Ejemplo de un informe de resultado de tipo citológico de diversos

CITOLOGICOS CERVICO VAGINAL

Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado

HOSPITAL REGIONAL 1° DE OCTUBRE

C-10-10

Subdirección General Médica

Informe Citológico Cervico Vaginal

Nombre: Josefa XXXX XXXXX **Edad:** 59

No. Expediente: MERJ510401-2

Fecha: 2010-08-26 **Unidad Medica:** 1° de octubre

Médico tratante: Dr. Flores

Detección de Cáncer Cervico Uterino: Primera vez

Antecedentes ginecobstétricos: Ninguno

Fecha de la última regla: 2010-08-11

Actualmente presenta: Post menopausia

A la exploración se observa: Cuello aparentemente sano

Utensilio con el que se tomó la muestra: Espátula de Ayre

Características de la muestra: Presencia de células endocervicales

Diagnóstico citológico: Neg. con proceso inflama

Hallazgos adicionales: bacterias

Repetir estudio: No

Motivo:

Nombre del citotecnólogo: Citotecnologo1

La muestra fue revisada por el patólogo: Si

Nombre del patólogo: Dr. Fernando Rodas Romero

Firma del patólogo

Observaciones: Dar tratamiento y repetir estudio en un año.

Citotecnologo1

2010-08-26

FIGURA 3.68 Ejemplo de un informe de resultado de tipo citológico cervico vaginal.

- **REPORTES**

Para verificar la productividad que hay en el Departamento de Servicio se contará con una pantalla **REPORTES** (véase la figura 3.69) para realizar reportes por paciente, por patólogos, por citotecnólogos, por fecha, etc. En cuando se oprima el botón **BUSCAR** en la pestaña de **Resultado** aparecerá en una tabla dinámica la información obtenida de la consulta y automáticamente se generará el archivo con los datos.

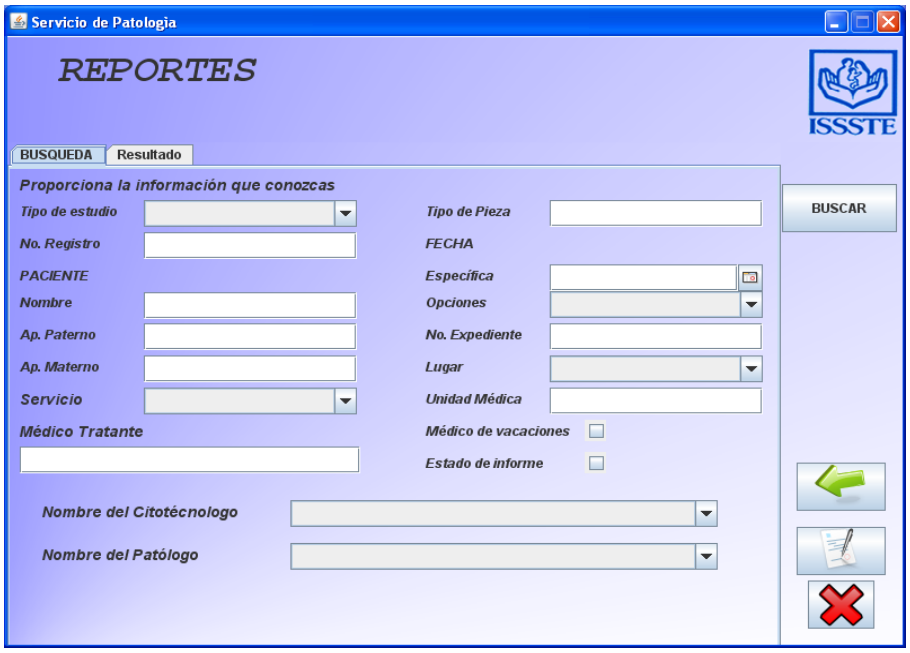


FIGURA 3.69 Pantalla **REPORTES**.

Para entender el flujo que datos que se efecturá al oprimir el botón **BUSCAR** se crea un diagrama que se muestra e la figura 3.70.

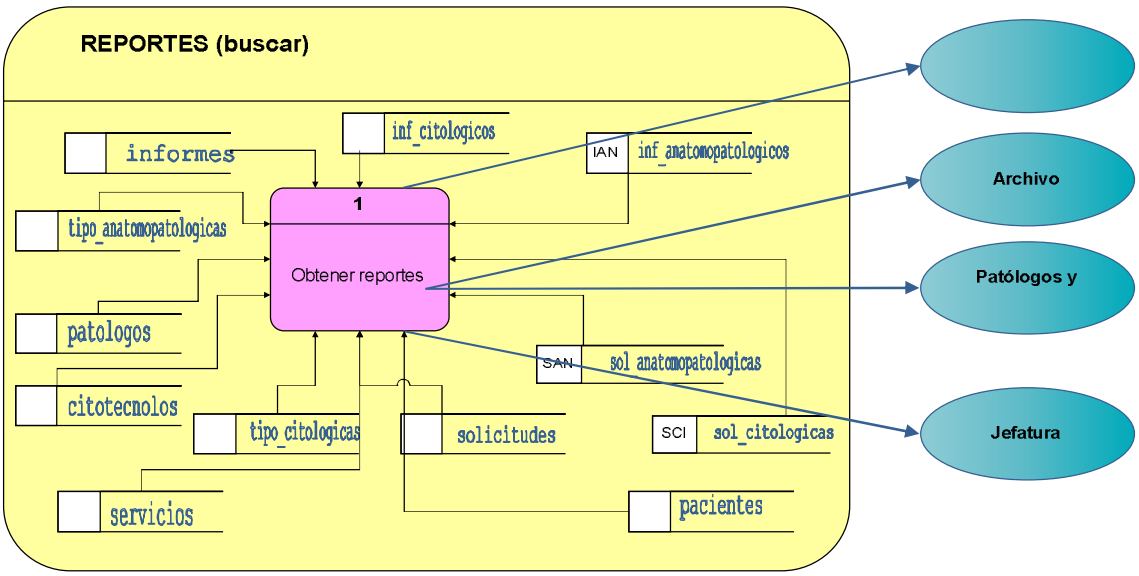
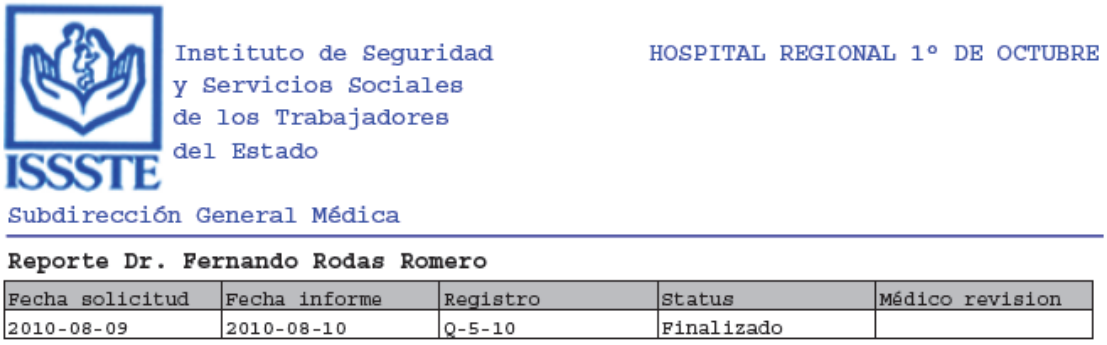


FIGURA 3.70 Diagrama de flujo de datos de la función **BUSCAR**.

Algunos ejemplos de los reportes que se obtienen se muestran en las figuras 3.71, 3.72 y 3.73.

REPORTE DE PATÓLOGO



Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado

HOSPITAL REGIONAL 1° DE OCTUBRE

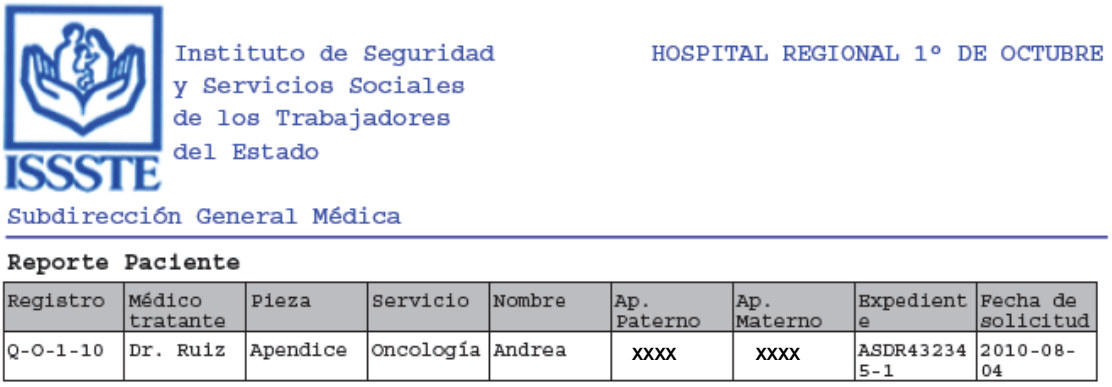
Subdirección General Médica

Reporte Dr. Fernando Rodas Romero

Fecha solicitud	Fecha informe	Registro	Status	Médico revision
2010-08-09	2010-08-10	Q-5-10	Finalizado	

FIGURA 3.71 Ejemplo de un reporte por patológo.

REPORTE DE PACIENTE



Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado

HOSPITAL REGIONAL 1° DE OCTUBRE

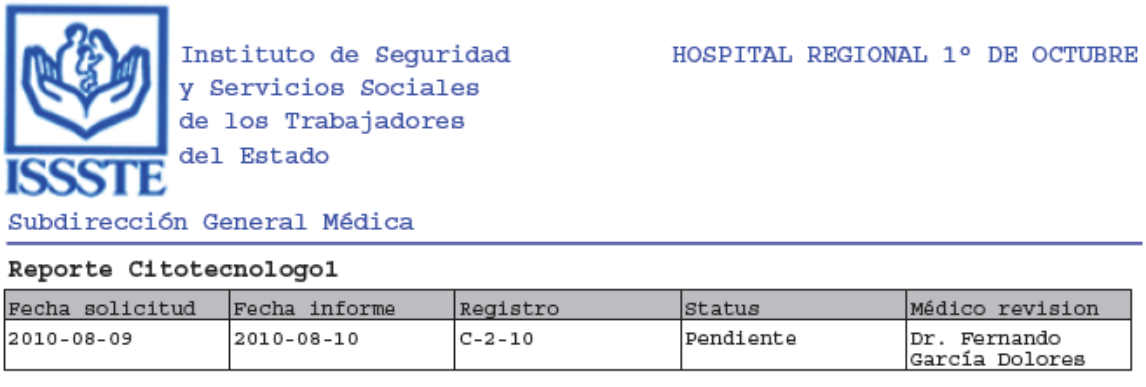
Subdirección General Médica

Reporte Paciente

Registro	Médico tratante	Pieza	Servicio	Nombre	Ap. Paterno	Ap. Materno	Expediente	Fecha de solicitud
Q-O-1-10	Dr. Ruiz	Apendice	Oncología	Andrea	xxxx	xxxx	ASDR432345-1	2010-08-04

FIGURA 3.72 Ejemplo de un reporte por paciente.

REPORTE DE CITOTECNÓLOGOS



Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado

HOSPITAL REGIONAL 1° DE OCTUBRE

Subdirección General Médica

Reporte Citotecnologo1

Fecha solicitud	Fecha informe	Registro	Status	Médico revision
2010-08-09	2010-08-10	C-2-10	Pendiente	Dr. Fernando García Dolores

FIGURA 3.73 Ejemplo de un reporte por citotecnólogo.

Esta pantalla también contempla que el administrador del sistema pueda cambiar el estado de los patólogos, es decir, indicar si están de vacaciones o activos. Por otra parte se puede modificar el estado del informe de finalizado a pendiente para poder realizar cambios. Se indica que el acceso a la pantalla **REPORTES** únicamente es para el administrador, en este caso es el Jefe del Departamento del Servicio.

Para realizar los cambios anteriormente mencionados se ejecutan algunos pasos mostrados en las figuras 3.74 y 3.75:

CAMBIAR ESTADO DE UN PATÓLOGO

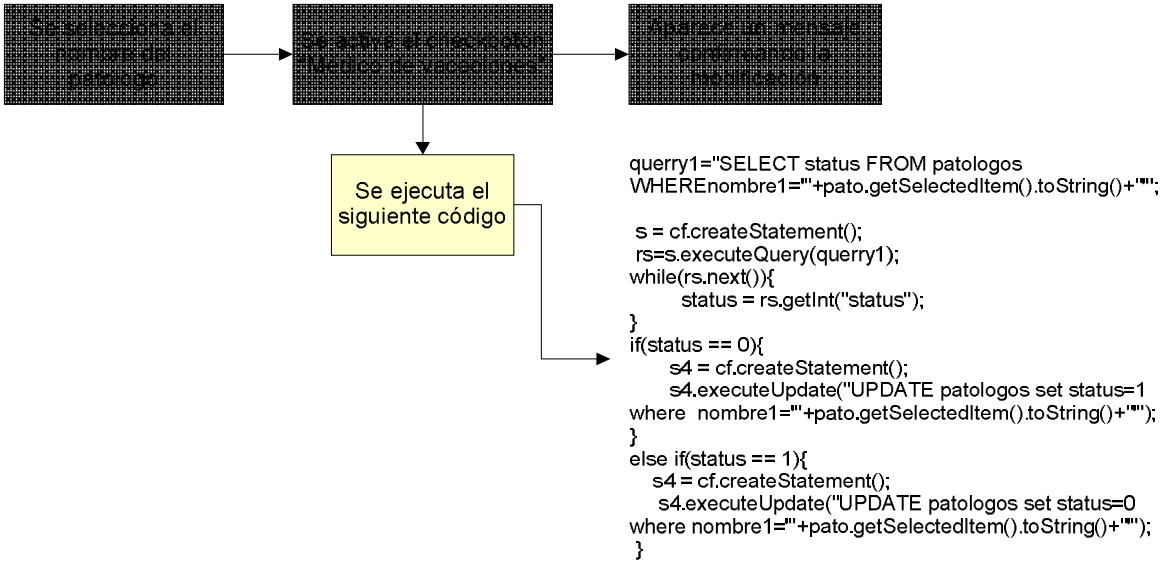


FIGURA 3.74 Procedimiento para cambiar el estado de un patólogo.

CAMBIAR ESTADO DEL INFORME

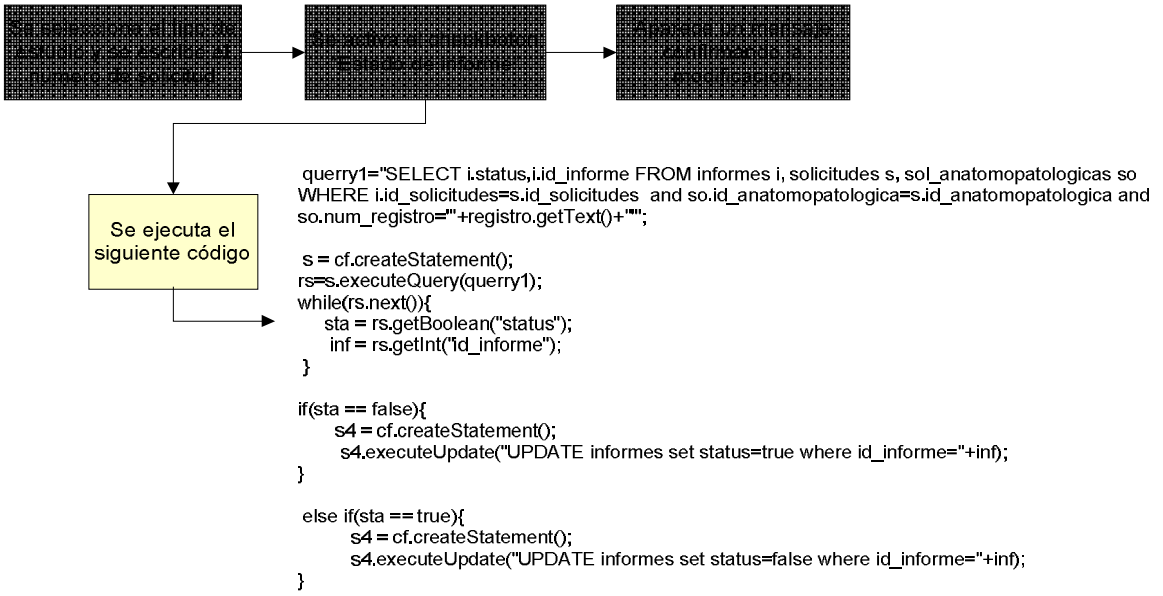


FIGURA 3.75 Procedimiento para cambiar el estado de un informe.

Después de describir las pantallas que formaran la aplicación, se mencionan ciertos aspectos extras que se utilizarán para poner en funcionamiento el sistema.

Validación de cadenas

Todas las pantallas cuentan con validaciones para los datos que serán ingresados en los campos, dichas validaciones se respaldan en expresiones regulares que se indican en la tabla 3.10.

Campo	Expresión regular
Números	[1-9][0-9]*
Letras	([A-ZÑ][a-zñ]*) | ([A-ZÑ][a-zñ]*\\s[A-ZÑ][a-zñ]*)|([A-ZÑ][a-zñ]*\\s[A-ZÑ][a-zñ]*\\.)|([A-ZÑ][a-zñ]*\\s[A-ZÑ][a-zñ]*\\.\\s[A-ZÑ][a-zñ]*)
Expedientes	([A-ZÑ0-9]+-[0-9])|([A-ZÑ]+\\s[0-9]+-[0-9])|([A-ZÑ0-9]+)|([A-ZÑ]+\\s[0-9]+)
Fecha	Valida que no sea mayor a la fecha actual
Doctor	(Dr\\.|Dra\\.)((\\s[A-ZÑ][a-zñ]+)|(\\s[A-ZÑ][a-zñ]*\\.)+)+)

TABLA 3.10 Expresiones regulares para la validación de los datos ingresados en los campos.

Librerías extras

- JCalendar: Es la que permite introducir calendarios dinámicos (véase la figura 3.76), facilita al usuario de no preocuparse en que orden debe escribir la fecha, únicamente lo que tiene que hacer es seleccionar el día, mes y año, automáticamente se pondrá la fecha en el orden correcto.

FIGURA 3.76 Calendario dinámico de la librería JCalendar.

- iText: Esta librería permite generar archivos PDF con el formato que se desee. Se puede poner un encabezado y un pie de página, definirse formatos de letras, crear tablas e insertar imágenes.

Para crear un documento pdf se siguen los siguientes pasos:

1. Se indica el nombre y su ubicación del documento.

```
PdfWriter writ = PdfWriter.getInstance(doc, new FileOutputStream("C: \\reportes\\"+"Rep"+dats+".pdf"));
writ.setPageEvent(new GenInforme1());
```

2. Se abre el documento para realizar la escritura, se agregan los componentes diseñados y por último se cierra el documento.

```
doc.open();
doc.add(t);
doc.close();
```

Para realizar el diseño se utilizan tablas, se indica cuantas columnas tendrá al igual que el tamaño de cada una de ellas.

```
PdfPTable table = new PdfPTable(6);
table.setWidthPercentage(100);
 int i[] = {10,40,7,8,8,17};
table.setWidths(i);
```

Para abrir automáticamente el archivo generado se ejecuta la siguiente instrucción:

```
Runtime.getRuntime().exec("cmd.exe /c start "+path+datos[0]+".pdf");
```

Consultas y transacciones distribuidas

Para realizar consultas y transacciones distribuidas dentro de la aplicación en el código se agregan las consultas de la siguiente manera:

```
dblink('dbname=”nombre de la bd” port=”puerto”
        host=”nombre del host”
        user=”nombre del usuario”
        password=”password”',
        querry (select,insert,update ó delete))
```

Ejemplos:

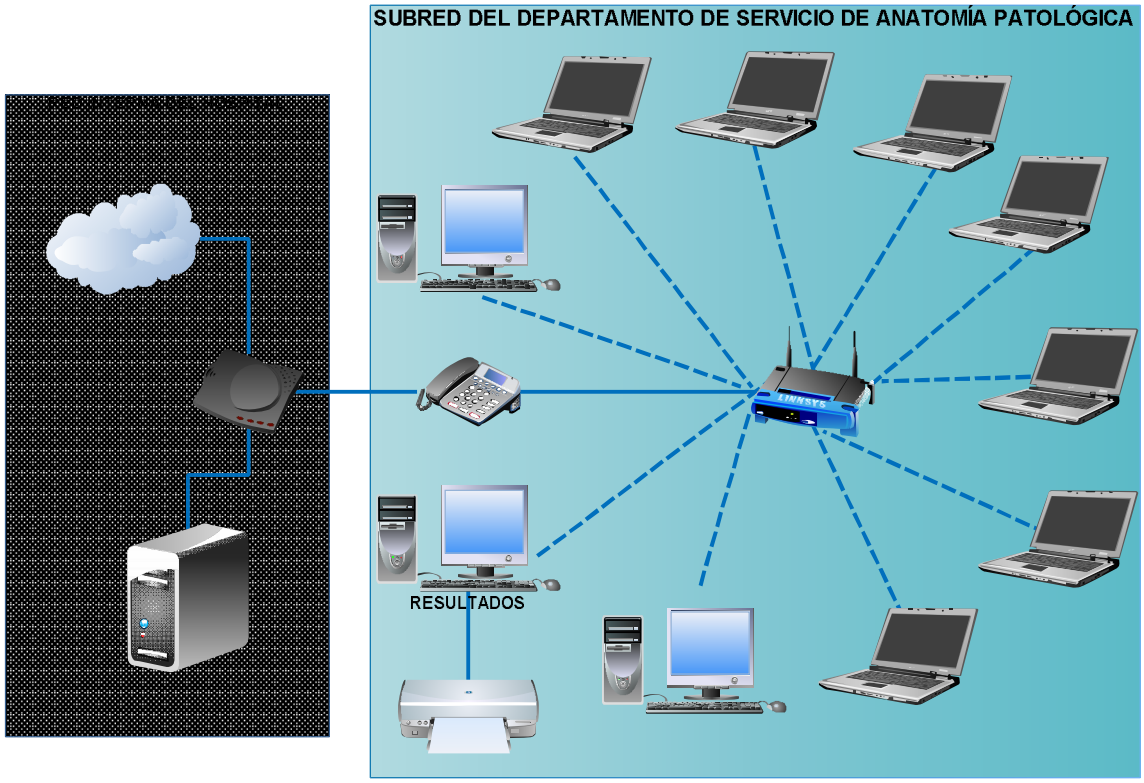
A)
```
dblink('dbname=Patologia port=5432
  host=192.162.130.150
  user=postgres
  password=********,
  SELECT medico_responsable FROM solicitudes WHERE id_solicitudes="+id_soli)
```

B)
```
dblink('dbname=Patologia port=5432
 host=192.162.130.150
 user=postgres
 password=********,
   INSERT into inf_anatomopatologicos (id_patologo,descripcion_macroscopica,diagnostico_histopatologico)
 VALUES (?,?,?))
```

C)
```
dblink('dbname=Patologia port=5432
  host=192.162.130.150
  user=postgres
  password=********,
  UPDATE informes set fecha_informe=?,dia1=? WHERE id_inf_anato="+id_info_anato)
```

3.3 DISEÑO DE LA RED DE DATOS INALÁMBRICA

Para el diseño de la red de datos inalámbrica se construye el diagrama de conexión basado en una topología de modo infraestructura que se muestra en la figura 3.77.



FIGRUA 3.77 Diagrama de conexión de la red de datos inalámbrica.

Por razones de seguridad las direcciones IP del servidor y de los nodos que conforman la subred del Departamento de Servicio de Anatomía Patológica no aparecerán en la documentación, esto se debe a que son direcciones IP públicas, las cuales son visibles para cualquier ordenador conectado a Internet.

CONFIGURACIÓN DEL ACCESS POINT (AP)

La señal de internet es suministrada al Departamento de Servicio de Anatomía Patológica a través de cableado. Por otro lado los nodos utilizados en la subred se comunicarán por medio inalámbrico, por esta razón se requiere de un Access point (AP) que convierta la señal alambrica a inalámbrica.

Para ingresar a la página de las configuraciones en el Explorador de Internet, se coloca la dirección 192.168.1.1 (véase en la figura 3.78).

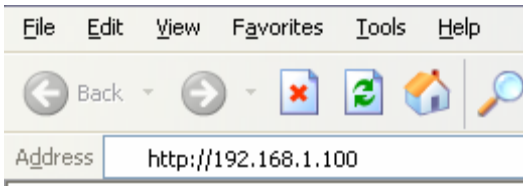


FIGURA 3.78 Página del Explorador de Internet con la dirección del AP.

Se ingresa el usuario y el password para realizar los cambios al Access point como se observa en la figura 3.79.

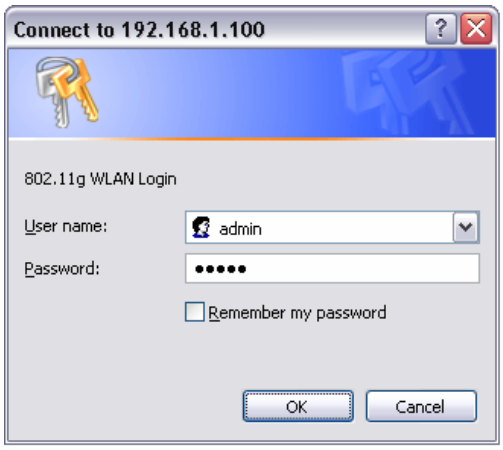


FIGURA 3.79 Pantalla del AP para ingresar el usuario y el password.

Ahora se modifican las características del Access point siguientes los pasos que a continuación se indican:

1. El access point que se utiliza contempla diferentes modos de operación por lo cual se elije en modo access point. (véase la figura 3.80).

FIGURA 3.80 Elección del modo Access point.

2. Se elije el canal en el que trabajará el AP, soporta los siguientes rangos:
   - Norteamérico (FCC) 1 al 11
   - Europeo (ETSI) 1 al 13
   - Japonés 1 al 14

   Se recomienda utilizar el canal 6 como se muestra en la figura 3.81 si la red está instalada en un solo piso.

FIGURA 3.81 Elección del canal 6.

3. Se coloca el identificar de servicio, nombre único para compartir entre los nodos de la red inalámbrica. En la figura 3.82 se muestra un ejemplo de un identificador.

FIGURA 3.82 Ejemplo de un identificador de servicio.

4. El tipo de autenticación por defecto está configurado como sistema abierto, el Access point contempla seis opciones: Sistema abierto; Shared Key, WPA-PSK, WPA2-PSK, WPA y WPA2 (véase la figura 3.83). Todos los nodos de la red deben utilizar el mismo tipo de autenticación.

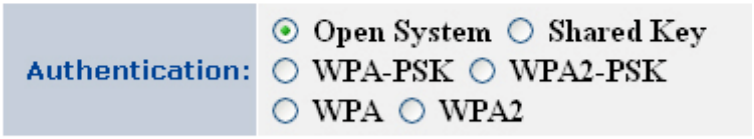


FIGURA 3.83 Tipos de autenticación.

5. Para activar la seguridad WEP, se debe dar clic en la opción "Activar". La seguridad WEP, contempla 2 tipos para seleccionar – 64 bits y 128 bits. Cuando se seleccionado el valor de la clave se debe introducir en formato ASCII o HEX. Todos los nodos deben tener habilitado el protocolo WEP con el valor de la clave que anteriormente se introdujo. Lo anterior se visualiza en la figura 3.84.

FIGURA 3.84 Configuración de la seguridad WEP en el AP.

CONFIGURACIÓN DE CADA NODO

1. Ir a panel de control y seleccionar la opción de Conexiones de red.

2. Dar clic derecho en el icono de red inalámbrica como se indica en la figura 3.85.

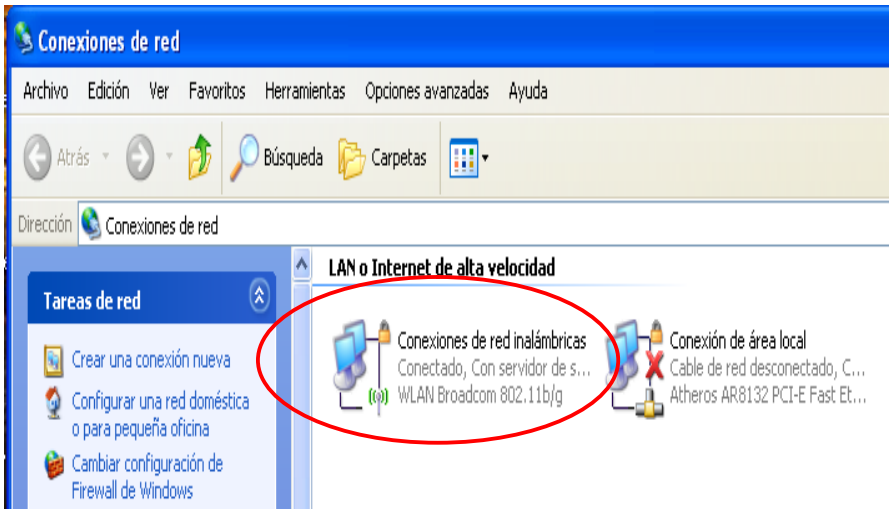


FIGURA 3.85 Ejemplo del icono de la red inalambrica.

3. En el submenú que aparece seleccionar propiedades (véase la figura 3.86).

FIGURA 3.86 Submenú del icono de red inalámbrica.

4. En el diagrama de la figura 3.87 se muestra la configuración que se tiene que realizar para activar la seguridad WEP en cada nodo.

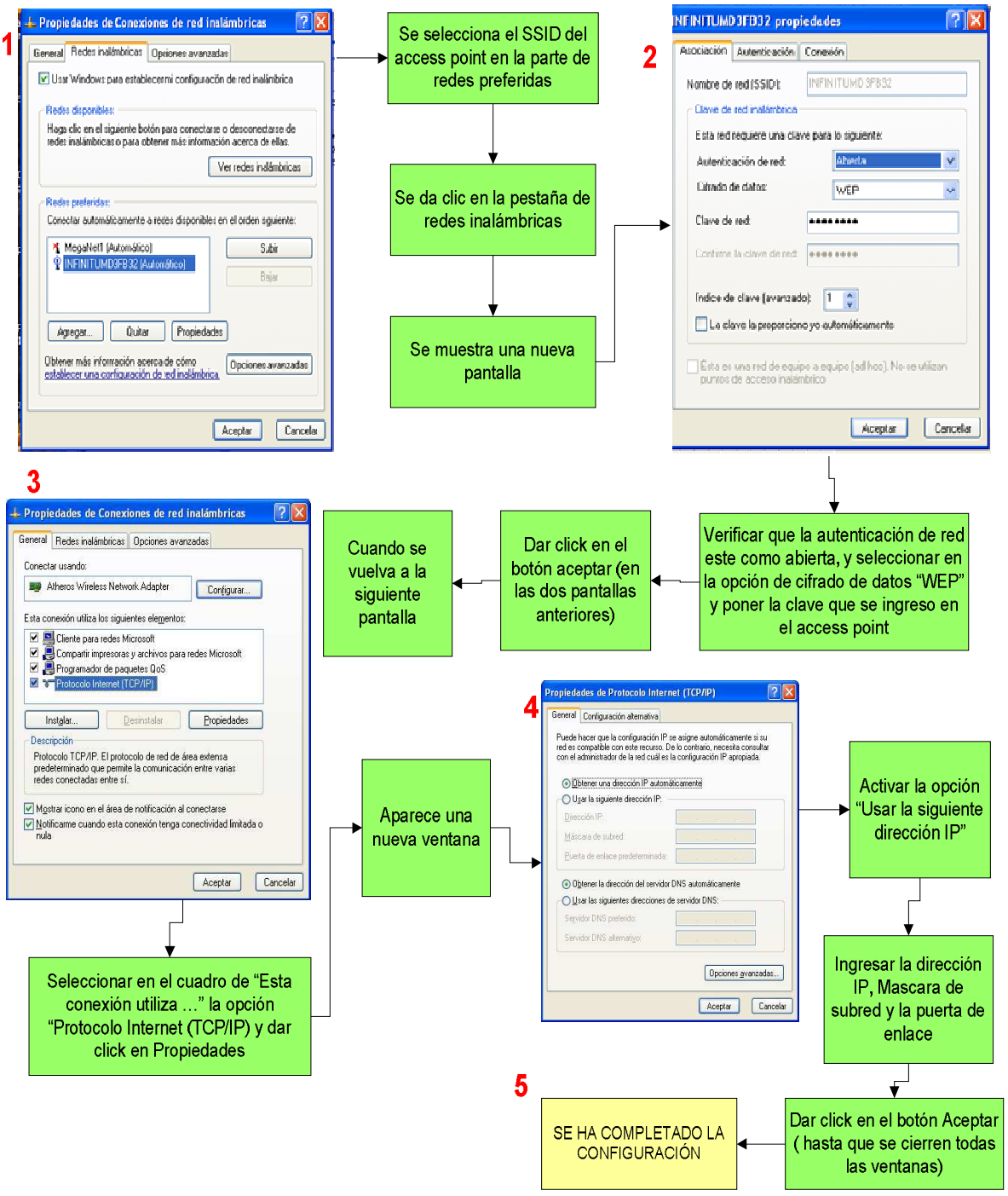


FIGURA 3.87 Configuración de la seguridad WEP en cada nodo de la red inalámbrica.

Para validar la configuración realizada se ejecuta un ping para comprobar la comunicación entre los distintos nodos de la red como se muestra en la figura 3.88.

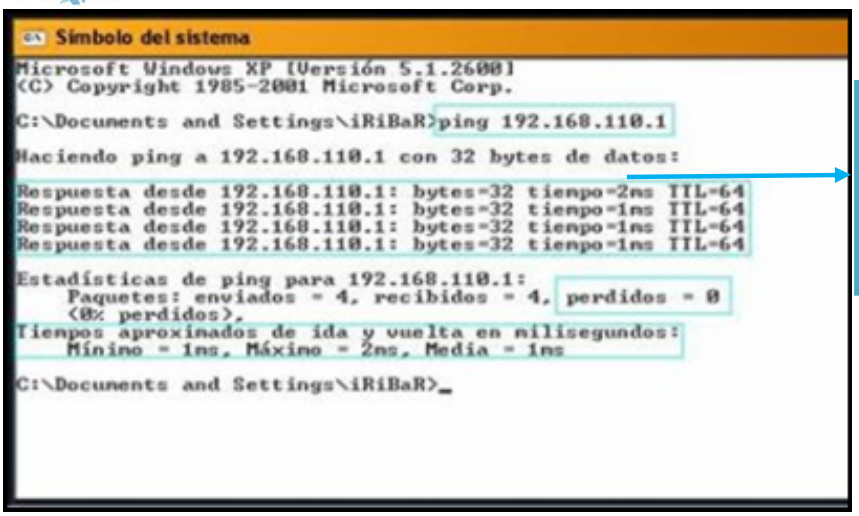


FIGURA 3.88 Ejecución en consolo del comando ping.

Con lo anterior se confirma que la comunicación entre los nodos está establecida y soporta el envió y recepción de información, lo cual es muy importante para el buen funcionamiento del Sistema.

- **CREACIÓN DE UN GRUPO DE TRABAJO**

1. Abrir el **Panel de control**, en **Sistema y mantenimiento** y, a continuación, en **Sistema**.
2. En **Configuración de nombre, dominio y grupo de trabajo del equipo**, dar clic en **Cambiar la configuración.**
3. Dar clic en la ficha **Nombre del equipo** y, a continuación, en **Cambiar**.
4. En **Miembro de**, dar clic en **Grupo de trabajo** y realice una de las acciones siguientes:

- Para unirse a un grupo de trabajo existente, escriba el nombre del grupo de trabajo al que se quiere unir y, a continuación, haga clic en **Aceptar**. (véase la figura 3.89)
- Para crear un nuevo grupo de trabajo, escriba el nombre del grupo de trabajo que quiere crear y, a continuación, haga clic en **Aceptar.** (véase la figura 3.89)

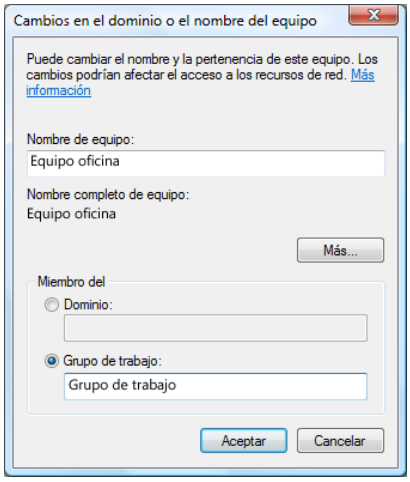


FIGURA 3.89 Pantalla para cambiar o unirse a un grupo.

CAPÍTULO 4

IMPLEMENTACIÓN

La implementación es un lenguaje
y lo principal es cómo usas ese lenguaje.
Tibor Kalman

4.1 IMPLEMENTACIÓN DEL SISTEMA SIDIPAT

Posteriormente al diseño y construcción de la base de datos, interface de usuario y la red inalámbrica se procede a la implementación en las instalaciones del Departamento de Servicio de Anatomía Patológica. El proceso que se sigue para la implementación se describe a continuación:

- **Para todos los nodos**

1. Instalar el jre-6u21-windows-i586 de JAVA (con la finalidad de poder ejecutar aplicaciones JAVA). En la figura 4.1 se muestra el icono de la versión de JAVA requerida.

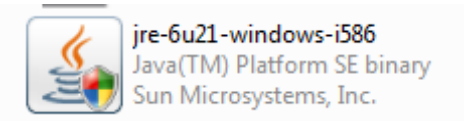


FIGURA 4.1 Icono de la versión de JAVA.

2. Adicionar las librerías jText y jCalendar, al igual que el driver de postgreSQL para java en la carpeta donde se instalo el JRE de JAVA (C:\Archivos de programa\Java\jre6\lib\ext). Lo anterior se puede observar en la figura 4.2.

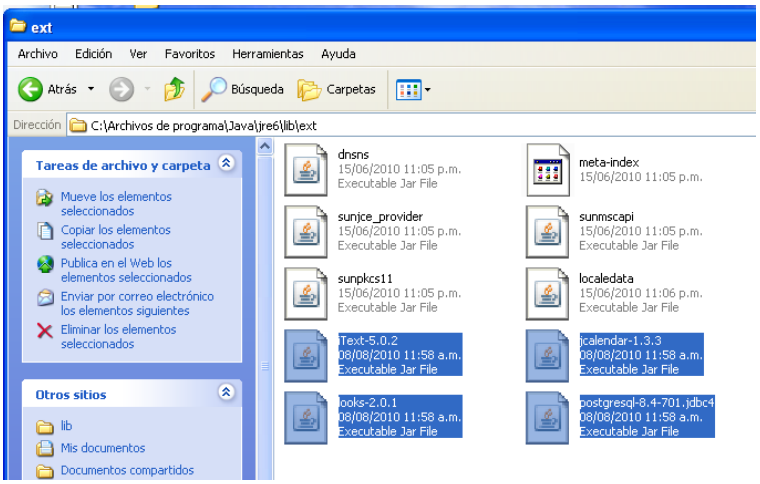


FIGURA 4.2 Imagen del directorio que contiene las librerías.

3. Crear una carpeta llamada “reportes” en la siguiente ubicación “C:\Documents and Settings\All Users\Escritorio”. En esta carpeta se guardaran todos los archivos PDF que creara la aplicación.

4. Se instala el ejecutable de la aplicación en la unidad C:\ y se crea un enlace directo en el Escritorio.

- **Para los nodos que contienen la base de datos**

1. Crear una carpeta llamada “Proyecto” en la siguiente ubicación “C:\”. Esta carpeta será útil para crear el tablespace de la base de datos.

2. Dar permisos a la carpeta de Proyecto (verificar que en opciones de carpeta este deshabilitada la opción “Utilizar uso compartido simple de archivos”).

3. Instalación del manejador de base de datos PostgreSQL (Apéndice B se encuentran los pasos para realizar la instalación).

4. Creación de los usuarios en el manejador de bases de datos con ayuda de pgAdmin. El procedimiento se muestra en la figura 4.3.

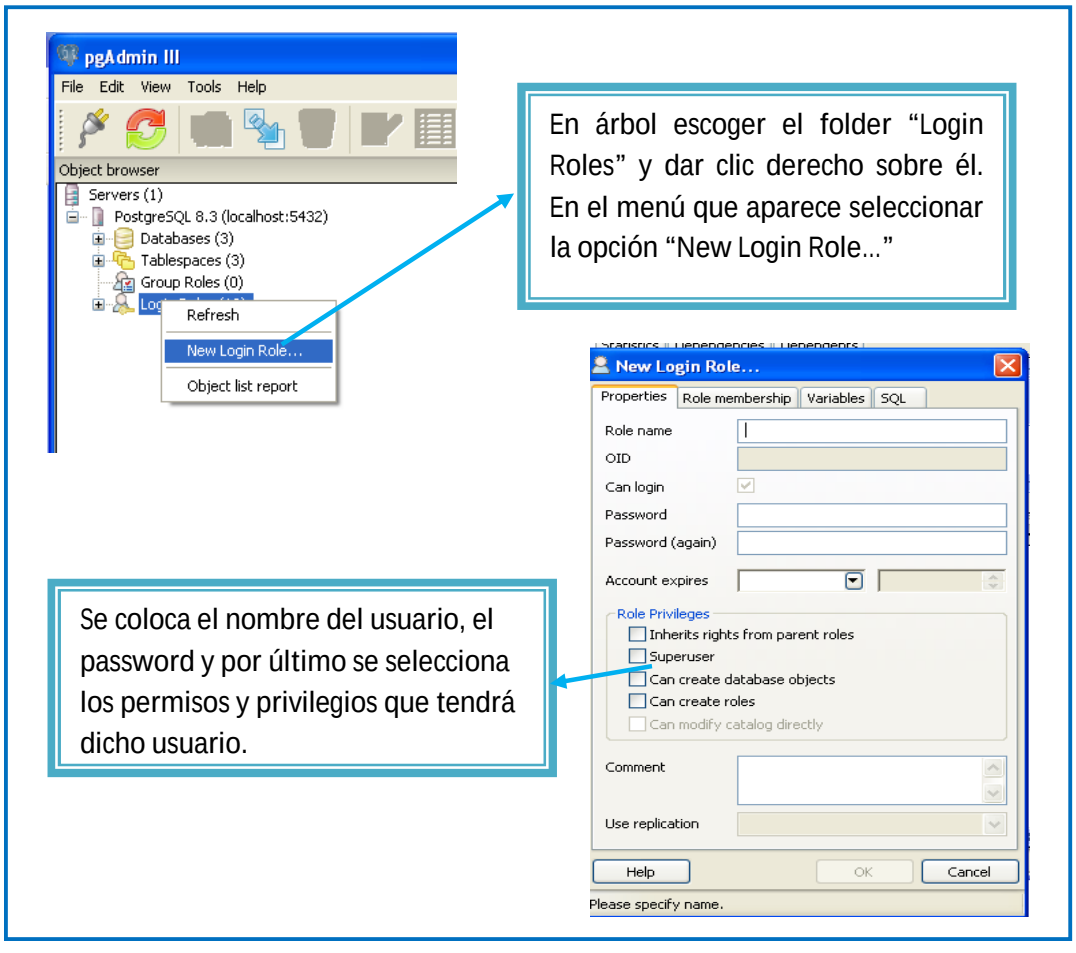


FIGURA 4.3 Procedimiento para la creación de usuarios.

La lista de usuarios se muestra en la tabla 4.1.

USUARIO
DrVazquez
DraFranco
DrRodas
DrHernandez
DrMegia
DrPopoca
DrDolores
Secretaria
Citotecnologo

TABLA 4.1 Lista de usuario para el sistema ***SIDIPAT.***

5. Creación de tablespace patología.

6. Creación de la base de datos “Patologia”.

7. Carga del respaldo que se obtuvo en la etapa de diseño y construcción de la base de datos. (De acuerdo a las características de fragmentación y replicación de cada nodo).

4.1.1 CONFIGURAR LA REPLICACIÓN

PostgreSQL cuenta con varias aplicaciones para replicar, el único inconveniente que presentan es que la mayoría únicamente soportan plataformas Linux. La aplicación que está comprobada parra llevar a cabo la función de replicación en plataformas Windows es Slony - I.

1. Instalar la aplicación Slony-I (Apéndice B se muestra el proceso de instalación).

2. En una consola de MS-DOS, ubicar el siguiente directorio "C:\Program Files\PostgreSQL\8.2\bin" y ejecutar el comando: ***Slon*** en la figura 4.4 se observa el resultado de dicho comando.

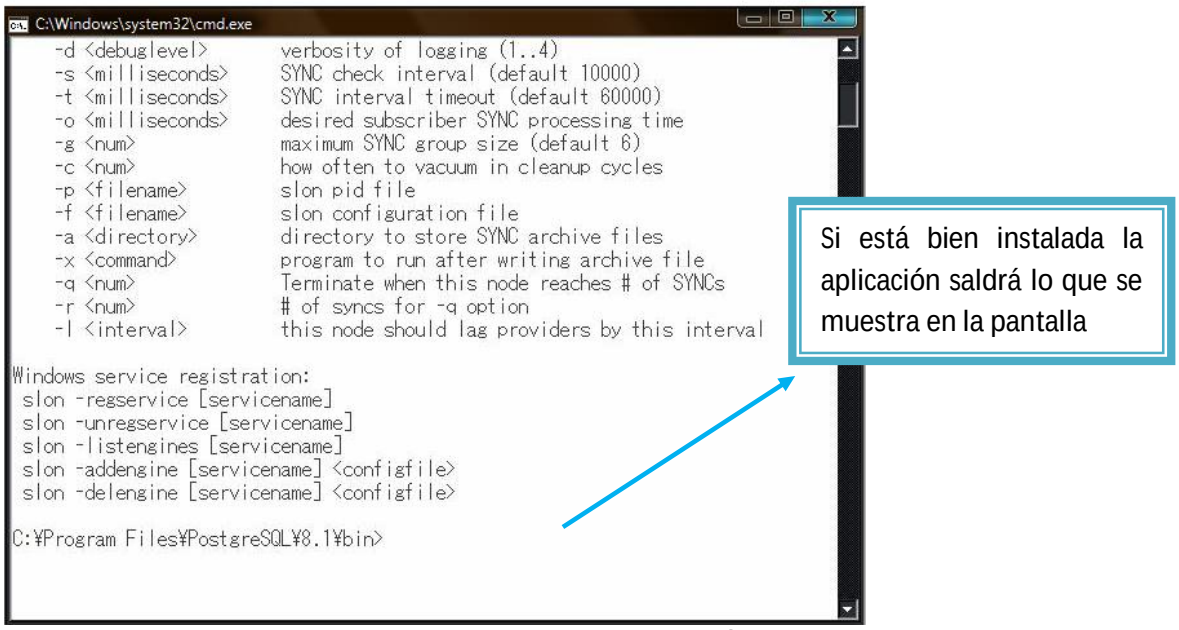


FIGURA 4.4 Pantalla que muestra la ejecución del comando Slon.

3. Con la ayuda de pgAdmin se da clic en el folder de "Database", seleccionar la base de datos que se desea replicar. Aparece un folder "Replication", dar clic derecho y seleccionar la opción "New Slony-I cluster". En la parte inferior de la nueva ventana se observa el siguiente mensaje "No hay comandos de creación de Slony ". Lo anterior se muestra en la figura 4.5.

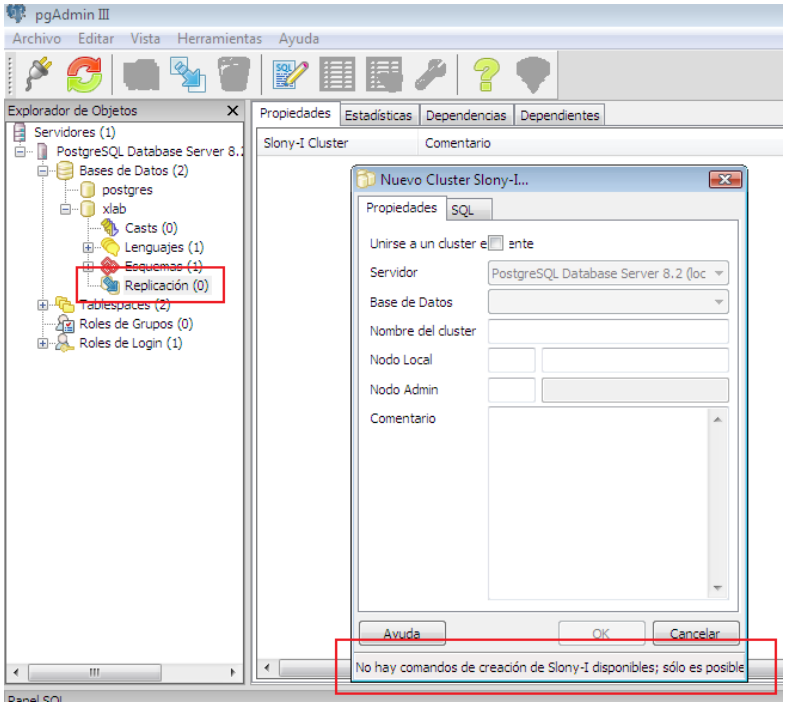


FIGURA 4.5 Pantalla que muestra la elección de la base de datos a replicar.

Se indica la ubicación donde se instalaron los archivos de Slony-I, para ello se da clic en Archivo posteriormente en Opciones y se introduce la ruta "C:\Program Files\PostgreSQL\8.2\share". En la figura 4.6 se observa el mensaje que aparece "Por favor especifique nombre".

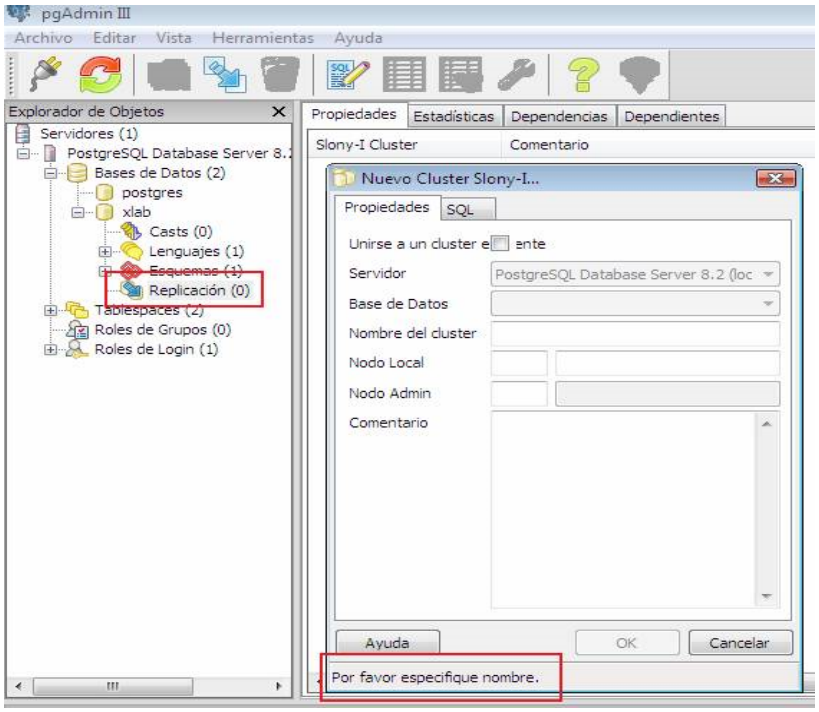


FIGURA 4.6 Pantalla que muestra el mensaje que aparece cuando se especifica la ruta de los archivos de Slony –I.

4. Se crea un script para especificar las tablas que se replicaran, los nodos que se comunicarán en la replicación.

```
############################################################################################
#--
# define the namespace the replication system uses in our example it is
# slony_example
#--

cluster name = patología_replica;

#--
# admin conninfo's are used by slonik to connect to the nodes one for each
# node on each side of the cluster, the syntax is that of PQconnectdb in
# the C-API - replace the parameters which are appropriate for your setup
#
# this section tells slonik how to connect to the various nodes when running
# this script.
# --

node 1 admin conninfo = 'dbname=Patologia host=***.***.***.*** user=postgres
password= ********';
node 2 admin conninfo = 'dbname=Patologia host=***.***.***.*** user=postgres
password= ********';

#--
# init the first node. Its id MUST be 1. This creates the schema
# _$CLUSTERNAME containing all replication system specific database
# objects.

#--

init cluster (id=1, comment = 'Nodo Maestro');

#--
# this is an example of how to work around a table that has no primary key
# and is based on an example by Robert Treat. The history table referred to
# is one of the tables being replicated.
#
# Because the history table does not have a primary key or other unique
# constraint that could be used to identify a row, we need to add one.
# The following command adds a bigint column named
# _Slony-I_$CLUSTERNAME_rowID to the table. It will have a default value
# of nextval('_$CLUSTERNAME.s1_rowid_seq'), and have UNIQUE and NOT NULL
# constraints applied. All existing rows will be initialized with a
# number
#--
# Traducción: Esto solo es necesario usarlo si tienes una tabla que no tenga una
llave primaria, procura hacer un buen diseño en tu base de datos.

#table add key (node id = 1, fully qualified name = 'public.nombre de la tabla
que no tiene llave primaria);

#--
# Slony-I organizes tables into sets. The smallest unit a node can
# subscribe is a set. The following commands create one set containing
# all tables to be replicated. Tables not named here will not be replicated.
# The master or origin of the set is node 1. Note that the second id must be
# incremented for each table added.
#--

create set (id=1, origin=1, comment='aqui van todas mis tablas');
```

```
set add table (set id=1, origin=1, id=1, fully qualified name =
'public.solicitudes',comment='solicitudes');
set add table (set id=1, origin=1, id=2, fully qualified name =
'public.informes',comment='informes');
#set add table (set id=1, origin=1, id=3, fully qualified name = 'public.nombre
de la tabla que no tiene llave primaria', key = serial, comment='nombre de la
tabla que no tiene llave primaria');

#--
# Create the second node (the slave) tell the 2 nodes how to connect to
# each other and how they should listen for events.
# we have to repeat the conninfo detils here, because these details are written
into
# the slony database.
#--

store node (id=2, comment = 'Nodo Esclavo');
store path (server = 1, client = 2, conninfo='dbname=Patologia
host=***.***.***.*** user=postgres password=********');
store path (server = 2, client = 1, conninfo='dbname=Patologia
host=***.***.***.*** user=postgres password=********');
store listen (origin=1, provider = 1, receiver =2);
store listen (origin=2, provider = 2, receiver =1);

##############################################################################################
```

5. Cuando se termina de crear el script se guarda en el directorio "C:\Program Files\PostgreSQL\8.2\bin" , ubicándose en dicho directorio en la consola de MS-DOS se ejecuta el siguiente comando: ***slonik nombre_script*** observando en la figura 4.7 el resultado de su ejecución.

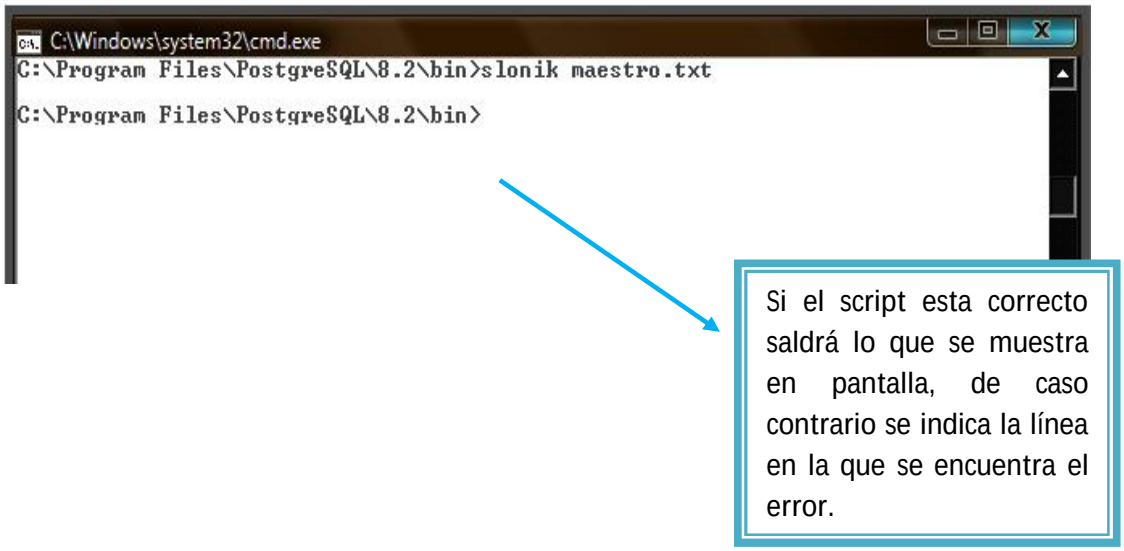


FIGURA 4.7 Pantalla que muestra la ejecución del comando slonik.

6. En el archivo pg_hba.conf que se ubica en "C:\Archivos de programa\PostgreSQL\8.3\data" (véase en la figura 4.8) colocar las direcciones IP de los nodos que se comunicaran como se muestra en la tabla 4.2, para que el manejador pueda realizar conexiones remotas a través de TCP/IP.

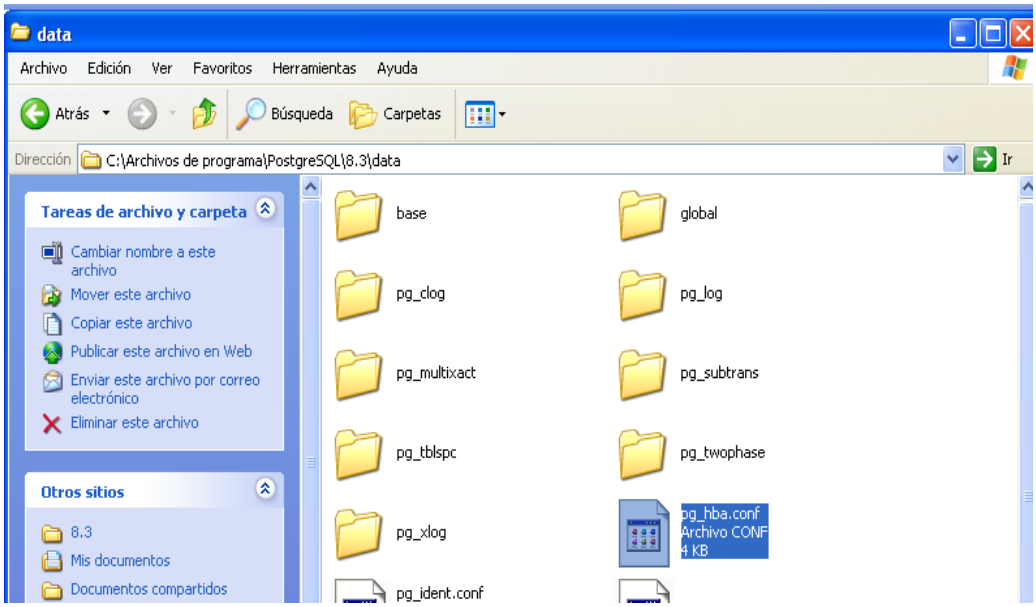


FIGURA 4.8 Ubicación del archivo pg_hba.conf.

```
# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD

# IPv4 local connections:
host    all         all         127.0.0.1/32            md5
host    all         all         ***.***.***.***/32      md5
host    all         all         ***.***.***.***/32      md5
host    all         all         0.0.0.0 0.0.0.0         md5

# IPv6 local connections:

#host    all         all          ::1/128                md5
```

TABLA 4.2 Ejemplo de direcciones IP que están contenidas en el archivo pg_hba.conf.

7. Para verificar que se ha configurado adecuadamente la replicación, se abre pgAdmin , en el folder de "Replication" de la base de datos se mostrará la configuración de la replicación. En la figura 4.9 se muestra dicha configuración.

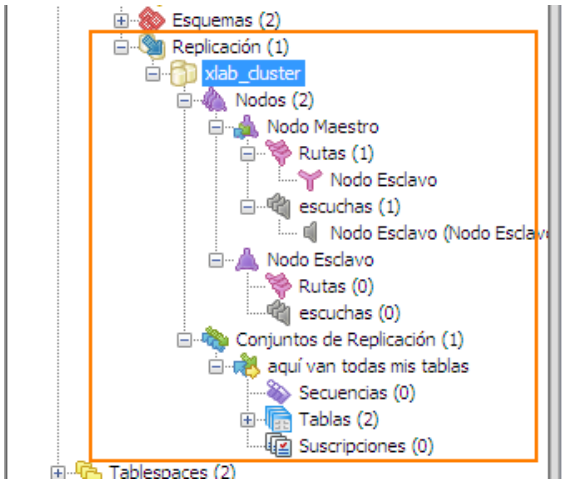


FIGURA 4.9 Configuración de la replicación.

8. Repetir el paso 7 en la computadora que actuará como esclavo para verificar la replicación.

9. En el nodo esclavo se realiza un script para configurar las suscripciones

```
##############################################################################################
CLUSTER NAME = patología_replica;

node 1 admin conninfo = 'dbname=Patologia host=***.***.***.*** user= postgres
password= ********';

node 2 admin conninfo = 'dbname=Patologia host=***.***.***.*** user= postgres
password= ********';

SUBSCRIBE SET (ID = 1, PROVIDER = 1, RECEIVER = 2, FORWARD = YES);

##############################################################################################
```

10. Finalizado el script se guarda en el directorio "C:\Program Files\PostgreSQL\8.2\bin". Estando en dicho directorio en la consola de MS-DOS se ejecuta el siguiente comando: ***slonik nombre_script***, en la figura 4.10 se observa el resultado de la ejecución.

FIGURA 4.10 Pantalla de la ejecución del comando slonik.

11. En el nodo maestro se ejecuta el siguiente comando:

    slon patologia_replica "dbname=**Patologia** user=**postgres password=********"

    Posteriormente se ejecuta en el nodo esclavo. Al realizar la ejecución de dicho comando se inicia el proceso de replicación, esto se muestra en la figura 4.11.

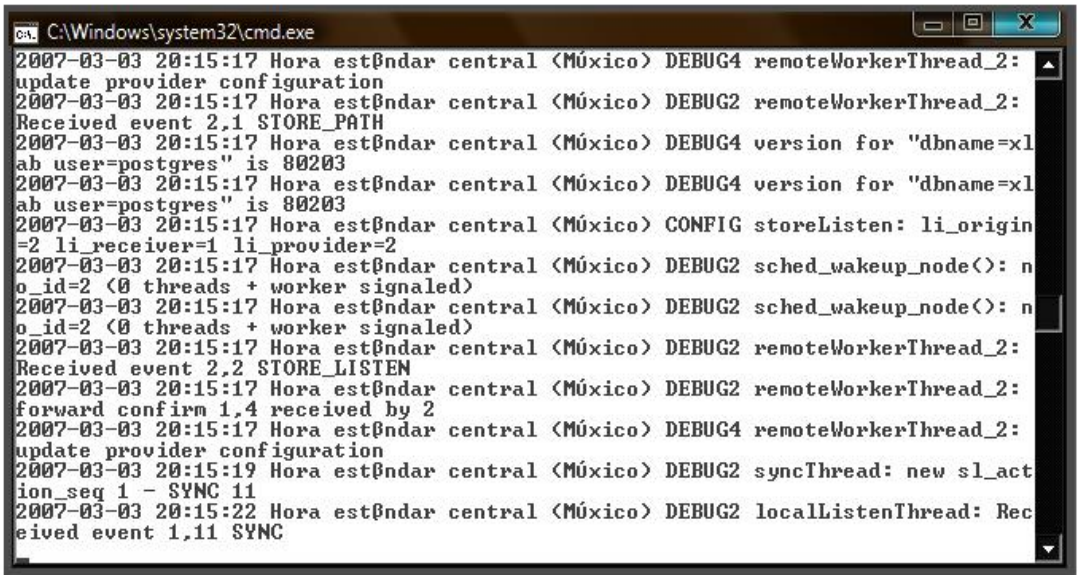


FIGURA 4.11 Pantalla de la ejecución del comando slon, inicio de la replicación.

NOTA: La consola donde se ejecute dicho comando en el nodo maestro y esclavo no debe ser cerrada, de caso contrario se interrumpe la replicación.

12. Se crea un servicio en Windows para que se inicie automáticamente la replicación. En línea de comandos en C:\Program Files\PostgreSQL\8.2\bin se ejecuta ***slon –regservice.***

13. Se crea un archivo para el servicio en Windows llamado slon.conf en C:\Program Files\PostgreSQL\8.2\bin

```
##################################################################################

cluster_name = 'patologia_replica' conn_info = 'dbname=Patologia
host=***.***.***.*** user=postgres password=********'

##################################################################################
```

14. Se cierra la consola del nodo maestro y esclavo donde se activa la replicación (consolas del paso 11). En línea de comandos ubicarse en C:\Program Files\PostgreSQL\8.2\bin y ejecutar el siguiente comando: ***slon -addengine slon.conf.***

15. Abrir los servicios de Windows, buscar el que se llama “Slony-I” (véase en la figura 4.12) y dar clic en “Iniciar”, automáticamente la base de dato se estará replicando.

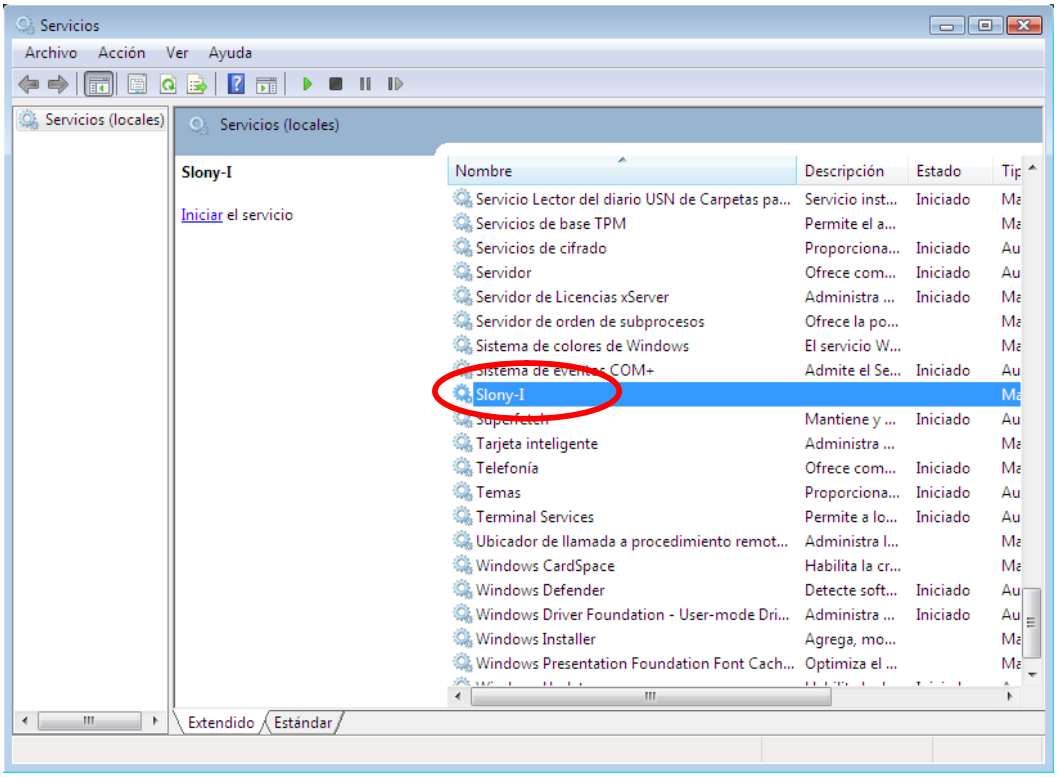


FIGURA 4.12 Servicio de Windows Slony –I.

CAPÍTULO 5

PRUEBAS Y SEGURIDAD

Los grandes trabajos no son hechos
por la fuerza, sino por la perseverancia.
Ben Jonson

En este capítulo se muestran algunas pruebas realizadas al sistema ***SIDIPAT*** así como las configuraciones de seguridad consideradas en dicho sistema.

5.1. PRUEBAS

Las pruebas son un elemento crítico para la garantía de la calidad de la aplicación. Además implica los siguientes pasos:

- Verificar la interacción de componentes.
- Verificar la integración adecuada de los componentes.
- Verificar que todos los requisitos se han implementado correctamente.

Las pruebas son un proceso de ejecución de una aplicación con la intención de descubrir un error. Una prueba tiene éxito si descubre un error no detectado hasta entonces.

- **PRUEBAS DE VALIDACIÓN**

La primera prueba que se ejecuta en el sistema es la validación del password y del usuario. Se provoca que el password tenga una longitud mayor a ocho. Si se está validando adecuadamente aparecerá un mensaje indicando el error como se muestra en la figura 5.1.

FIGURA 5.1 Pantalla de error de validación de password.

También se ocasiona que no sea posible la conexión con la base de datos, al ingresar un usuario que no está dado de alta en el servidor. Por lo tanto tiene que enviar un mensaje de error indicando que se tiene que verificar el usuario y password. Ver la figura 5.2.

FIGURA 5.2 Pantalla de error de validación de usuario y password.

En la figura 5.3 se observa la última validación se realiza con los formatos de los campos de las solicitudes (estas validación son las mismas que se aplican en las demás pantallas). Cuando el sistema detecta que no se ha cumplido con el formato adecuado descrito en las expresiones regulares mandará un mensaje indicando en que campo se encuentra el error y no efectúa cambios en la base de datos.

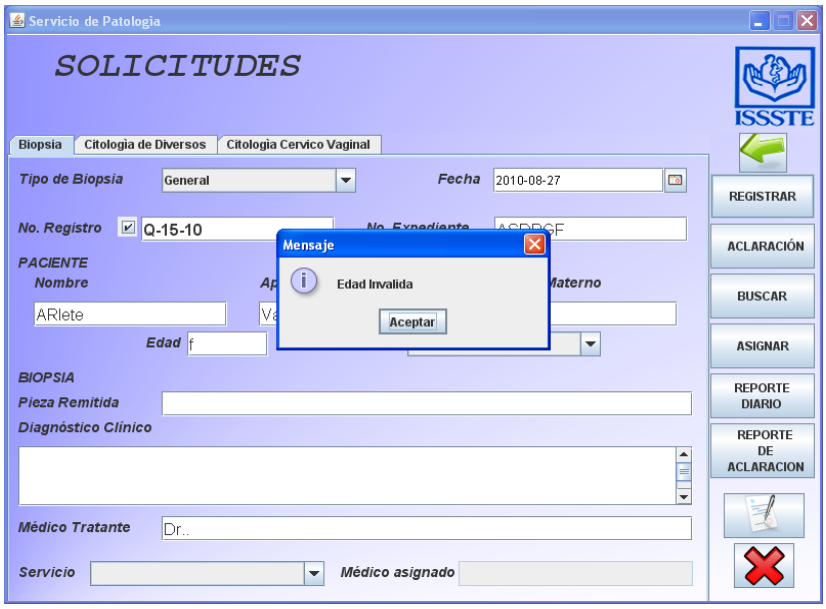


FIGURA 5.3 Pantalla de error de validación de campos.

- **PRUEBAS DE DISEÑO Y REQUERIMIENTO**

Cuando se realizó la prueba con el usuario final se encontraron ciertos detalles que no cumplían con las necesidades que requiere el usuario. Por lo tanto se señalan las siguientes modificaciones al sistema

- El usuario final indica que el tamaño de la pantalla de la aplicación debería de ser un poco más grande. Por esta razón se ajusta el código JAVA para que obtenga la medida de la pantalla donde se va a ejecutar la aplicación, y en base a estas medidas se ajusta la pantalla de la aplicación al igual que todos los componentes a través de porcentajes establecidos para tener un equilibrio con el tamaño de la pantalla. Con la siguiente función se determina el tamaño de la pantalla.

Dimension d=Toolkit.getDefaultToolkit().getScreenSize();

- La letra de los campos tenga un tamaño mayor de fuente y que resalte la información de ciertos campos. En la figura 5.4 se muestra un ejemplo de las modificaciones de la presentación de las letras de los campos.

FIGURA 5.4 Ejemplo de mejora en la letra de los campos.

- En la pantalla solicitud anatomopatológica se muestra el nombre del médico asignado, el cual no puede ser modificado. Lo anterior se puede observar en la figura 5.5.

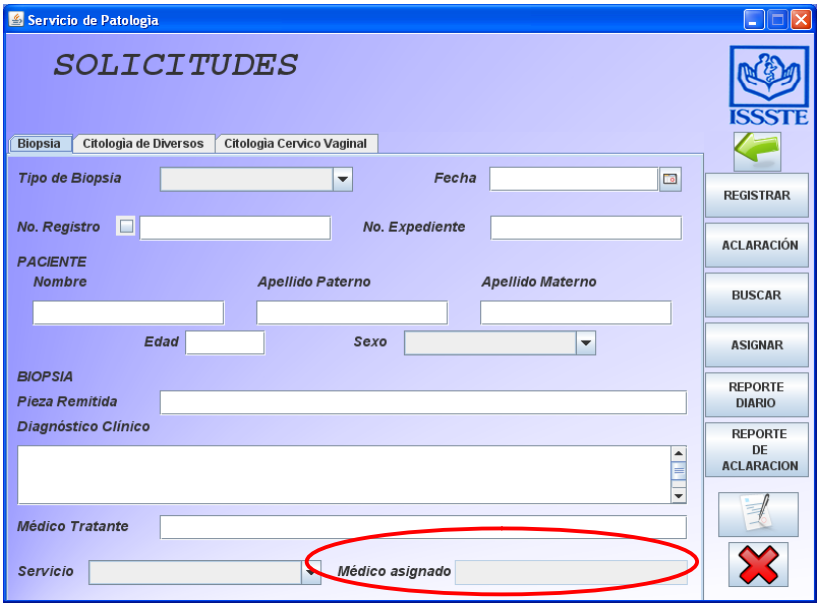


FIGURA 5.5 Pantalla que contiene el campo del médico asignado.

- Uno de los requerimientos es cambiar el estado de los patólogos e informes, el jefe del Departamento de Servicio indica que no está esa funcionalidad, por lo tanto se agrega a la pantalla de reportes estas acciones para la versión final. Véase en la figura 5.6.

FIGURA 5.6 Pantalla que contiene la funcionalidad para cambiar el estado de los patólogos e informes.

- Se realiza la modificación de la base de datos en la tabla INFORMES, agregando un atributo que guarde el estado en el que se encuentra cada informe (pendiente o finalizado). Esta modificación se justifica en la pantalla de solicitudes de informes, que se muestra en la figura 5.7, ya que cuando se selecciona la solicitud a la cual se requiere generar su informe se enlistan todas las solicitudes sin poder diferenciar aquellas que cuentan con un informe finalizado. La ventaja de realizar este cambio es que únicamente se enlistaran las solicitudes que no tengan ningún informe asociado o este en estado pendiente.

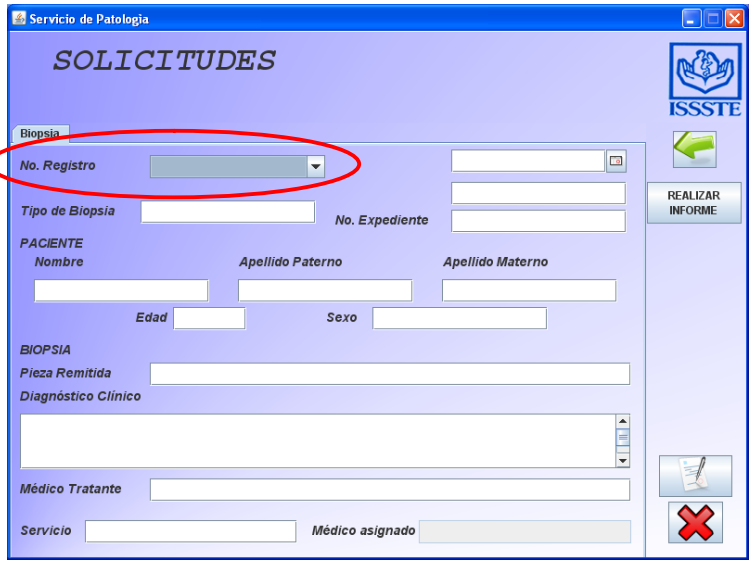


FIGURA 5.7 Pantalla que justifica las modificaciones de la base de datos.

- Se comprobó que se estén integrando las funcionalidades de la aplicación con las acciones hacia la base de datos. Estas pruebas se realizaron con los botones de la pantalla solicitudes, se encuentra que en el botón Registrar falta una validación que verifique que el sistema está generando el numero de solicitud que irá a la base de datos y no está siendo ingresado manualmente. Por lo tanto se verifica el funcionamiento del checkbox que está asignado al campo número de registro.

- Se verifica que las consultas realicen las funciones de inserción, eliminación, selección y modificación en la base de datos. Ya que se detectó que algunas de estas no se ejecutaban satisfactoriamente por cuestiones de errores de escritura de la sentencia (omisión de espacios) o porque faltaba alguna condición a considerar. En la figura 5.8 se muestra en forma de porcentaje la frecuencia en la que ocurría cada uno de los errores anteriores.

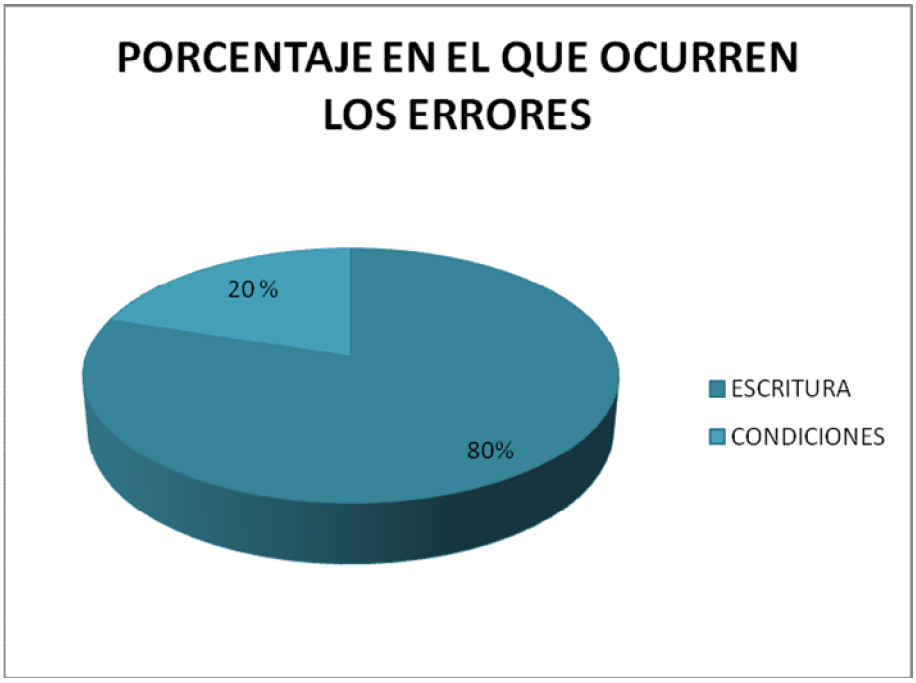


FIGURA 5.8 Gráfica que muestra el porcentaje en el que ocurren los errores.

PRUEBAS DE STRESS DE LA BASE DE DATOS

Se utiliza un programa llamado DB Stress (obtenido de *http://www.sqledit.com/stress/*), el cual se conecta con el manejador de base de datos para simular la actividad que tendrá al ejecutar la aplicación. La prueba se realizó al ejemplificar las tareas que con mayor frecuencia se ejecutan, que son el registro de solicitudes y generación de informes. Prueba de esto se muestra en la figura 5.9.

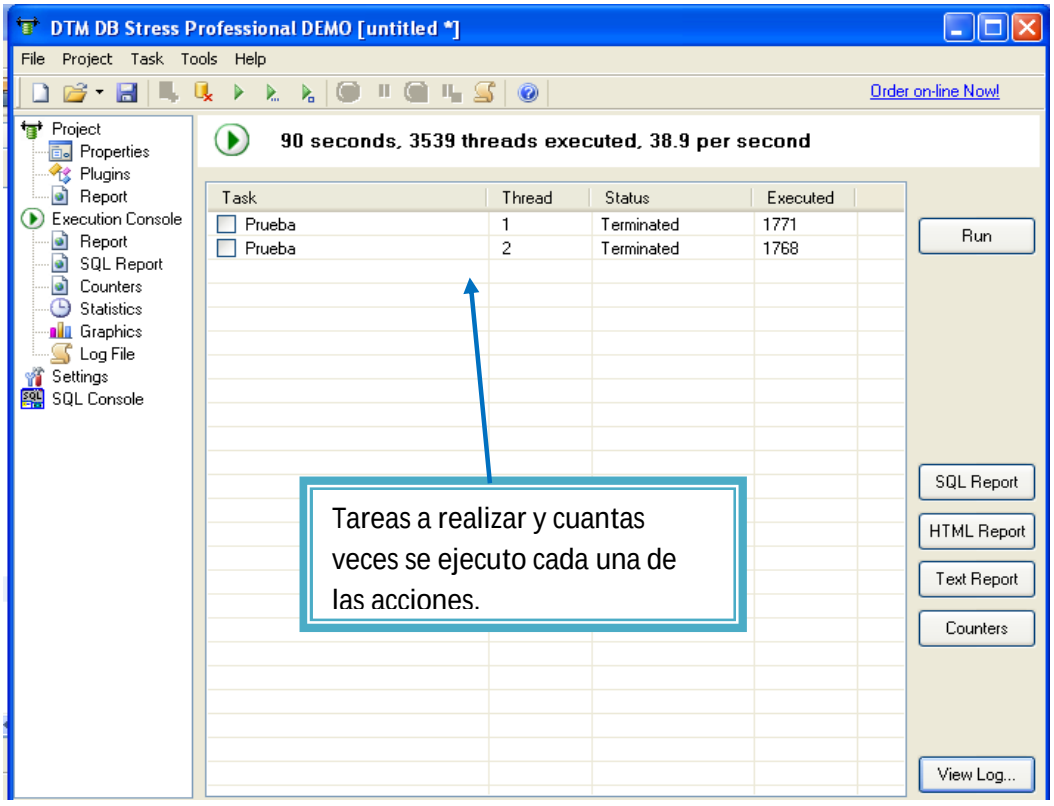


FIGURA 5.9 Prueba del funcionamiento del programa DB Stress.

Este tipo de programas es útil para saber el estado que tiene la base de datos, como las etapas en las que se desconecta el manejador y por cuento tiempo, el tiempo de ejecución de cada una de las tareas, etc.

Durante 90 segundos se ejecutan los procesos los procesos de registro de solicitudes y generación de informes. En la tabla 5.1 se muestra el número de iteraciones que se realizaron por cada proceso en dicho tiempo.

TAREA	NÚMERO DE ITERACIONES
Registrar solicitudes	1771
Generar informes	1768

TABLA 5.1 Número de iteraciones por proceso.

La gráfica correspondiente se encuentra en la figura 5.10.

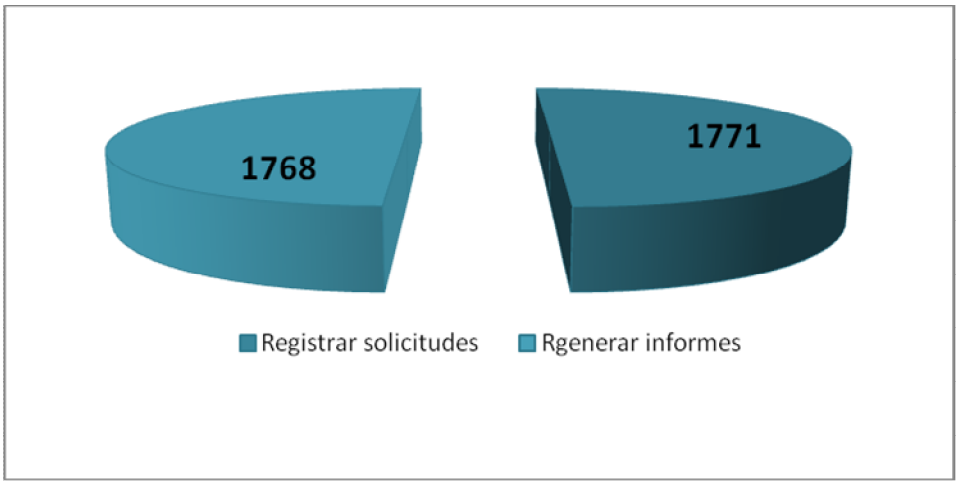


FIGURA 5.10 Gráfica del número de iteraciones por proceso.

El reporte de esta prueba de Stress se muestra en la tabla 5.2:

Velocidad en porcentaje que se tardo en ejecutar las iteraciones

Se observa que no tuvo ningún problema de conexión

Task Priority	Normal
Task Initialization Time, seconds	0.2970
Total Task Duration, seconds	174.4780

ThreadNo	Executed, times	Average Speed, iterations/second	Average Duration, seconds	Max Duration, seconds	Min Duration, seconds	Connect Time, seconds	Disconnect Time, seconds	Total Duration, seconds
1	1771	20.2845	0.0493	3.7800	0.0300	0.1560	0.0000	87.3080
2	1768	20.2822	0.0493	3.7640	0.0300	0.1410	0.0000	87.1700

TABLA 5.2 Resultados de la prueba de Stress realizada.

5.2. SEGURIDAD

A través de ésta sección se explican las configuraciones de seguridad implementadas en el sistema ***SIDIPAT***.

Para el ámbito de seguridad se debe considerar

- Seguridad física: Únicamente están autorizados a entrar al Departamento de Servicio de Anatomía Patológica el personal que labora en dicho Departamento. Por otro lado los equipos podrán ser ocupados por el personal administrativo, patólogos, citotecnólogos y el jefe del departamento.

- Seguridad de contraseña: Fueron creados por el administrador. Teniendo un formato especifico:

  - Debe tener una longitud de 8 caracteres
  - Debe contener un carácter numérico
  - Debe contener una letra mayúscula
  - No debe contener ninguna fecha importante
  - No debe contener ningún nombre de alguna persona o lugar.

  Cada usuario debe conservar su password en un lugar seguro, para tener mayor seguridad se cambiaran mensualmente los password. Ningún password será reutilizado para que no se pueda sustituir a ninguna persona.

- Seguridad de roles en el manejador de la base de datos: Únicamente el usuario del jefe del Departamento y postgres tienen el privilegio de superusuario. Los demás usuarios únicamente tiene los privilegios para realizar operación de insert, delete, select y update en ciertas tablas.

- Las estaciones de trabajo se configuran con SSL para tener una conexión segura de nuestra base de datos, a través de los siguientes pasos:

1. Se debe contar con una distribución de OpenSSL. La instalación de este programa se debe realizar en el directorio “C:\Archivos de programa\OpenSSL\” e incluir en la variable PATH “C:\Archivos de programa\OpenSSL\bin”.

2. Se edita el archivo postgresql.conf que se ubica en “C:\Archivos de programa\PostgreSQL\8.3\data” para permitir conexiones TCP/IP y SSL.

```
# - Connection Settings -

listen_addresses = '*'                  # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost', '*' = all
                                        # (change requires restart)
# - Security and Authentication -

authentication_timeout = 1min                           # 1s-600s
ssl = on                                                # (change requires restart)
ssl_ciphers = 'ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH'       # allowed SSL ciphers
                                                        # (change requires restart)
```

3. Crear la llave primaria del servidor (`server.key`) y el certificado SSL (`server.crt`). El procedimiento se muestra en la figura 5.11, también se explica en la tabla 5.3 los parámetros que se consideran en el archivo server.req.

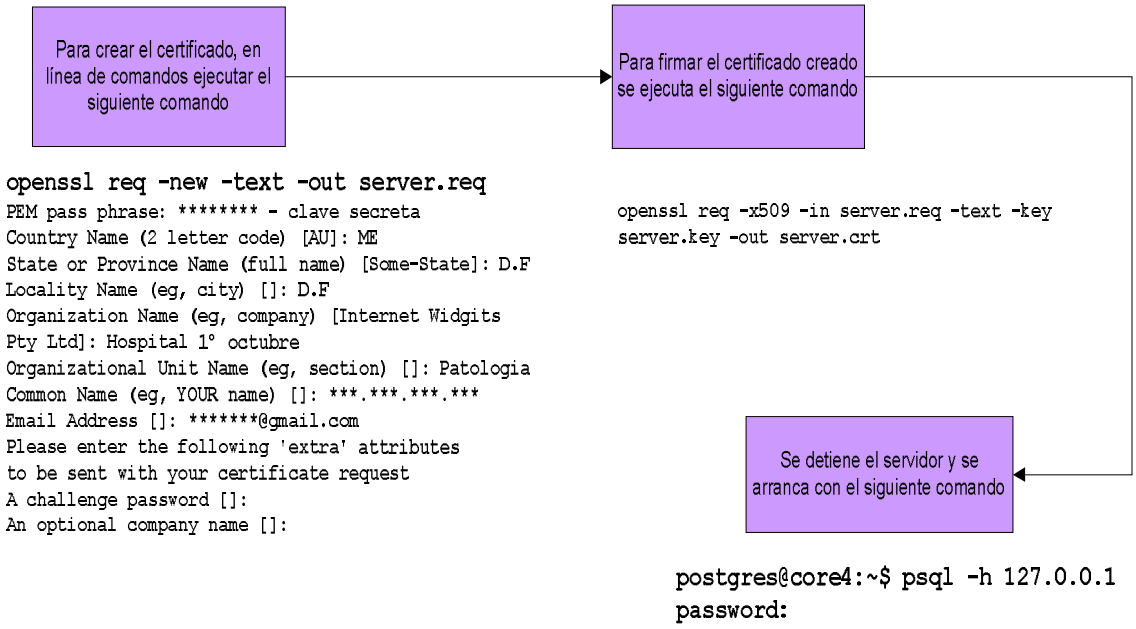


FIGURA 5.11 Procedimiento para la creación de la llave primario y del certificado SSL.

Parámetros del archivo server.req
1. La frase PEM es una contraseña de seguridad del certificado.
2. Los valores de los campos *Country Name, State Name, Locality Name, Organization Name, Organizational Unit Name e Email Address* son informativos.
3. El campo *Common Name* debe corresponder con el *hostname* de la máquina que tendrá el certificado.
4. El *Challenge Password y el Optional Company Name* pueden obviarse (en blanco).

TABLA 5.3 Explicación de los parámetros del archivo server.req

Si se desea remover la contraseña SSL se ejecuta el siguiente comando:

openssl rsa -in privkey.pem -out server.key
PEM pass phrase: clave_secreta
del privkey.pem

5. Para autenticar usuarios por medio del certificado creado con SSL

Para activar este tipo de autenticación se realiza con el método cert cuando se define el acceso en el archivo pg_hba.conf.

Lo primero que se tiene que hacer es crear un certificado para el usuario, atributo *Common Name*, y firmarlo con el certificado del servidor. En el archivo C:\Archivos de programa\PostgreSQL\8.3\data\pg_hba.conf se agrega la siguiente línea

```
hostssl  all     postgres     127.0.0.1/32     cert
```

Por último se reinicia el servidor para que tome los cambios realizados.

- Seguridad de la red inalámbrica: Como anteriormente ya se había descrito se utilizará el protocolo de seguridad WEP. Únicamente se podrán conectar a la red si cuentan con la contraseña de acceso.

- Seguridad de código de la aplicación: se utiliza un programa que nos permite ofuscar el código de la aplicación programada en JAVA.

5.3. MANTENIMIENTO

Cuando se pone en funcionamiento un sistema, éste debe de tener un proceso de mantenimiento continuo, con la finalidad de que la aplicación tenga un alto porcentaje de desempeño. En la figura 5.12 se valida que el mantenimiento es el proceso más costoso en el desarrollo de un sistema, ya que la continua revisión de la aplicación y sus componentes se hace de manera minuciosa para ver las posibles fallas o factores que pueden estar disminuyendo el desempeño de la aplicación.

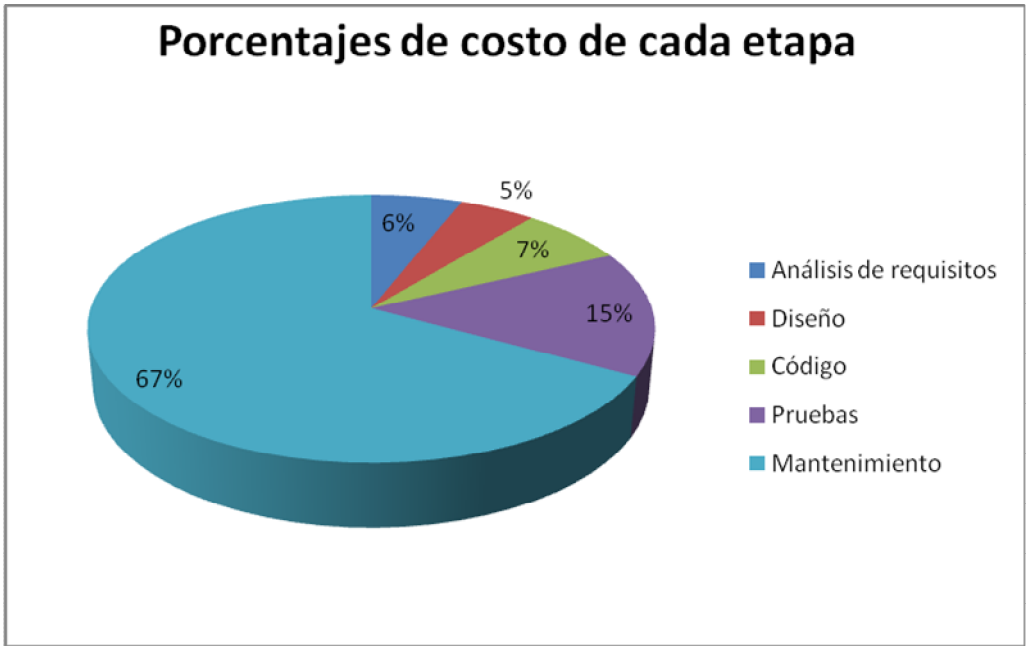


FIGURA 5.12 Gráfica que muestra los porcentajes de costo entre las diferentes etapas del desarrollo de una aplicación

Para realizar estas mejoras se debe tener en cuenta los siguientes puntos:

- Mantenimiento perfectivo: se efectúa un conjunto de actividades para mejorar o añadir nuevas funcionalidades requeridas por el Departamento de Servicio de Anatomía Patológica.
- Mantenimiento adaptativo: es el conjunto de actividades para adaptar el sistema a los cambios de hardware ó software, en cuanto a la base de datos es adicionar entidades, atributos ó relaciones.
- Mantenimiento correctivo: son las actividades dedicadas a corregir defectos en el hardware o en el software reportados por los usuarios durante la explotación del sistema. En la base de datos se realiza una recuperación de datos.
- Mantenimiento preventivo: conjunto de actividades para facilitar el mantenimiento futuro del sistema. Se realizan respaldos, se optimiza el código de la aplicación, etc.
- Control de permisos y privilegios de los usuarios
- Eliminar usuarios inactivos así como los passsword eliminados cada mes.
- Realizar una auditoría de la base de datos contemplando su integridad, su concurrencia, su consistencia, etc.
- Ejecutar una auditoria de seguridad con la finalidad de detectar puntos débiles dentro del sistema
- Tener reportes periódicos de la funcionalidad y productividad que está generando el sistema.

CONCLUSIONES

Nuestra recompensa se encuentra en el esfuerzo
y no en el resultado. Un esfuerzo total es
una victoria completa
Mahatma Gandhi

CONCLUSIONES

La información que se maneja en el Departamento de Servicio de Anatomía Patológica es muy importante y confidencial, ya que los informes de resultados de las biopsias o muestras ayudan a diagnosticar y ofrecer un tratamiento a los padecimientos que pueden presentar los pacientes.

Al conocer los procedimientos que se realizan en dicho departamento se detecta que no se está brindando un servicio de calidad al paciente por el tiempo que se lleva el personal en generar los informes de resultados. Por lo tanto, se propone implementar un sistema distribuido que permita mejorar el proceso de registro de solicitudes de biopsias y muestras.

Después del análisis, diseño e implementación del sistema distribuido en el Departamento de Servicio de Anatomía Patológica se determina que se han cumplido los objetivos generales señalados al inicio de este trabajo, los cuales se observan en las siguientes ventajas obtenidas:

- Ventajas de tiempo:

  Se disminuye el tiempo que tarda el personal administrativo en registrar una solicitud, dado que ya no tiene que vaciar la información en varias libretas para llevar un control mensual o semanal de las solicitudes entrantes.

  El proceso de entrega de informes de resultados se retrasaba porque no se tenía concentrada la información de tal forma, esto ocasionaba que el personal administrativo debía buscar al especialista que tuviera los resultados de alguna solicitud que se requería. Al tener consolidada toda la información en una base de datos distribuida el personal administrativo logra consultar los informes finalizados generados por los especialistas (patólogos y citotecnólogos) sin tener que moverse de su lugar, minimizando el tiempo de entrega.

- Ventajas de operación:

  Uno de los problemas que se presentaba era la repartición injusta del trabajo entre los patólogos en la operación diaria. Actualmente el sistema realiza una asignación aleatoria equitativa de las solicitudes de los estudios de biopsia, considerando únicamente a los patólogos que no se encuentran de vacaciones.

- Ventajas de seguridad:

  El sistema tiene la capacidad de generar el número de cualquier tipo de solicitud, con esto se regula que no se puedan ingresar biopsias o muestras externas a la Unidad Hospitalaria o perder alguna solicitud.

  Un punto fuerte del sistema es que cuenta con medidas de seguridad propias como son filtro de usuarios, *password* con ciertos estándares, uso de protocolos de seguridad de la red inalámbrica, etc. que protegen la información para que únicamente sea utilizada por el personal autorizado del Departamento de Servicio de Anatomía Patológica. Originalmente ésta información estaba en papel pudiendo leerla cualquier persona.

- Ventajas de control del proceso:

  El sistema permite al jefe de Departamento tener un mayor control del proceso de generación de informes de resultados:

  1. Puede obtener reportes que indican el estado en el que se encuentran los informes.
  2. Puede obtener reportes diarios con las solicitudes que fueron ingresadas ese día.
  3. Así como reportes que le permiten medir la productividad que cada patólogo o citotenólogo.

  Ahora, el jefe de Departamento es capaz de plantear estrategias que le permiten identificar y mejorar factores de atraso dentro del proceso de registro, actualización y documentación de informes de resultados de biopsias ó muestras.

- Ventajas en uso de espacios:

  Al utilizar el sistema se presentan ciertas ventas en la utilización de espacio, dado que se pueden almacenar los respaldos de la información en unidades de almacenamiento CD evitando el uso de biblioratos y papel para almacenar todas las solicitudes e informes de cada año.

Al realizar un análisis final de las mejoras que se observan, se logra disminuir la problemática que se presenta en el Departamento de Servicio de Anatomía Patológica un 40%, este porcentaje aproximado se obtiene de comparar el uso del sistema ***SIDIPAT*** contra el proceso que anteriormente se lleva en dicho Departamento durante un mes. Se detecta que el tiempo que se tarda en ejecutar el proceso completo para generar los informes de resultados, así como la eficiencia que presenta el personal del Departamento muestran adelantos importantes, por lo tanto de seguir utilizando el sistema ***SIDIPAT*** se observará un avance significativo en los servicios prestados por dicho Departamento. (Dichas mejoras están avaladas en el Oficio N° 090201/2.6.3./072/|10 proporcionado por el jefe del Departamento de Servicio de Anatomía Patológica que se encuentra anexo al final de esta sección).

Actualmente se han recibido propuestas de los directivos del Hospital Regional 1° de Octubre ISSSTE para implementar algunas funcionalidades del sistema ***SIDIPAT*** en otros Departamentos de Servicio que conforman dicho Hospital, ya que ha cumplido con los requerimientos manifestados y proporcionado una solución al problema de registro de solicitudes y generación de informes de resultados que presentaba el Departamento de Servicio de Anatomía Patológica.

De manera personal la optimización del proceso de registro de solicitudes y generación de informes, da como resultado una atención oportuna para el derechohabiente, la cual le puede salvar o mejorar la calidad de vida.

APÉNDICES

APÉNDICE A

GLOSARIO DE TÉRMINOS

El hombre nada puede aprender sino
en virtud de lo que sabe.
Aristóteles

APÉNDICE A GLOSARIO DE TÉRMINOS

.Net Es un componente de software que puede ser o es incluido en los sistemas operativos Microsoft Windows. Provee soluciones pre-codificadas para requerimientos comunes de los programas y gestiona la ejecución de programas escritos específicamente para este programa.

ACK *acknowledgement* un mensaje que informa que un grupo de datos ha llegado a su destino final sin ningún problema o error; así como también puede traer malas noticias indicando que los datos ha llegado con errores.

Algoritmo Michael invierte la clave MIC usada para proteger los paquetes enviados desde el Punto de acceso al cliente. Fue el más fuerte que los diseñadores de WPA pudieron crear, bajo la premisa de que debía funcionar en las tarjetas inalámbricas más viejas; sin embargo es susceptible a ataques.

Anatomopatológico Se trata de un examen que se realiza mediante el uso de un microscopio de las alteraciones a nivel celular utilizando para ello un pedazo de órgano del cuerpo (biopsia) o del órgano extirpado.

ANSI/SPARC *American National Standard Institute* grupo que acepta las recomendaciones del grupo de tarea de base de datos y aumentó los estándares de bases de datos en 1975 a través de su comité SPARC *Standards Planning and Requirements Committee* propuso una arquitectura de tres niveles para los sistemas de bases de datos.

API Grupo de rutinas que provee un sistema operativo, una aplicación o una biblioteca, que define como invocar desde un programa un servicio que éstos prestan. Es una interface de comunicación entre componentes de software.

Applets Componente de software escrito en un lenguaje de programación que se ejecuta bajo el control de una aplicación más grande que lo contiene.

Backend Es la parte que procesa la entrada de los datos hacia el *frontend* (interactúa con el usuario).

Backup una copia completa de la base de datos asegurando la recuperación completa de todos los datos en caso de un desastre físico o una falla de integridad de la base de datos.

Bases de datos es una estructura alojada en una computadora que aloja un conjunto de datos relacionados, contiene dos tipos de datos: datos para el usuario final y metadatos.

Bases de datos distribuidas guarda una base de datos lógicamente relacionada en más de dos sitios físicamente independientes.

Biopsia Es un fragmento de tejido de cualquier localización, extraído a una persona, que tratado con una técnicas adecuadas, puede estudiarse al microscopio para hacer un diagnostico correcto.

Broadcast un modo de transmisión de información donde un nodo emisor envía información a una multitud de nodos receptores de manera simultánea, sin necesidad de reproducir la misma transmisión nodo por nodo.

Bytecode Código intermedio entre el código fuente y el código maquina. Suele tratarse como un fichero binario que contiene un programa ejecutable similar a un modulo objeto.

Certificados X.509 Estándar para definir certificados digitales. Actualmente es una recomendación de la ITU, por lo tanto no ha sido aprobado para su uso estandarizado.

Citología: Es una muestra constituida por células desprendidas libremente de las superficies corporales o extraídas de diferentes regiones del organismo por medio de diversos procedimientos clínicos.

Citología cérvico-vaginal Es la muestra de células de la pared vaginal, el ecto y endo cervix.

Citología de diversos Es la muestra de células de boca, tracto digestivo, tracto respiratorio (esputo y cepillados o aspirados bronquiales), líquidos (orina, derrames, líquido cefalorraquídeo, ampollas cutáneas, etc.).

Citotecnólogo Realiza observación microscópica de muestras ginecológicas, distinguiendo imágenes normales y patología inflamatoria; somete a verificación casos sospechosos y de diagnóstico difícil.

Clave de sesión una clave que se crea para cifrar una comunicación entre dos entidades. La clave se establece de forma específica para cada sesión, terminando su utilidad una vez finalizada ésta.

Clave privada sirva para firmar nuestros mensajes y para descifrar los mensajes que nos han enviado cifrados.

Clave pública sirve que para que los demás puedan comprobar nuestra firma y para que nos envíen mensajes cifrados.

CLS (Especificación de lenguajes comunes) Es un conjunto de directivas proporcionadas por Microsoft para el desarrollo de lenguajes compatibles con .NET.

Computadoras incrustadas Tipo de computadora que tiene un propósito especifico, por ejemplo para controlar humedad o temperatura.

Conjunto de servicio básico (BSS) Está formado por dos o más nodos inalámbricos o estaciones que se han reconocido entre sí y que han establecido comunicaciones.

Conjunto de servicio básico independiente (IBSS) Es un modo de funcionamiento en un sistema 802.11 que permite la comunicación directa entre dispositivos 802.11 sin necesidad de establecer una sesión de comunicación con un punto de acceso.

Conjunto de servicio extendido (ESS) es un conjunto de dos o más conjuntos de servicio básico que forman una subred sencilla.

Correctitud Teoría de la computación que corresponde a una propiedad que distingue a un algoritmo de un procedimiento efectivo.

Criptosistemas Un conjunto finito denominado alfabeto, que permite representar tanto el texto en claro como el criptograma. A bajo nivel en bits, y a más alto nivel en caracteres ASCII o MIME.

CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance* intenta evitar colisiones utilizando un paquete explícito de reconocimiento (ACK), en donde un paquete ACK es enviado por la estación receptora confirmando que el paquete de datos llegó intacto.

Diccionario de datos Contiene las características lógicas de los datos que se utilizan en el sistema.

DDC catálogo de datos distribuidos, es un diccionario de datos que contiene la descripción (nombre de fragmentos, ubicaciones, replicaciones) de una base de datos distribuida.

DDL *Data Definition Language* permite crear objetos en la Base de datos.

DML *Data Manipulation Language* se refiere a los comandos que permiten manipular los datos, es decir, añadir consultar, borrar o actualizar.

DSSS secuencia directa de espectro ensanchado, donde los datos se transmite por multiplicación con un código de canalización hasta antes de la conversión a una frecuencia intermedia.

EAP *extensible authentication protocol* es una extensión del Protocolo punto a punto (PPP) que admite métodos de autenticación arbitrarios que utilizan intercambios de credenciales e información de longitudes arbitrarias. EAP se ha desarrollado como respuesta a la creciente demanda de métodos de autenticación que utilizan dispositivos de seguridad, como las tarjetas inteligentes, tarjetas de identificación y calculadoras de cifrado. EAP proporciona una arquitectura estándar para aceptar métodos de autenticación adicionales junto con PPP.

Eiffel Lenguaje de programación orientado a objetos centrado en la construcción de software robusto, creado por Bertrand Meyer en 1985.

Endoscopio Es un dispositivo acompañado de luz que se utiliza para mirar dentro de una cavidad u órgano corporal.

Firewall contrafuego que tiene como función el filtrar los intentos de establecimiento de conexión de forma que se pueda detectar e impedir el acceso al sistema a posibles intrusos sin que ni siquiera se haya llegado a establecer un enlace directo entre ellos.

Fragmentación proceso que divide una tabla en dos o más fragmentos. Los fragmentos de base de datos se tratan como una base de datos lógica única.

Framework Es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado.

Frontend Es la parte del software que interactúa con los usuarios.

Frotis Preparación para el microscopio hecha tomando parte de un tejido, membrana o líquido que se necesita examinar.

FTP Protocolo de transferencia de archivos, es ideal para transferir grandes bloques de datos por la red. Permite enviar o recibir cualquier tipo de archivos hacia o desde un servidor.

Grant Clausula para otorgar privilegios sobre un objeto de la base de datos.

Herramientas *CASE* Conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.

HTTP Protocolo de transferencia de hipertexto, es el método más común de intercambio de información en la world wide web, el método mediante el cual se transfieren las páginas web a una computadora.

IFS es el tiempo de interval entre tramas.

ICV *Integrity check value* es el control de integridad de valor que se entiende como el resultado del proceso de integridad. Esto normalmente implica el HMAC *(Hash Message Authentication Code*) y el algoritmo MD5.

Informix Es una base de datos transaccional para las prácticas actuales del ebusiness, que proporciona la infraestructura de datos critica necesaria para gestionar un negocio.

ISAM Método de Acceso secuencial indexado, método para almacenar información a la que se puede acceder rápidamente.

Itinerancia es un concepto utilizado en comunicaciones inalámbricas que está relacionado con la capacidad de un dispositivo para moverse de una zona de cobertura a otra.

Java Lenguaje de programación orientado a objetos, fue desarrollado por James Gosling y sus compañeros de Sun Microsystems al principio de la década de los 90.

JDBC (*Java Database Connectivity*) es un API para trabajar con bases de datos desde Java, independientemente de la base de datos a la que accedemos.

JSP Es una tecnología JAVA que permite a los desarrolladores de software generar dinámicamente HTML, XML u otros tipos de documentos, en respuesta al requerimiento de un cliente web.

JVM Aplicación que interpreta y ejecuta programas escritos en el lenguaje de programación JAVA. Específicamente puede interpretar el bytecode generado al compilar en JAVA.

Licencia BSD *Berkeley Software Distribution* Este tipo de licencia permite a los programadores utilizar, modificar y distribuir a terceros el código fuente y el código binario del programa de software original con o sin modificaciones.

Licencia GPL Licencia creada por la Free Sotfware Foundation y orientada principalmente a los términos de distribución, modificación y uso de software libre.

Lua Es un lenguaje de programación libre caracterizado por ser de extensión (no tiene un programa principal). Igualmente, puede usar funciones propias de C, lo que hace que Lua aumente considerablemente su potencial como lenguaje de programación.

MACA Acceso múltiple con prevención de colisiones, este algoritmo busca que el receptor envíe un marco pequeño, el cual sirva para indicar a las estaciones accesibles que va a estar ocupado. Posteriormente, inicia la transferencia de datos.

Metadatos se componen de datos sobre datos esto es, sus características y relaciones.

MIC mensaje integridad código proporción protección contra hackers.

MIMO es el método de utilización de múltiples antenas para comunicaciones inalámbricas.

Modelo de datos corporativo se refiere al modelo de todos los datos de la organización.

Modelo OSI *Open System Interconnection* fue creado por la Organización Estándar Internacional y en él pueden modelarse o referenciarse diversos dispositivos que reglamenta la ITU (Unión de Telecomunicación Internacional), con el fin de poner orden entre todos los sistemas y componentes requeridos en la transmisión de datos, además de simplificar la interrelación entre fabricantes. Así, todo dispositivo de cómputo y telecomunicaciones podrá ser referenciado al modelo y por ende concebido como parte de un sistema interdependiente con características muy precisas en cada nivel.

Multicast es el envío de información en una red a múltiples receptores de forma simultánea, un emisor envía un mensaje y son varios los receptores que reciben el mismo.

Multithread ó multihilo conjunto de varias sentencias que se ejecutan simultáneamente.

Normalización Es un proceso que clasifica relaciones, objetos, formas de relación y demás elementos en grupos, en base a las características que cada uno posee.

Object Pascal Es la evolución del lenguaje de programación Pascal, que incluye elementos del paradigma orientado a objetos. Es conocido como el lenguaje de programación de Borland Delphi, que modifica algunas características del Object Pascal original.

ODBC Es un estándar de acceso a bases de datos, que permiten mantener independencia entre los lenguajes de programación, los sistemas de bases de datos, y los sistemas operativos.

OFDM *orthogonal frequency-division multiplexing* es una técnica de modulación FDM que permite transmitir grandes cantidades de datos digitales sobre una onda de radio. Divide la señal de radio en muchas sub-señales que son transmitidas simultáneamente hacia el receptor en diferentes frecuencias.

Ondas electromagnéticas son ondas de radio cuya longitud de onda es superior a los 30 cm. Son capaces de recorrer grandes distancias, y pueden atravesar materiales sólidos. Son ondas multi-direccionales, es decir, se propagan en todas las direcciones. Su mayor problema son las interferencias entre usuarios. Estas ondas son las que emplean las redes inalámbricas.

Patología Parte de la medicina **que se ocupa del** estudio **de las enfermedades y del** conjunto **de sus síntomas.**

Patólogo Médico que examina los cuerpos y sus tejidos, que tiene la responsabilidad de realizar exámenes de laboratorio. Ayuda a otros médicos a hacer diagnósticos y es un miembro importante del equipo de tratamiento.

Perl Lenguaje de programación de scripts multiplataforma desarrollado por Larry Wall en 1987. Toma características del lenguaje C, del Shell, de AWK, Lisp, etc

PHP Lenguaje de programación usando generalmente en la creación de contenidos para sitios web. Este lenguaje interpretado especialmente usado para crear contenido dinámico web y aplicaciones para servidores, aunque también es posible crear aplicaciones gráficas utilizando la biblioteca GTK+.

Pike Es un lenguaje de programación orientado a objetos e interpretado, se asemeja a C y C++ pero su aprendizaje y uso son más sencillos. Se puede utilizar tanto para realizar pequeños scripts como grandes programas.

PITR *Point in Time Recovery* es un tipo de backup avanzado utilizado en sistemas PostgreSQL que trabajan con datos importantes los cuales no pueden perderse en caso de fallo.

PL/pgSQL (Procedural Language / PostgreSQL Structured Query Language) es un lenguaje procedimental cargable para el sistema de base de datos PostgreSQL. Permite efectuar comandos SQL mediante sentencias imperativas y uso de funciones para dar un mayor control automático que las sentencias SQL básicas.

Punto de acceso es un dispositivo que conecta dispositivos de comunicación inalámbrica entre sí para formar una red inalámbrica.

Python Lenguaje de programación desarrollado como proyecto de código abierto y es administrado por la empresa Python software Foundation.

Replicación Proceso de copiar y mantener objetos de la base de datos en múltiples bases de datos que forman un sistema de bases de datos distribuidos.

Revoke Instrucción que permite quitar un privilegio otorgado a un usuario o grupo.

Rollback Esto ocurre cuando hay algún tipo de fallo en una transacción de bases de datos, todos los datos regresan a su estado original.

RTS/CTS (Request to Send / Clear to Send) es el mecanismo usado por 802.11 protocolo de establecimiento de una red inalámbrica para reducir las colisiones del marco introducidas por problemas terminal.

Ruby Es un lenguaje de programación dinámico y de código abierto enfocado en la simplicidad y productividad.

Savepoints Es un punto de retorno dentro de una nueva transacción.

Servlets Pequeño programa que corre en un servidor. Por lo general son aplicaciones JAVA que corren en un entorno de servidor web.

SGBD software que actúa como intermediario entre el usuario y la base de datos. Traduce las solicitudes del usuario en código de computadoras requerido para atenderlas. Ayuda a los usuarios a administrar los datos guardados dentro de la base de datos.

Smalltalk es un lenguaje de programación que permite realizar tareas de computación mediante la interacción con objetos virtuales.

SQL92 Es el estándar ISO 9075 de sintaxis de SQL que se divide en tres partes Lenguaje de definición de datos, lenguaje de manipulación de datos y lenguaje de administración de datos.

SQL99 Es un estándar de SQL donde su antecesor es el SQL92. Incorporando nuevas características para realizar consultas de la base de datos.
C/C++ Lenguaje de programación diseñado a mediados de la década del 80 por Bjarne Stroustrup, como extensión del lenguaje C.

Tablespaces Es un grupo lógico de archivos de datos en una base de datos. Una base de datos suele contener al menos un tablespace, y por lo general dos o más. Dentro de la base de datos, el tablespace desempeña un papel similar al de una carpeta en el disco duro de una computadora.

TCL Es un lenguaje de programación de scripts creado por John Ousterhout en 1988.

Thread ó hilo es un único flujo de control dentro un programa.

TKIP *Temporary Key Integrity Protocol* es un protocolo de seguridad usado en WPA para mejorar el cifrado de datos en redes inalámbricas.

Transceptor es un dispositivo que realiza funciones tanto de envío como de recepción de señales, empleando elementos comunes del circuito para ambas funciones.

WAP son las siglas de protocolo de aplicaciones inalámbricas, un estándar seguro que permite que los usuarios accedan a información de forma instantánea a través de dispositivos inalámbricos. WPA es utilizado en redes inalámbricas para corregir deficiencias en el antiguo estándar de seguridad WEP.

WEP Privacidad Equivalente al Cable es el algoritmo opcional de seguridad para brindar protección a las redes inalámbricas, incluido en la primera versión del estándar IEEE 802.11, mantenido sin cambios en las nuevas 802,11a y 802.11b, con el fin de garantizar compatibilidad entre distintos fabricantes. El WEP es un sistema de encriptación estándar implementado en la MAC y soportado por la mayoría de las soluciones inalámbricas. En ningún caso es compatible con IPSec.

Write ahead logging Es un método estándar para que el registro de transacciones en disco se realice antes que el registro de los datos modificados, con el objetivo de asegurar la recuperación de datos.

APÉNDICES

APÉNDICE B

DICCIONARIO DE DATOS

Duda siempre de ti mismo,
hasta que los datos no dejen lugar a dudas.
Louis Kronenberg

APÉNDICE B DICCIONARIO DE DATOS

En este apéndice se encuentran las características de cada entidad del diagrama entidad – relación de la base de datos (véase el capítulo 3 – figura 3.1).

- C. CANCER

Catálogo que contiene cuando se realizó la última citología para detectar cáncer del Cervix.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
Id_cancer	integer	NOT NULL	Clave principal para las filas del catálogo cancer. El valor es autoincremental.
descripcion	character varying(30)	NOT NULL	Descripción de cuando se detecto cáncer del Cérvix.

TABLA B.1 Características del catálogo CANCER.

- C. CARACTERISTICAS_MUESTRA

Catálogo que contiene las características de la muestra que se va a estudiar.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
clave_carac_muestra	integer	NOT NULL	Representa la clave principal para el catálogo características_muestra, es un valor autoincremental.
descripcion	character varying(30)	NOT NULL	Contiene la descripción de la muestra que va a ser estudiada.

TABLA B.2 Características del catálogo CARACTERISTICAS_MUESTRA.

- CITOTECNOLOGOS

Entidad que contiene la información de los citotecnólogos.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
Id_citotec	integer	NOT NULL	Es un valor autoincremental, que representa la clave principal de la entidad citotecnologos
nombre	character varying(60)	NOT NULL	Nombre completo del citotecnólogo.
status	integer	NOT NULL	Indica si el citotecnólogo está activo o se encuentra de vacaciones.

TABLA B.3 Características de la entidad CITOTECNOLOGOS.

- C. DIAGNOSTICOS_CITOLOGICOS

Catálogo que contiene posibles diagnósticos citológicos para la muestra estudiada.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
clave_diagnostico	integer	NOT NULL	Representa la clave principal del catálogo diagnosticos_citologicos que es autoincremental.
descripcion	character varying(50)	NOT NULL	Señala la valoración final de la muestra.

TABLA B.4 Características del catálogo DIAGNOSTICOS_CITOLOGICOS.

- C. EXPLORACIONES

Catálogo que señala las características que puede tener la muestra cuando se realiza su exploración.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
Id_exploracion	integer	NOT NULL	Es un valor autoincremental que señala la clave principal del catálogo exploraciones.
descripcion	character varying(50)	NOT NULL	Característica de apariencia de la muestra de tipo cervico vaginal.

TABLA B.5 Características del catálogo EXPLORACIONES.

- HALLAZGOS

Entidad que contiene los hallazgos adicionales encontrados al finalizar el estudio de la muestra.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
clave_hallazgo	Integer	NOT NULL	Clave principal de la entidad hallazgos, es una valor autoincremental.
herpes	boolean	NULL	Indica si hay presencia de herpes.
tricomona	boolean	NULL	Señala si se detecto presencia de tricomona.
bacteria	Boolean	NULL	Indica si se encontraron bacterias.
hongo	boolean	NULL	Señala si se detecto algún hongo.
alteracion	Boolean	NULL	Indica que se encontraron otras alteraciones que no están incluidas en el informe.
descripcion	character varying(50)	NULL	Descripción de otras alteraciones encontradas.

TABLA B.6 Características de la entidad HALLAZGOS.

- GINECOBSTETRICOS

Entidad que contiene los antecedentes ginecobstetricos.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
clave_ginecobstetrico	integer	NOT NULL	Es un valor autoincremental, que representa la clave principal de la entidad ginecobstetricos.
puerpuerio	boolean	NULL	Indica si sufrió algún postaborto ó postparto.
diu	boolean	NULL	Señala si se uso el método anticonceptivo diu.
tratamiento_hormonal	boolean	NULL	Señala si la paciente tomo algún tratamiento hormonal.
tratamiento_ginecologico	boolean	NULL	Indica si la paciente estuvo en otros tratamientos wginecológicos.
descripción_ginecologica	character varying(30)	NULL	Nombre de los tratamientos ginecológicos.
embarazo	boolean	NULL	Indica si la paciente ha estado embarazada.
post_menopausia	boolean	NULL	Señala si la paciente se encuentra en la etapa de postmenopausia.
histerectomia	boolean	NULL	Señala si hubo presencia de histerctomia.
fecha_regla	date	NOT NULL	Fecha de la última regla.

TABLA B.7 Características de la entidad GINECOBSTETRICOS.

- INF_ANATOMOPATOLGICOS

Entidad que contiene la información que compone a los informes anatomopatologicos.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
id_inf_anato	integer	NOT NULL	Valor autoincremental que representa la clave principal de la entidad inf_anatomopatologicos.
descripcion_macroscopica	Text	NULL	Características de la biopsia a estudiar (color, textura, etc.).
diagnostico_histopatologico	Text	NULL	Descripción de la valoración hecha a la biopsia.

TABLA B.8 Características de la entidad INF_ANATOMOPATOLOGICOS.

- INF_CITOLOGICOS

Entidad que contiene toda la información que compone a los informes citológicos.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
id_info_cito	Integer	NOT NULL	Representa la clave principal de la entidad inf_citologicos, el cuál es un valor autoincremental.
muestra	character varying(150)	NULL	Tipo y características de la muestra (liquido, laminillas, etc.)
descripción_microscopica	text	NULL	Descripción de la muestra vista desde el microscopio.
diagnostico_citologico	text	NULL	Valoración de la muestra después de su estudio.
estudio_revision	boolean	NULL	Indica si se requiere repetir el estudio de la muestra.
motivo	character varying(50)	NULL	Describe la razón por la que se solicita repetir el estudio.
patologo_revision	boolean	NULL	Indica si la muestra requiere ser revisada por un patólogo.
observacion	text	NULL	Describe características adicionales que encontró el citotecnólogo.
tipo	character varying(40)	NULL	Describe el tipo de adenocarcinoma.

TABLA B.9 Características de la entidad INF_CITOLOGICOS.

- INFORMES

Entidad donde se guarda la fecha y el status de los informes.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
id_informe	Integer	NOT NULL	Valor autoincremental que representa la clave primaria de la entidad informes.
fecha_informe	date	NOT NULL	Fecha en la que se genera el infrome.
status	boolean	NOT NULL	Indica si el informe esta en estado pendiente o finalizado.

TABLA B.10 Características de la entidad INFORMES.

- C. METODOS_OBTENCION

Catálogo que contiene los posibles métodos de obtención de la muestra.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
id_metodo	Integer	NOT NULL	Clave principal del catálogo método_obtencion, este valor es autoincremental.
descripcion	character varying(50)	NOT NULL	Indica el nombre del método que se utilizo para adquirir la muestra.

TABLA B.11 Características del catálogo METODOS_OBTENCION.

- C. MUESTRAS_PROCE

Catálogo que presenta las posibles procedencias de la muestra.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
id_muestra_pro	Integer	NOT NULL	Valor autoincremental que representa la clave principal del catálogo muestra_proce.
descripcion	character varying(50)	NOT NULL	Indica la procedencia de la muestra que se va a estudiar.

TABLA B.12 Características del catálogo MUESTRAS_PROCE.

- PACIENTES

Entidad que almacena los datos de los pacientes.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
id_paciente	Integer	NOT NULL	Representa la clave principal de la entidad pacientes, el cuál es un valor autoincremental.
num_expediente	character varying(150)	NOT NULL	Número de afiliación
nombre	text	NOT NULL	Nombre del paciente (primer y segundo nombre)
apellido_paterno	text	NOT NULL	Apellido paterno del paciente.
apellido_materno	boolean	NOT NULL	Apellido materno del paciente.
sexo	character varying(50)	NOT NULL	Sexo del paciente (Femenino ó Masculino).
edad	boolean	NOT NULL	Edad del paciente.

TABLA B.13 Características de la entidad PACIENTES.

- PATOLOGOS

Entidad que almacena los datos y el estado de los patólogos.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
Id_patologo	integer	NOT NULL	Es un valor autoincremental, que representa la clave principal de la entidad patologos.
nombre1	character varying(60)	NOT NULL	Nombre completo del patólogo.
status	integer	NOT NULL	Indica si el patólogo está activo o se encuentra de vacaciones.

TABLA B.14 Características de la entidad PATOLOGOS.

- C. PRODUCTOS

Catálogo que presenta los diferentes productos para ser estudiados.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
clave_producto	Integer	NOT NULL	Valor autoincremental que representa la clave principal de la entidad productos.
descripcion	character varying(50)	NOT NULL	Indica el nombre del producto que se estudiara.

TABLA B.15 Características del catálogo PRODUCTOS.

- REGISTRO

Entidad auxiliar que contiene los valores necesarios para establecer el número de registro de la solicitud, así como el médico que será asignado.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
id	Integer	NOT NULL	Representa la clave principal de la entidad auxiliar de registro.
reg_general	integer	NOT NULL	Número de la última solicitud anatomopatológica general.
reg_onco	integer	NOT NULL	Número de la última solicitud anatomopatológica oncológica.
reg_cito	integer	NOT NULL	Número de la última solicitud citológica.
medico	integer	NOT NULL	Número que representa el nombre del patólogo asignado a la última solicitud anatomopatologica.

TABLA B.16 Características de la entidad REGISTRO.

- C. SERVICIOS

Catálogo que almacena los nombres de los Departamentos de Servicio.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
clave_servicio	Integer	NOT NULL	Valor autoincremental que representa la clave principal de la entidad servicios.
descripcion1	character varying(35)	NOT NULL	Indica el nombre del Departamento de Servicio.

TABLA B.17 Características del catálogo SERVICIOS.

- C. SINTOMAS

Catálogo que almacena los posibles síntomas que puede presentar el paciente.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
id_sintoma	Integer	NOT NULL	Clave principal de la entidad síntomas que es un valor autoincremental.
descripcion	character varying(50)	NOT NULL	Característica del síntoma que presenta actualmente el paciente.

TABLA B.18 Características del catálogo SINTOMAS.

- SOL_ANATOMOPATOLOGICAS

Entidad que almacena la información de las solicitudes anatomopatólogicas.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
id_anatomopatologica	integer	NOT NULL	Valor autoincremental que representa la clave principal de la entidad sol_anatomopatologicas.
num_registro	character varying(12)	NOT NULL	Número consecutivo que representa a la solicitud. (Formato Q-1-10 ó Q-O-3-10)
pieza_remitida	character varying(50)	NOT NULL	Descripción de la pieza recibida.
diagnostico_clinico	character varying(150)	NOT NULL	Diagnóstico previo de la muestra a estudiar.
medico_tratante	character varying(100)	NOT NULL	Nombre completo del médico que solicita el estudio.

TABLA B.19 Características de la entidad SOL_ANATOMOPATOLOGICAS.

- SOL_CITOLOGICAS

Entidad que contiene la información de las solicitudes citológicas.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
id_cito	integer	NOT NULL	Valor autoincremental que representa la clave principal de la entidad sol_citologicas.
num_citologico	character varying(12)	NOT NULL	Número consecutivo que representa a la solicitud. (Formato C-1-10)
nom_resp_muestra	character varying(70)	NOT NULL	Nombre completo del médico que solicita el estudio.

TABLA B.20 Características de la entidad SOL_CITOLOGICAS.

- SOLICITUDES

Entidad que contiene las características principales de las solicitudes.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
id_solicitudes	integer	NOT NULL	Clave principal de la entidad solicitudes. Es un valor autoincremental.
fecha_solicitud	date	NOT NULL	Fecha en la que se registro la solicitud.
medico_responsable	character varying(40)	NULL	Nombre completo del patólogo responsable del estudio de la muestra. Únicamente si es una solicitud anatomopatológica.
aclaracion	boolean	NOT NULL	Indica si la solicitud requiere ser revisada nuevamente.
statusinfo	integer	NOT NULL	Señala si la solicitud ya está relacionada con un informe. (1- SI, 0- NO)

TABLA B.21 Características de la entidad SOLICITUDES.

- C. TIPO_ANATOMOPATOLOGICAS

Catálogo que almacena los diferentes tipos de solicitudes anatomopatológicas.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
id_tipo_anatomo	integer	NOT NULL	Valor autoincremental que representa la clave principal del catálogo tipo_anatomopatolgicas.
descripcion	character varying(30)	NOT NULL	Nombre del tipo de solicitud e informe anatomopatologico. (General ó Oncológica)

TABLA B.22 Características del catálogo TIPO_ANATOMOPATOLOGICAS.

- C. TIPO_CITOLOGICAS

Catálogo que contiene los tipos de solicitudes citológicas.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
clave_tipo_cito	integer	NOT NULL	Representa la clave principal del catálogo tipo_citologicas. Valor autoincremental.
descripcion	character varying(30)	NOT NULL	Nombre del tipo de solicitud e informe citológico. (Diversos ó Cervico Vaginal)

TABLA B.23 Características del catálogo TIPO_CITOLOGICAS.

- C. TIPO_SOL_INF

Catálogo que almacena los tipos de solicitudes e informes.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
clave_tipo	integer	NOT NULL	Valor autoincremental que representa la clave principal del catálogo tipo_sol_inf.
descripcion	character varying(30)	NOT NULL	Nombre del tipo de solicitud o informe. (Anatomopatológico ó Citológico)

TABLA B.24 Características del catálogo TIPO_SOL_INF.

- UNIDADES_MEDICAS

Entidad que almacena la ubicación y el nombre de las unidades medicas.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
id_unidad	integer	NOT NULL	Valor autoincremental que representa la clave principal de la entidad unidades_medicas.
lugar	171character varying(30)	NOT NULL	Ubicación de la unidad medica (D.F, provincia, etc)
tipo_unidad	character varying(30)	NULL	Nombre de la unidad medica (20 de noviembre,Naucalpan, etc)

TABLA B.25 Características de la entidad UNIDADES_MEDICAS.

- C. UTENSILIOS

Catálogo que almacena los nombres de los utensilios que se ocupan para tomar la muestra.

COLUMNA	TIPO DE DATOS	NULABILIDAD	DESCRIPCIÓN
id_utensilio	integer	NOT NULL	Valor autoincremental que representa la clave principal del catálogo utensilios.
descripcion	171character varying(50)	NOT NULL	Nombre del utensilio utilizado para tomar la muestra.

TABLA B.26 Características del catálogo UTENSILIOS.

APÉNDICES

APÉNDICE C

MANUAL TÉCNICO

No hay que empezar siempre por la noción primera
de las cosas que se estudian, sino por aquello
que puede facilitar el aprendizaje.
Aristóteles

APÉNDICE C MANUAL TÉCNICO

REQUERIMIENTOS

El sistema ***SIDPAT*** tiene ciertos requerimientos de software y hardware que debe cumplir para tener un buen funcionamiento. A continuación se mencionan:

- Requerimientos de software

En los requerimientos de software se contempla el sistema operativo que es soportado, así como la paquetería que debe de instalarse.

Sistema Operativo: Windows XP, Windows Vista y Windows 7.

Manejador de bases de datos: PostgreSQL 8.3.10-1 para Windows

JAVA: jre-6u21-windows-i586

Adobe Reader cualquier versión (se recomienda tener la versión 9 ó 8)

Librerías de JAVA: iText-5.0.2, jcalendar-1.3.3, looks-2.0.1

Driver para conectar JAVA con el manejador de bases de datos: JDBC para PostgreSQL (postgresql-8.4-701.jdbc4)

- Requerimientos de hardware

En el caso de los requerimientos de hardware se señalan las capacidades necesarias que deben cumplir los equipos que están dentro de la red de datos inalámbrica.

Memoria RAM mínima: 256 MB

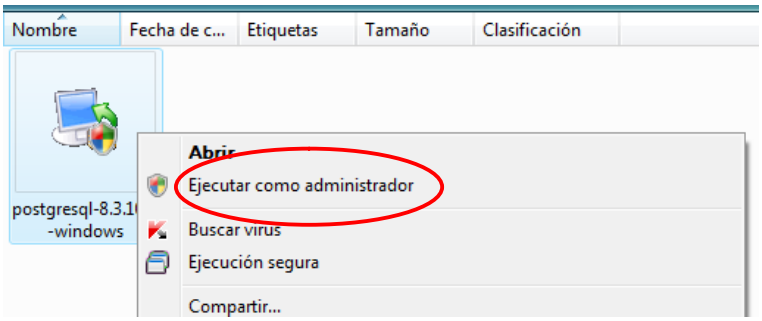
Tamaño mínimo de disco duro: 20 GB

Impresora: Compatible con la computadora que se encuentra en el nodo RESULTADOS

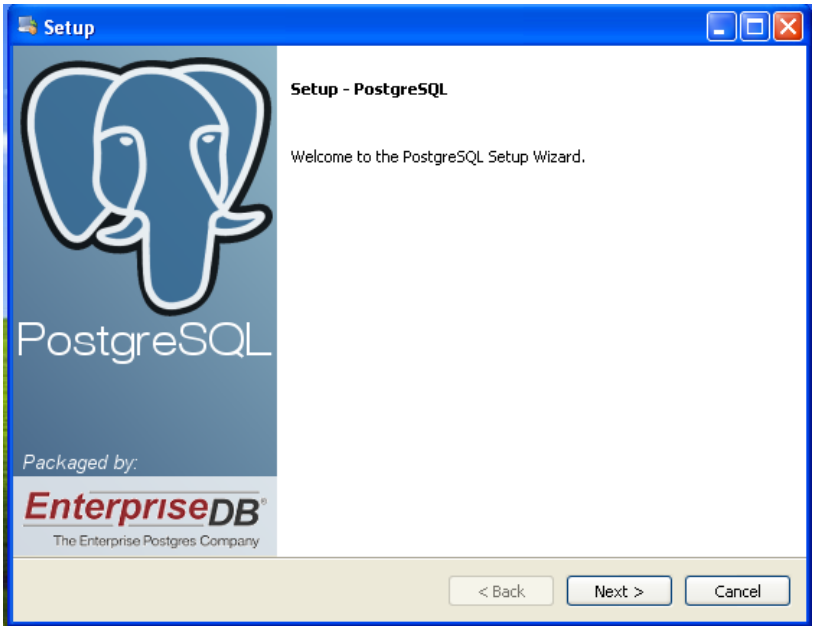
INSTALACION DE POSTGRESQL

Para llevar a cabo la instalación correcta de postgreSQL en cualquier versión de Windows soportada por el sistema ***SIDIPAT*** debe seguir los pasos que a continuación se mencionan:

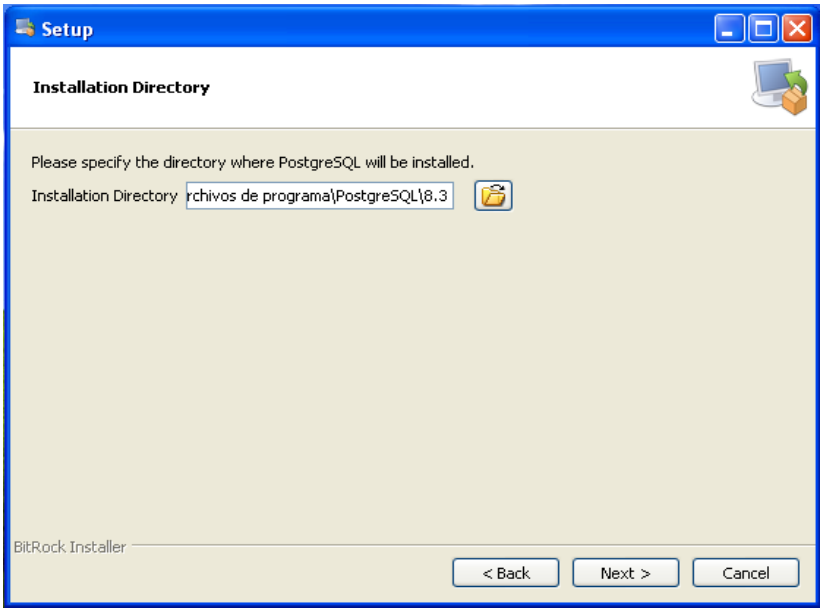
1. Dar clic derecho sobre el ejecutable de postgreSQL y elegir "Ejecutar como administrador". Se recomienda realizar esta para que no ocurra ningún problema en Windows Vista o Windows 7.



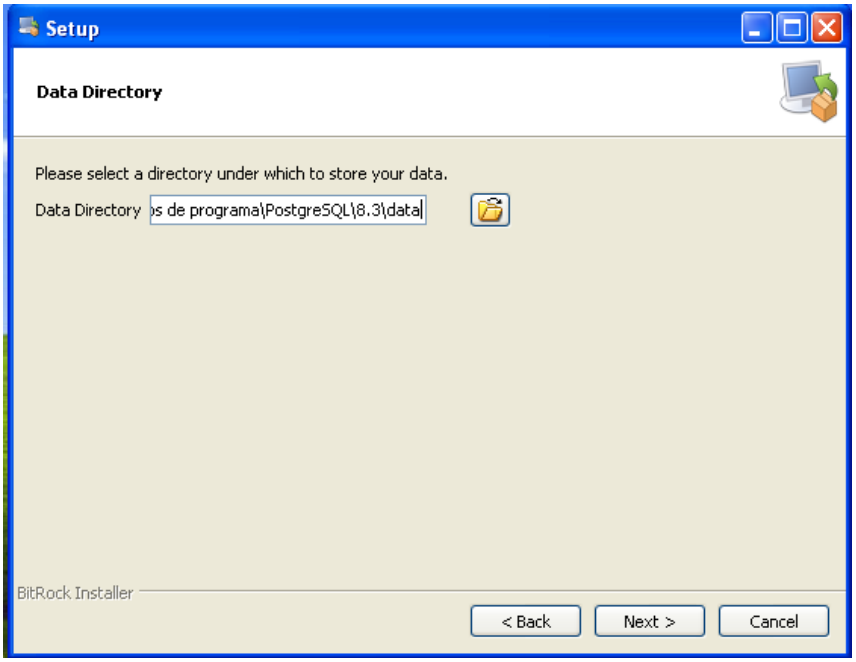
2. En la pantalla que aparece dar clic en el botón "Next".



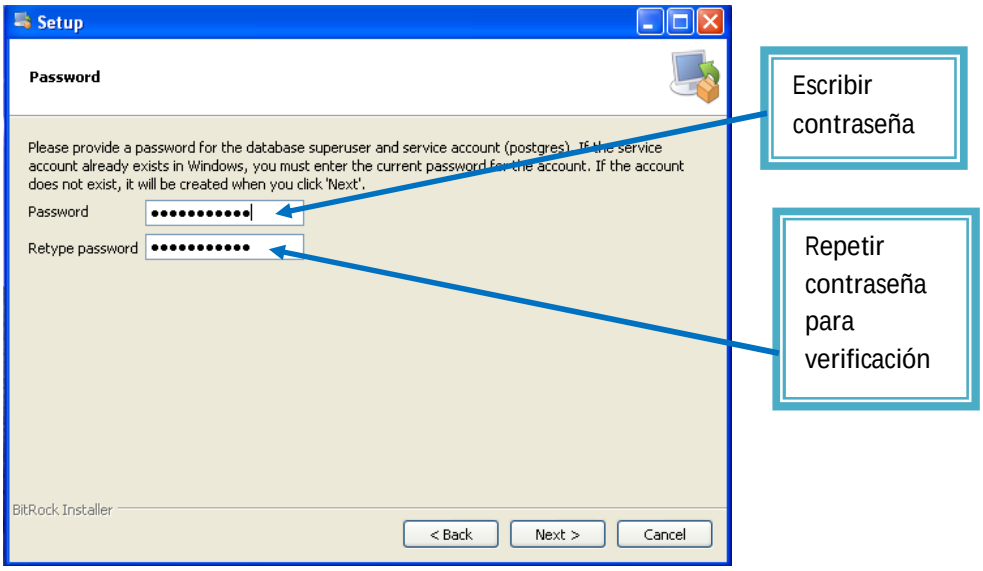
3. Se indica en que carpeta se guardará el archivo. Dar clic en el botón “Next”. Se recomienda dejar la que aparece por default.



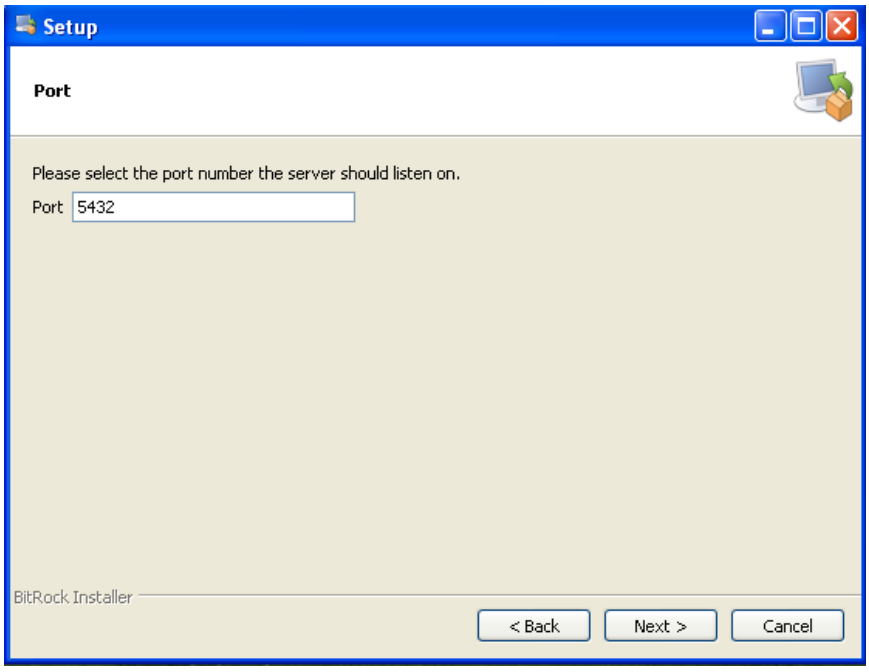
4. En la siguiente pantalla se elije la carpeta donde se guardará la información generada. Se recomienda dejar la que viene por default. Dar clic en el botón “Next”.



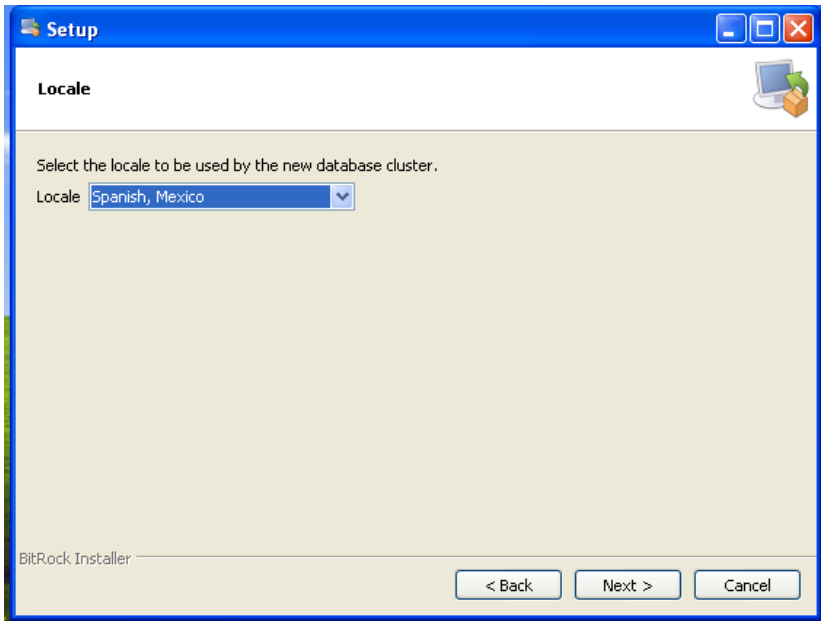
5. Insertar una contraseña para la cuenta que se creara de postgres. Se pide que la repita para validar que sean iguales. Después dar clic en el botón “Next”.



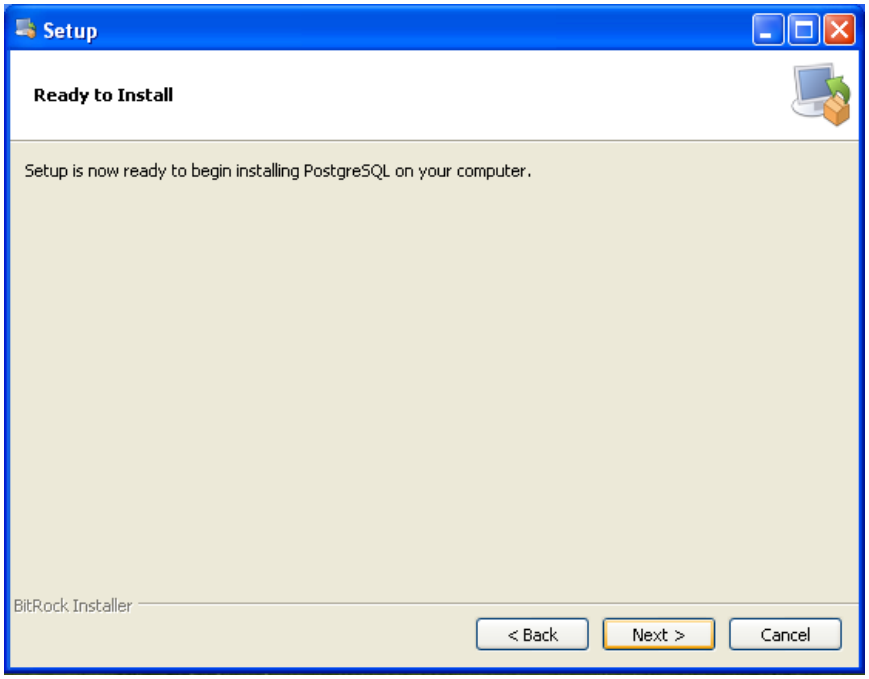
6. En la pantalla siguiente se selecciona el puerto para el servidor. Se puede elegir cualquier puerto pero se recomienda dejar el que viene por default). Dar clic en el botón de “Next”.



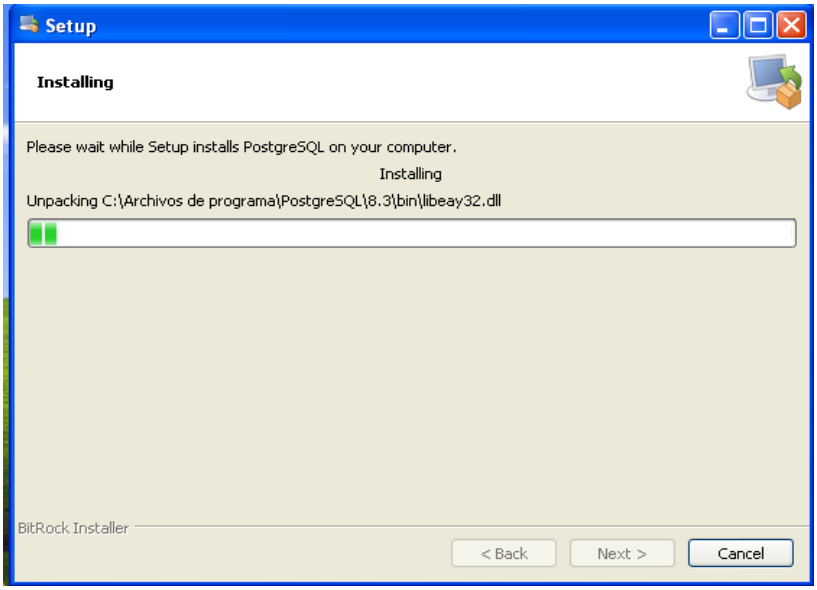
7. Posteriormente se selecciona la ubicación para el cluster. Se pide que elija el lugar de residencia en la lista (se muestra primero el idioma y después el país). Dar clic en el botón "Next".



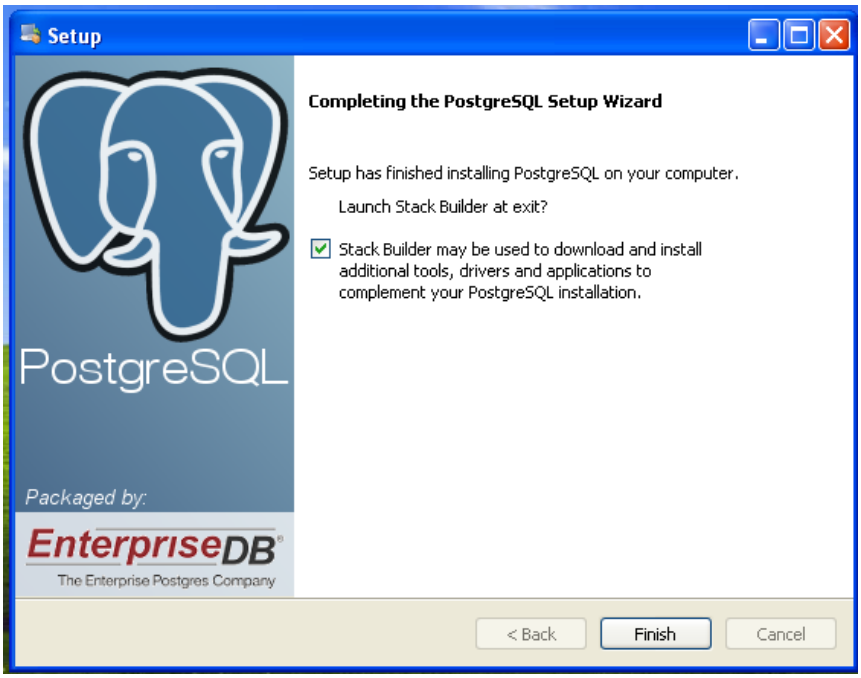
8. Dar clic en el botón "Next" para comenzar la instalación.



9. Inicia el proceso de instalación.

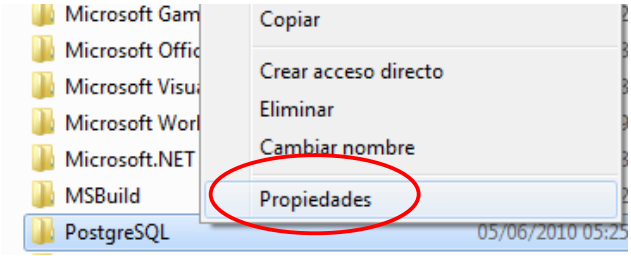


10. Se termina el proceso de instalación, dar clic en el botón “Finish”.

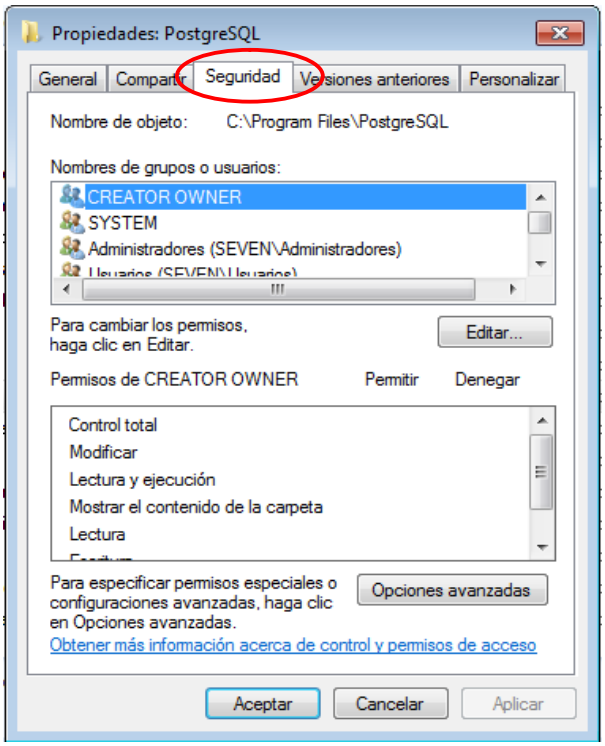


Para Windows Vista y Windows 7 se requiere cambiar los permisos de la carpeta donde se instaló el programa.

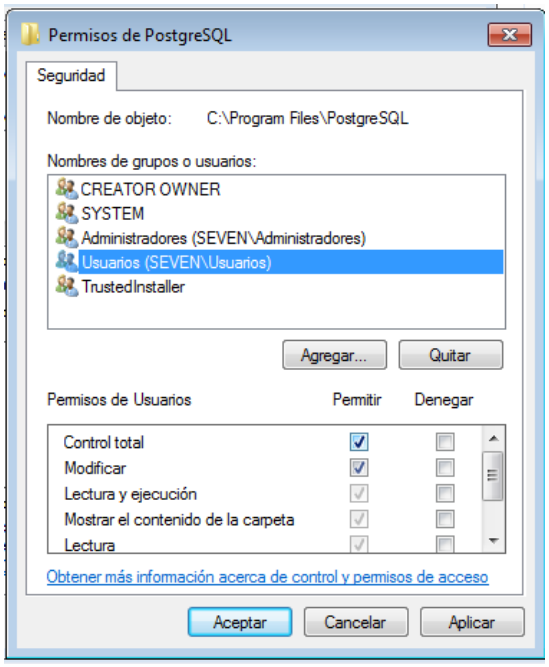
1. En la carpeta de "C:/Archivos de Programas/PostgreSQL" dar clic derecho y elegir la opción de "Propiedades".



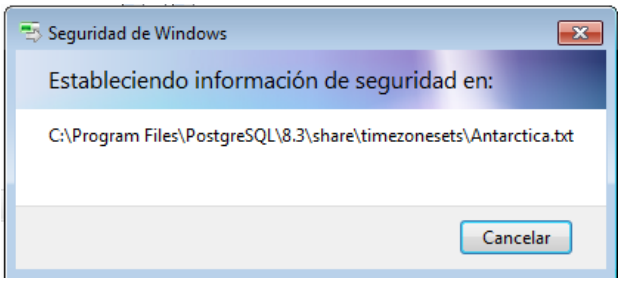
2. En la pantalla que aparece elegir la pestaña de "Seguridad". Dar clic en el botón "Editar".



3. Elegir al usuario “Usuarios” y darle permisos de control total sobre la carpeta. Dar clic en el botón “Aplicar”.



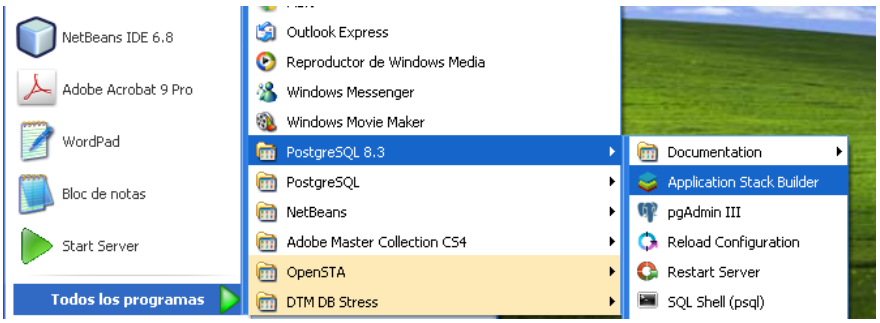
4. Se ejecuta el proceso de realizar los cambios a la carpeta. Dar clic en el botón “Aceptar” hasta que se cierren todas las pantallas.



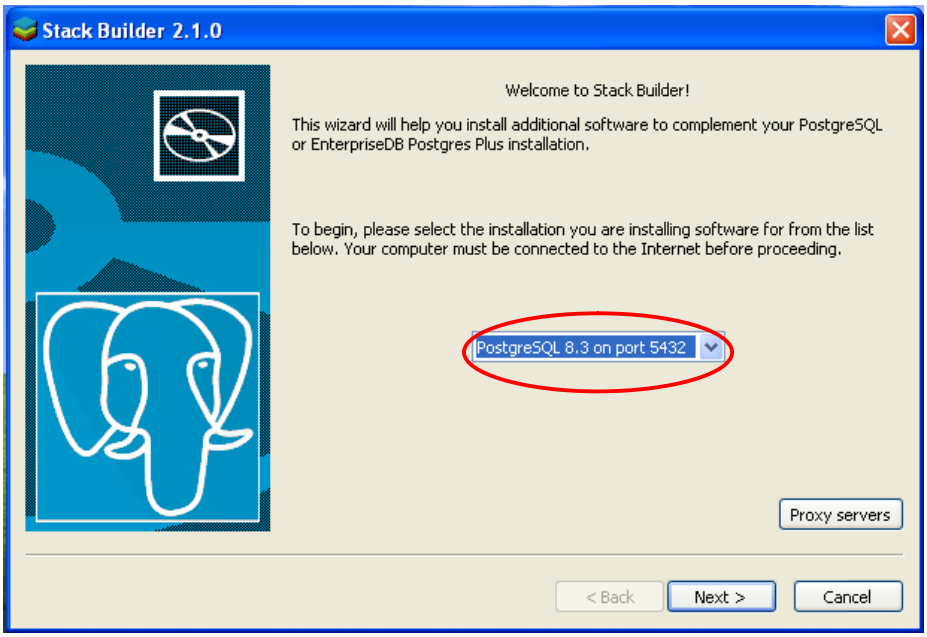
INSTALACION DE SLONY –I

Para la instalación de la aplicación de replicación SLONY – I se utiliza el Stack Builder de PostgreSQL realizando los siguientes pasos:

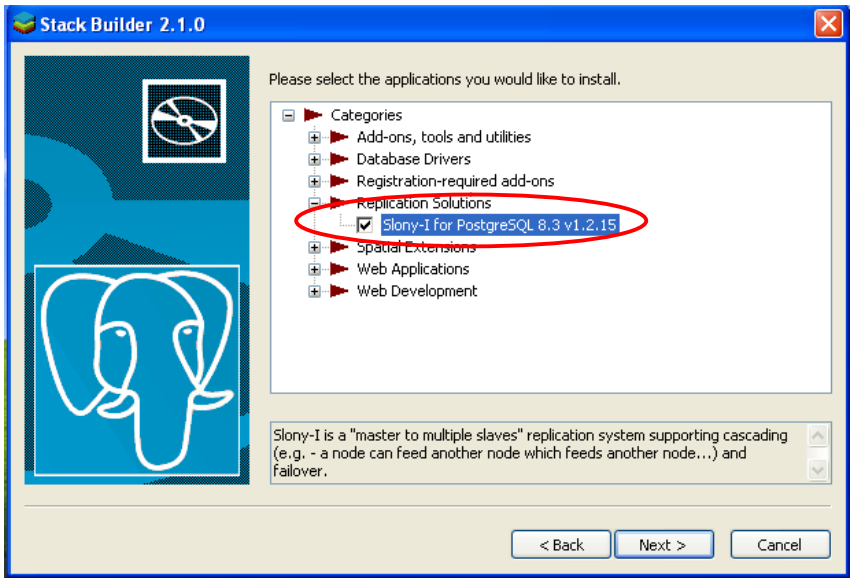
1. Posicionarse en Inicio -> Todos los programas -> PostgreSQL -> Aplication Stack Builder



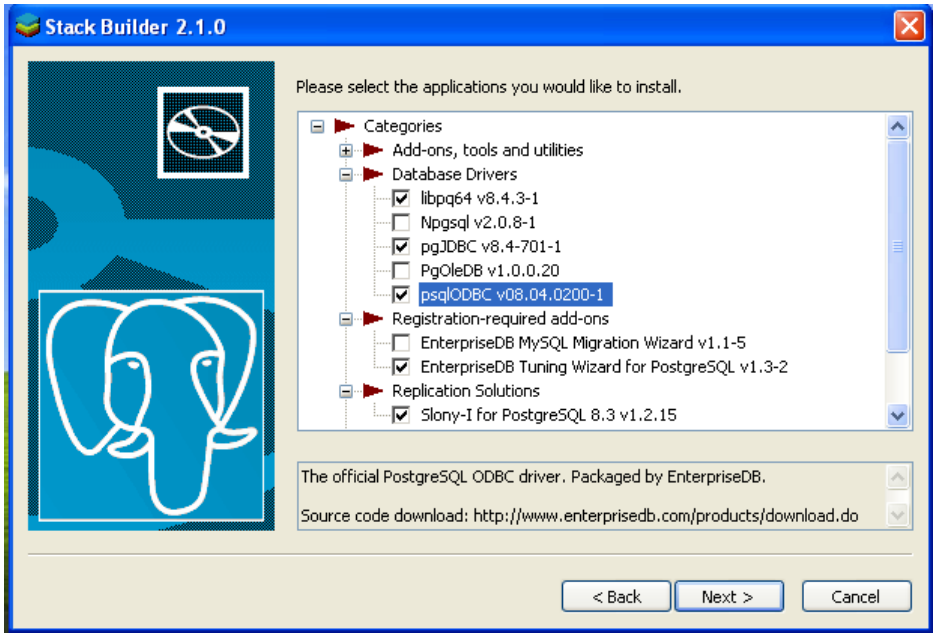
2. Se muestra una pantalla de bienvenida a la aplicación y se elije el servidor al cual se desea instalar la aplicación.



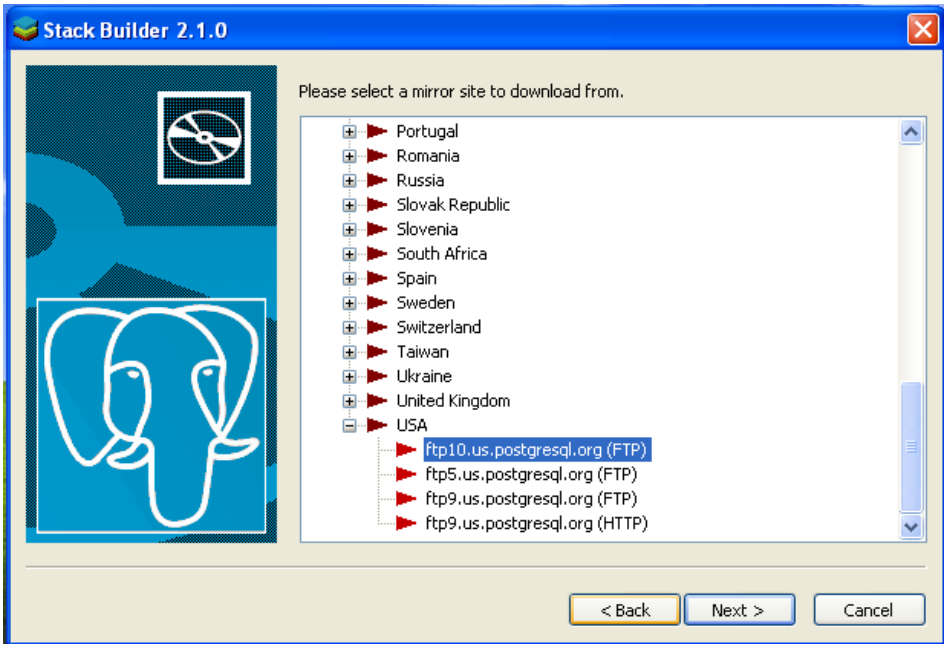
3. En la siguiente pantalla aparece una lista donde se selecciona la opción de “Replications Solutions”, se despliega una lista de la cual se elegirá la opción de “Slony –I for PostgreSQL 8.3 v1.2.15”.



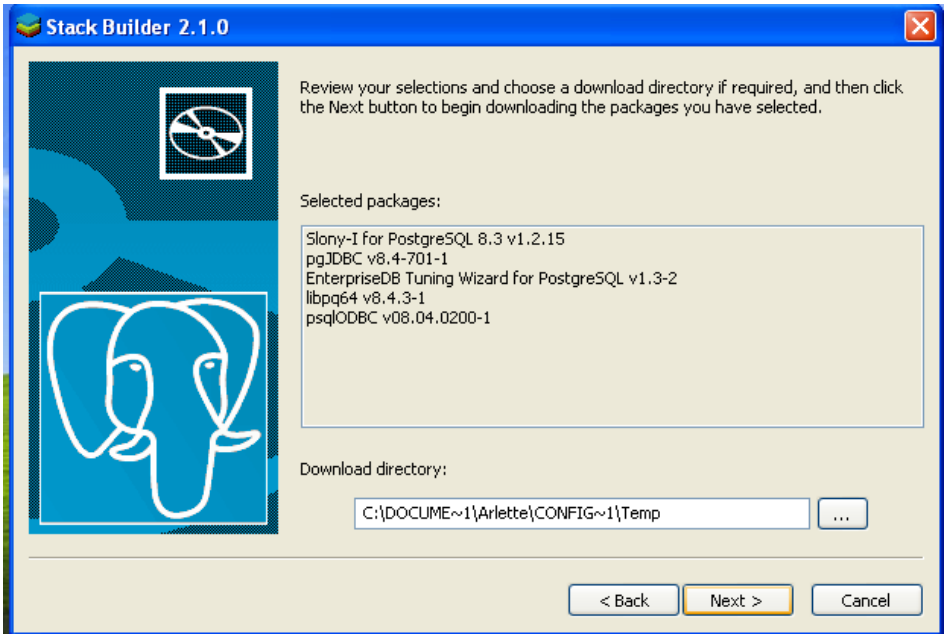
4. Para el buen funcionamiento de Slony-I, también se elije la opción de “Database Drivers” donde se seleccionará las opciones de “libq64 v8.4.3-1, pgJDBC v8.4-701-1,psqlODBC v08.04.0200-1”. Dar clic en el botón “Next”.



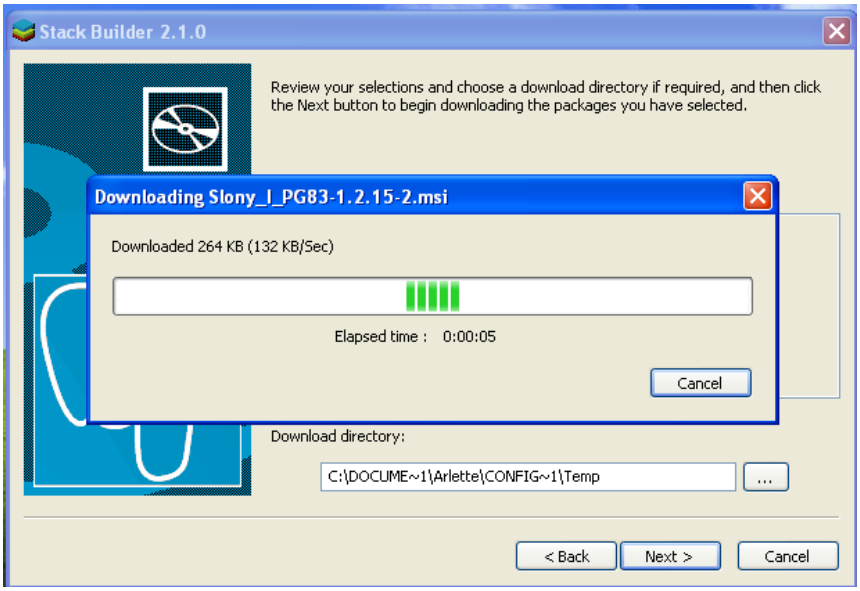
5. Se selecciona el servidor de donde se desea realizar la descarga, se recomienda dejar el que viene por default. Dar clic en el botón de “Next”.



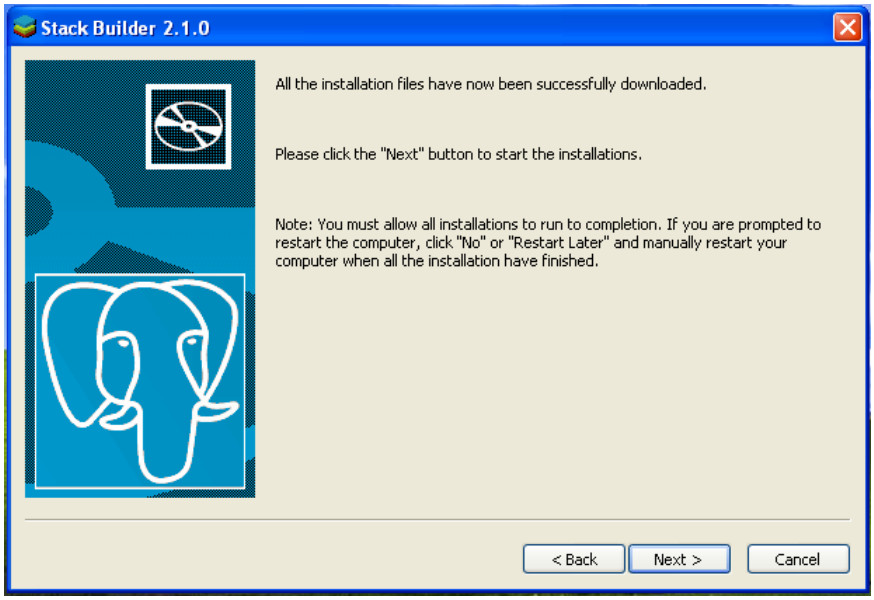
6. Se muestra una pantalla con la lista de aplicaciones que se instalarán. Dar clic en el botón “Next”



7. Inicia el proceso de descarga de los ejecutables de las aplicaciones.



8. En la siguiente pantalla se inicia el proceso de instalación. Dar clic en el botón “Next”.



9. Se finaliza la instalación de las aplicaciones.

APÉNDICES

APÉNDICE D

MANUAL DE USUARIO

Lo que se puede enseñar no vale gran cosa,
lo que vale es lo que tú tienes que aprender.
Eduardo Chillida

APÉNDICE D MANUAL DE USUARIO

MANUAL DE USUARIO

Este manual de introducción al Sistema ***SIDIPAT*** para el Hospital Regional 1°de Octubre del ISSSTE se elabora con la finalidad de proporcionar la información necesaria para poner en marcha todas las funciones que tiene dicho sistema a través de los siguientes apartados:

1. Acceso al sistema

2. Solicitudes (descripción de pantalla)
   - 2.1 Registro de una solicitud
   - 2.2 Asignación de patólogo a una solicitud (Para solicitudes de biopsia)
   - 2.3 Aclaración de una solicitud
   - 2.4 Buscar una solicitud
   - 2.5 Reporte diario de solicitudes
   - 2.6 Reporte de aclaraciones de solicitudes

3. Informes (descripción de pantalla)
   - 3.1 Selección de solicitud para generar informe
   - 3.2 Generar informe
   - 3.3 Ver informe
   - 3.4 Modificar informe
   - 3.5 Enviar Informe
   - 3.6 Acceso a notas personales para patólogos y citotecnólogos
   - 3.7 Reporte diario de informes

4. Resultados (descripción de pantalla)
   - 4.1 Obtener informes de resultados

5. Reportes (descripción de pantalla)
   - 5.1 Obtener reportes (activo únicamente para el jefe de Departamento)
   - 5.2 Cambiar el estado de un patólogo
   - 5.3 Cambiar el estado de un informe

6. Jefatura (descripción de pantalla)

7. Secretaria (descripción de pantalla)

Este manual consta de diagramas de flujo que describen los procesos a efectuar de una manera clara y entendible. Dichos diagramas utilizan cierta simbología descrita en la tabla D.1

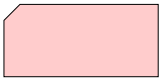
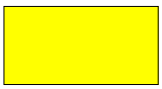
SIMBOLOGÍA	DESCRIPCIÓN
	Son notas que le ayudaran a seguir el proceso que esté realizando.
	Representan los pasos principales de cada proceso que usted tiene que realizar.
	Señalan los pasos secundarios que usted tendrá que seleccionar para continuar con el proceso.
	Indica que ha ocurrido un error durante el proceso que usted está llevando, e indica cual es el tipo de error.
	Representa que el proceso ha terminado satisfactoriamente, y puede volver a iniciarlo en el momento que usted desee.
	Esta flecha representa el camino del proceso sin que exista ningún error.
	Señala el camino cuando se está existiendo algún error, e indicara que paso realizar para regresar al camino correcto.
	Señala la unión con una nota.

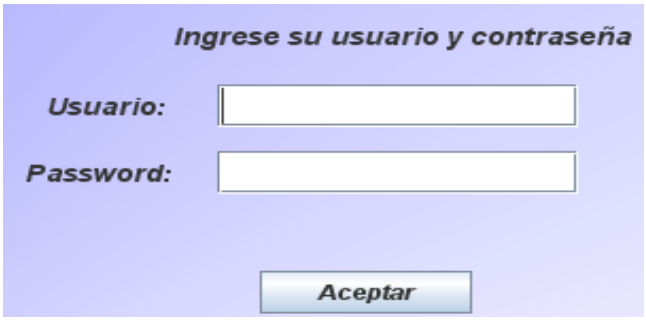
TABLA D.1 Simbología que utilizan los diagramas de flujo que componen el manual de usuario.

1. ACCESO AL SISTEMA

A continuación se describe el proceso para tener acceso a las funcionalidades del sistema ***SIDIPAT.***

La siguiente pantalla es la que se muestra cuando se corre la aplicación

Para tener acceso al menú correspondiente para cada usuario, primero deberá ingresar su usuario y paasword en los campos correspondientes



Una vez que se hayan ingresado los datos, deberá dar clic en el **botón ACEPTAR** para validar que no exista ningún error con el password, el usuario ó con la conexión a la base de datos. Si no existe ninguna contrariedad pasara a la siguiente pantalla, en caso contrario le mostrará alguno de los siguientes mensaje señalándole el problema:

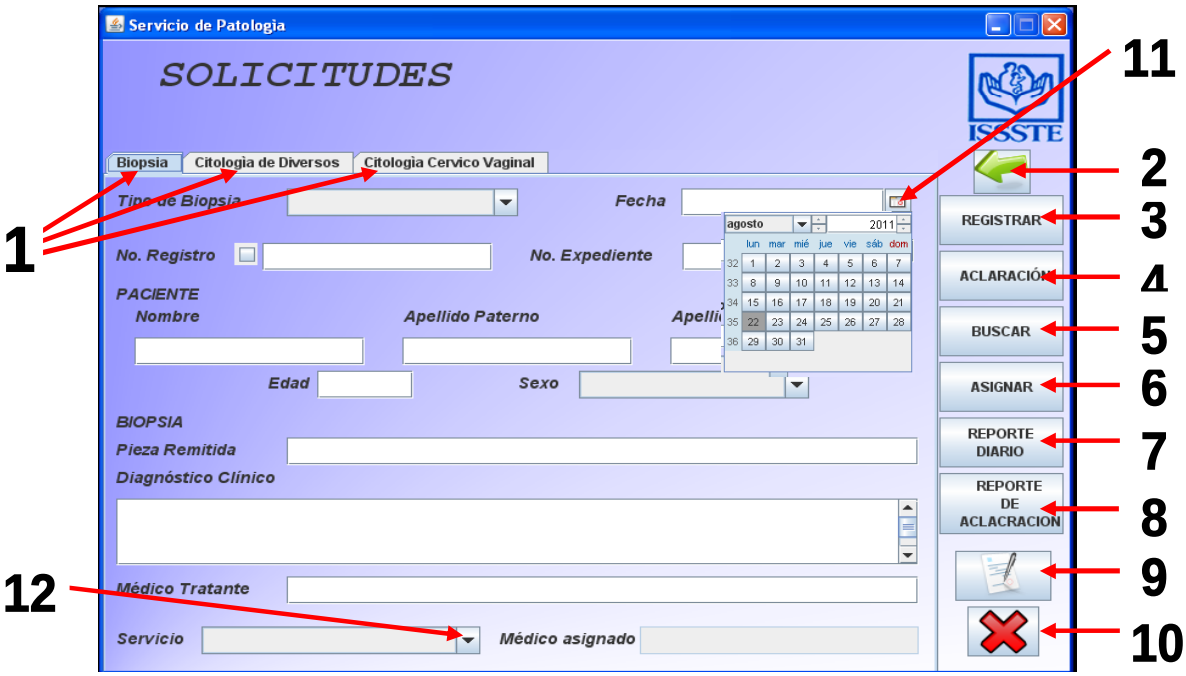
- La longitud del password es mayor o menor a la permitida por las reglas de seguridad del sistema. (Recordar que el password tiene una longitud de 8 caracteres). Para solucionar el problema verifique su password y vuélvalo a ingresar.

- Cuando el usuario no coinciden con los existentes en el sistema y la base de datos. Para solucionar este problema verifique su usuario así como su password y vuélvalos a ingresar, en caso de que persista el problema acudir con el administrador del sistema.

- El problema se ocasiona al no haber comunicación o no poder establecer conexión con la base de datos, puede ser ocasionado por ingresar incorrectamente el password o por que exista un error en algún nodo que contiene la base de datos. Para solucionar este error ingrese nuevamente sus datos o consulte a su administrador del sistema.

2. SOLICITUDES

En este apartado se mencionan los componentes de la **pantalla solicitud**, a continuación se describe cada uno de ellos:



Descripción

1. **PESTAÑA:** Son 3 pestañas que describen los tipos de solicitudes que existen, cada una de ellas contiene campos específicos que se requieren para registrar cada solicitud.
   - Solicitud de Biopsia.
   - Solicitud de Citología de Diversos
   - Solicitud de Citología Cervico Vaginal

2. **BOTÓN REGRESAR (fecha hacia la izquierda de color verde):** Este botón solo está activo en la sesión de Jefatura y Secretaria, tiene la finalidad de regresar al usuario a un menú principal.

3. **BOTÓN REGISTRAR:** Tiene la función de realizar la inserción a la base de datos cuando los datos requeridos en la solicitud estén completados correctamente.

4. **BOTÓN ACLARACIÓN:** Realiza una marca en la solicitud de la cual se pidió alguna aclaración o una nueva revisión de la muestra. Esta marca se guarda en la base de datos.

5. **BOTÓN BUSCAR:** Este botón nos lleva a una nueva pantalla donde se pueden realizar ciertas combinaciones para realizar búsquedas de las solicitudes ingresadas. El resultado de la búsqueda se mostrará en una tabla.

6. **BOTÓN ASIGNAR:** Efectuará la distribución equitativa de las solicitudes de tipo biopsia entre los diferentes patólogos. En la base de datos registrará el nombre del médico al cual le fue asignada dicha solicitud, con esto se llevará un control de que solicitudes serán estudiadas por cada médico.

7. **BOTÓN REPORTE DIARIO:** Genera un archivo que contiene los datos más relevantes de las solicitudes registradas durante el día.

8. **BOTÓN REPORTE DE ACLARACIONES:** Obtendrá mediante una consulta a la base de datos todas las solicitudes que tienen el status de aclaración, para generar un archivo de resumen.

9. **BOTÓN LIMPIA (icono de una hoja con un lápiz):** Después de finalizar el registro de una solicitud, limpiara todos los campos para poder ingresar la siguiente solicitud.

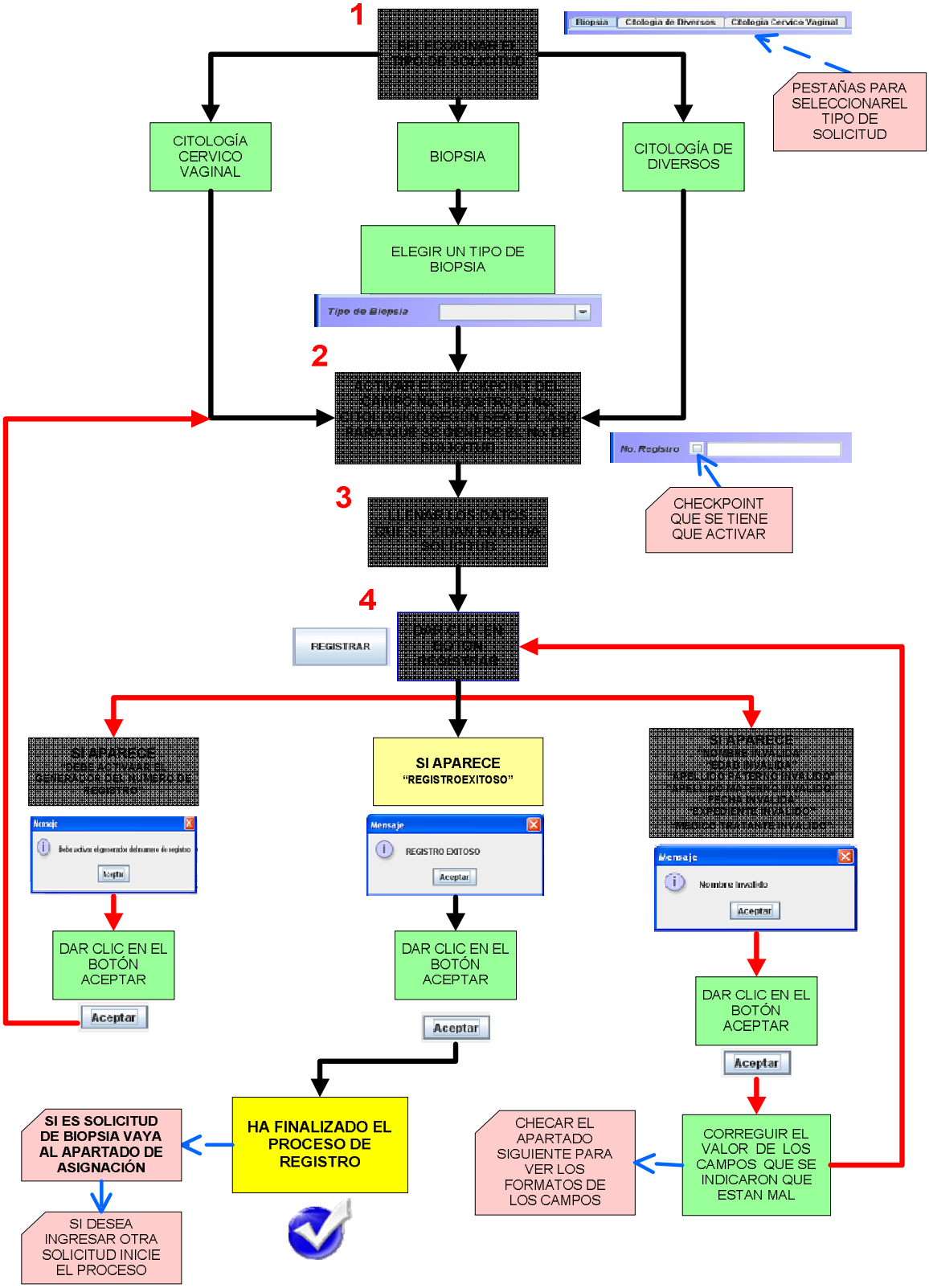
10. **BOTÓN CERRAR SESIÓN (X):** Finaliza la sesión del usuario.

11. **BOTÓN CALENDARIO:** Muestra un calendario dinámico donde se puede seleccionar la fecha deseada.

12. **BOTÓN SELECCIÓN:** Este botón se encuentra en todos los campos que contemplen una lista de opciones para elegir alguna en específico.

2.1 REGISTRO DE UNA SOLICITUD

Para registrar una solicitud con éxito se deben realizan los 5 pasos siguientes:



FORMATOS PARA CAMPOS DE LA PANTALLA SOLICITUD

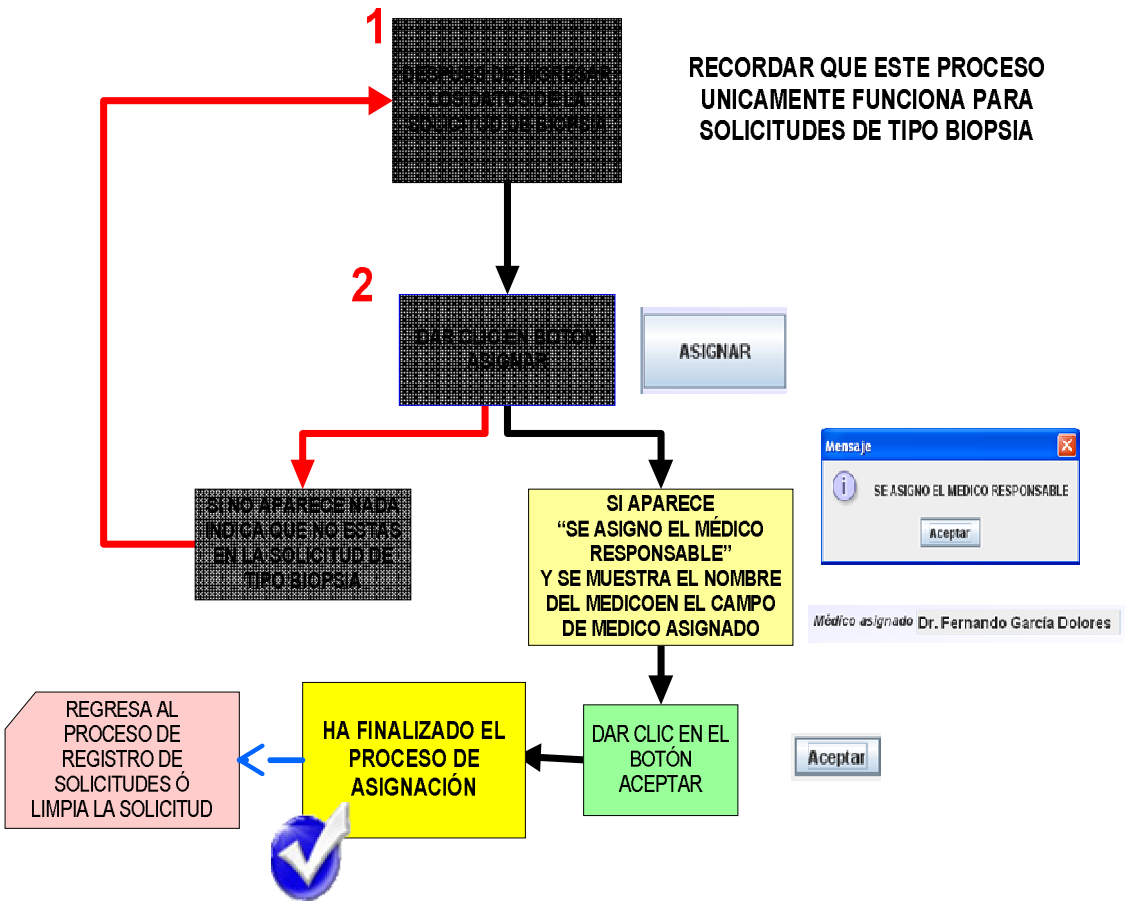
En la tabla D.2 se muestran los formatos que deben cumplir ciertos campos de la **pantalla solicitud** para evitar algún error en el proceso de registro de solicitud.

CAMPO	FORMATO
Edad	Acepta un número en el rango de 1 – 99
Nombre	Acepta cadenas que contemplan caracteres de la A-Z incluyendo la Ñ, donde la primera letra es mayúscula y las siguientes son minúsculas. Ejemplos de cadenas que acepta: **Andrea** **Andrea Ma.** **Andrea Ma. Rosa** **Andrea Rosa** **Nuñez**
Ap. Paterno	
Ap. Materno	
Fecha	Valida que la fecha no sea mayor a la fecha actual.
Expediente	Valida una cadena que contempla cuatro letras(L), siete números(N) y un guión. El formato de esta cadena es el siguiente: **LLLLNNNNNN-N** **LLLL NNNNNN-N** Espacio
Médico Tratante	Valida una cadena que puede contener letras desde la A-Z así como la Ñ, el formato que debe seguir se muestra en los siguientes ejemplos: **Dr. Nuñez** **Dra. Monica Garcia** **Dr. Raul G. Rodriguez** **Dra. Celeste Ma. Fuentes**

TABLA D.2 Formatos de los campos de la **pantalla solicitud.**

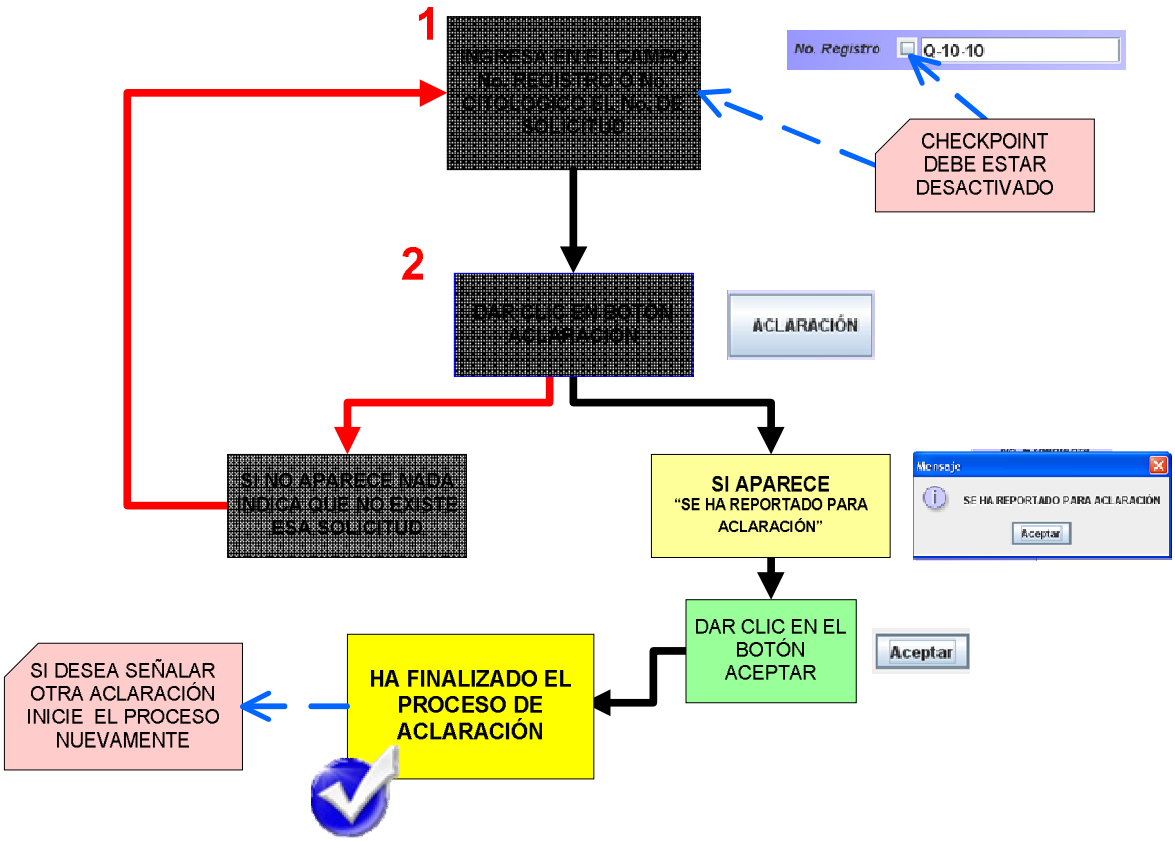
2.2 ASIGNACIÓN DE PATÓLOGO A UNA SOLICITUD

Recordar que este apartado únicamente es válido cuando se trabaja con una solicitud de biopsia. Para asignarle una solicitud a un patólogo se ejecutan los pasos que a continuación se muestran:



2.3 ACLARACIÓN DE UNA SOLICITUD

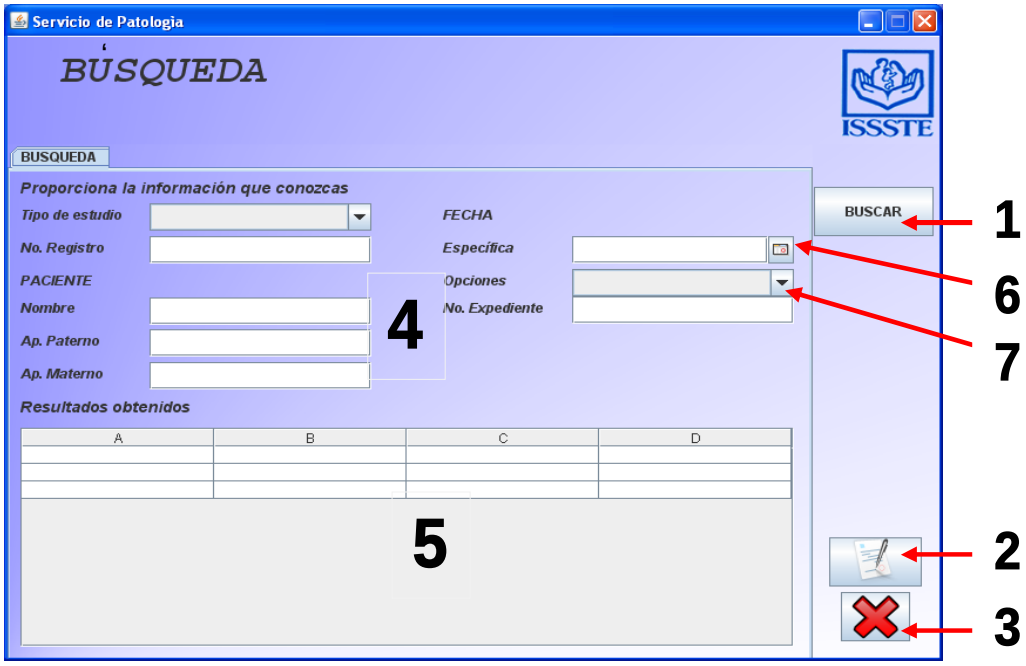
Para señalar que una solicitud requiere ser nuevamente revisada por alguna aclaración que pida el paciente ó el médico tratante se deben seguir los siguientes pasos:



2.4 BUSCAR UNA SOLICITUD

Para buscar una solicitud de cualquier tipo se debe realizar el siguiente procedimiento:

A continuación se describe la **pantalla de búsqueda**, al igual que su funcionamiento:



Descripción

1. **BOTÓN DE BÚSQUEDA:** Realiza la búsqueda en la base de datos de acuerdo a la información proporcionada.

2. **BOTÓN LIMPIA (icono de una hoja con un lápiz):** Después de finalizar la búsqueda limpia todos los campos.

3. **BOTÓN CERRAR SESIÓN (X):** Finaliza la función de búsqueda y regresa a la **pantalla solicitud**.

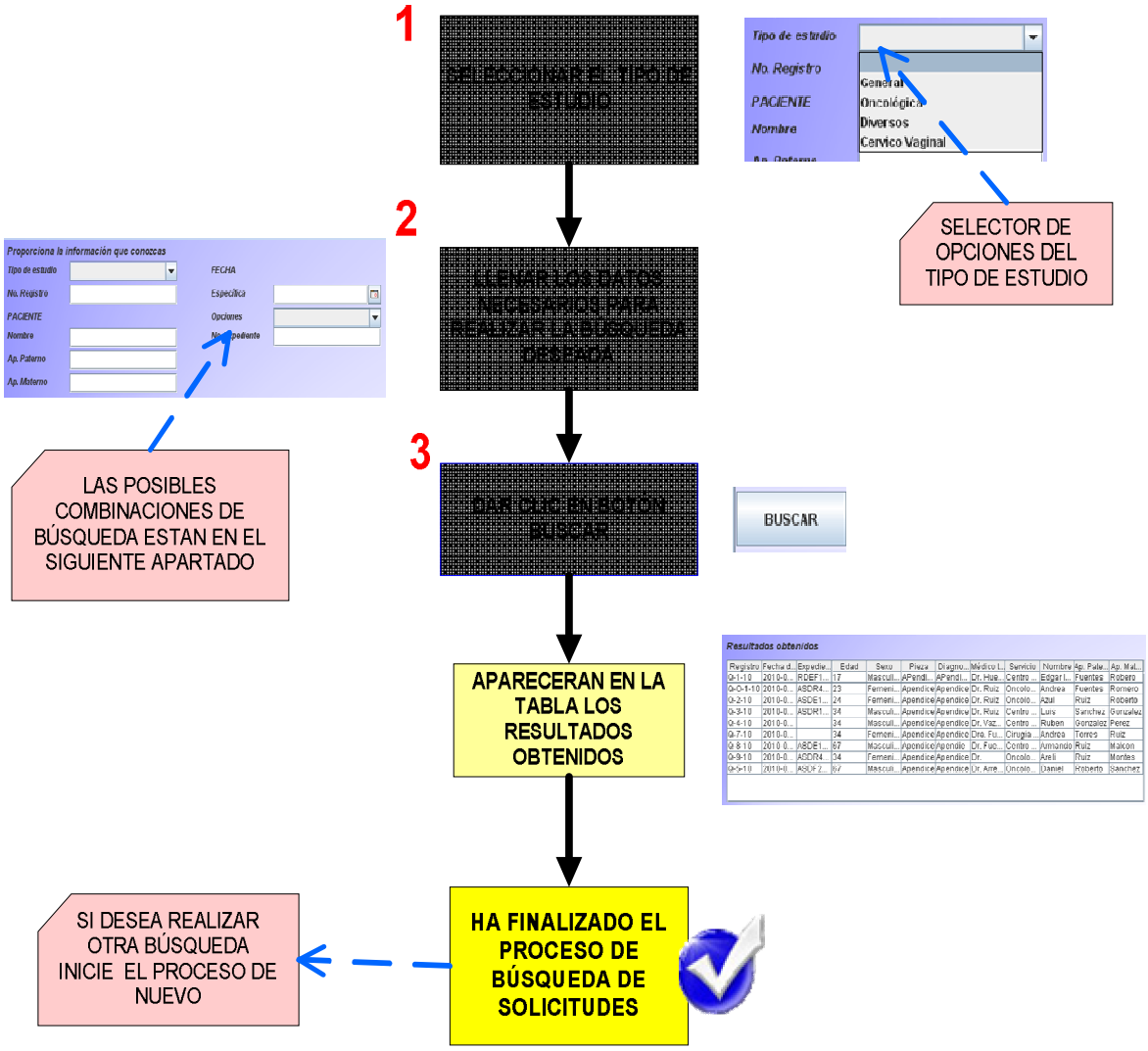
4. **SECCIÓN DE CAMPOS:** Son los campos que se llenan para realizar la búsqueda correspondiente.

5. **SECCIÓN DE RESULTADOS:** Tabla donde se muestran los resultados de la búsqueda.

6. **BOTÓN CALENDARIO:** Muestra un calendario dinámico donde se puede seleccionar una fecha específica.

7. **BOTÓN SELECCIÓN:** Este botón se encuentra en todos los campos que contemplen una lista de opciones para elegir alguna en específico.

Ya en la pantalla de búsqueda se deben realizar los siguientes pasos para obtener el resultado de una búsqueda específica:



COMBINACIONES POSIBLES DE BUSQUEDA

El sistema **SIDIPAT** soporta las combinaciones que se encuentran en la tabla D.3 para realizar una búsqueda específica de una solicitud.

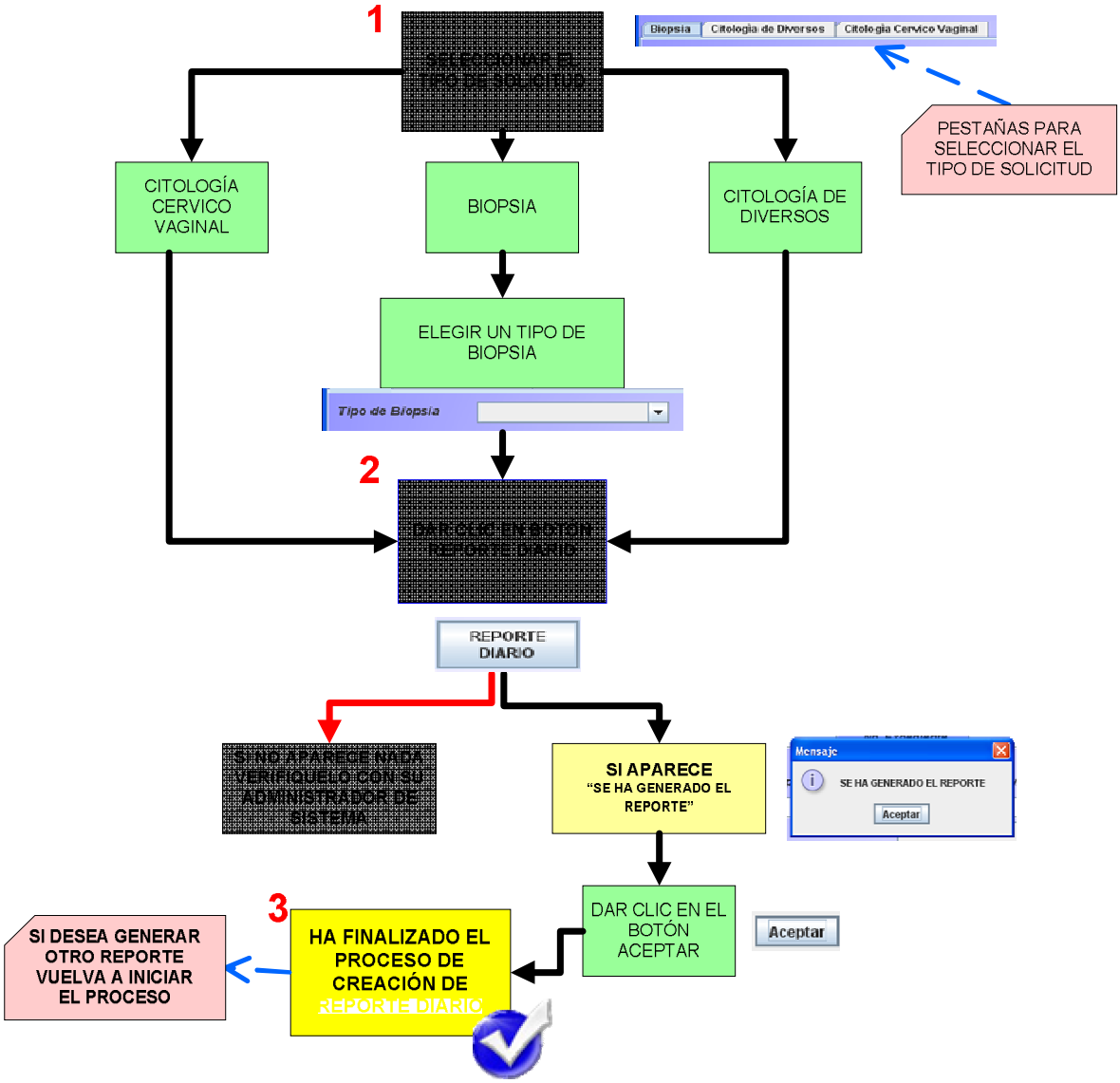
TIPO DE ESTUDIO (PARA TODOS)

COMBINACIONES
• Nombre y apellido materno
• Nombre y apellido paterno
• Nombre, apellido materno y paterno
• Fecha especifica
• Opciones de fecha
  ◦ 3 días atrás
  ◦ Mes actual
  ◦ 3 meses atrás
  ◦ Año en curso
  ◦ 1 año atrás
• Registro
  ◦ Junto con las combinaciones de nombre, apellido materno y paterno.
• Expediente
  ◦ Junto con las combinaciones de nombre, apellido materno y paterno.

TABLA D.3 Combinaciones posibles para buscar una solicitud.

2.5 REPORTE DIARIO DE SOLICITUDES

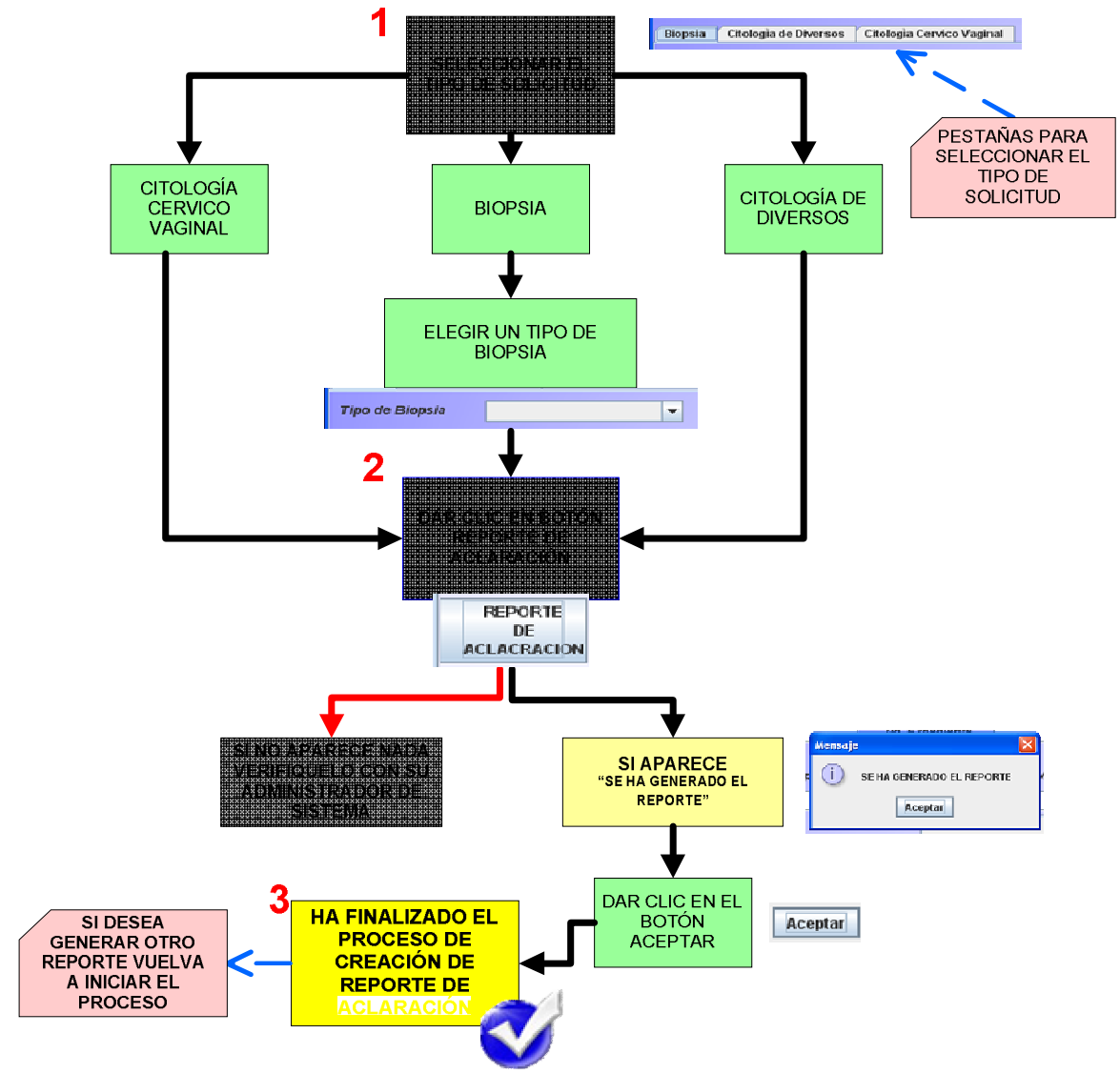
Los pasos descritos a continuación se ejecutan para obtener un reporte diario de las solicitudes que fueron registradas.



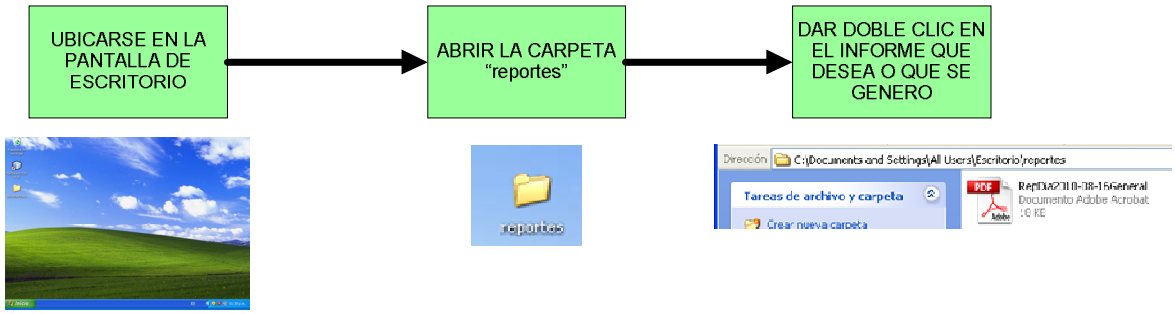
PARA VER EL CONTENIDO DEL REPORTE GENERADO REALIZAMOS LO SIGUIENTE

2.6 REPORTE DE ACLARACIONES DE SOLICITUDES

Los reportes de las solicitudes que están señaladas para alguna aclaración se obtienen al realizar el siguiente procedimiento:



PARA VER EL CONTENIDO DEL REPORTE GENERADO REALIZAMOS LO SIGUIENTE

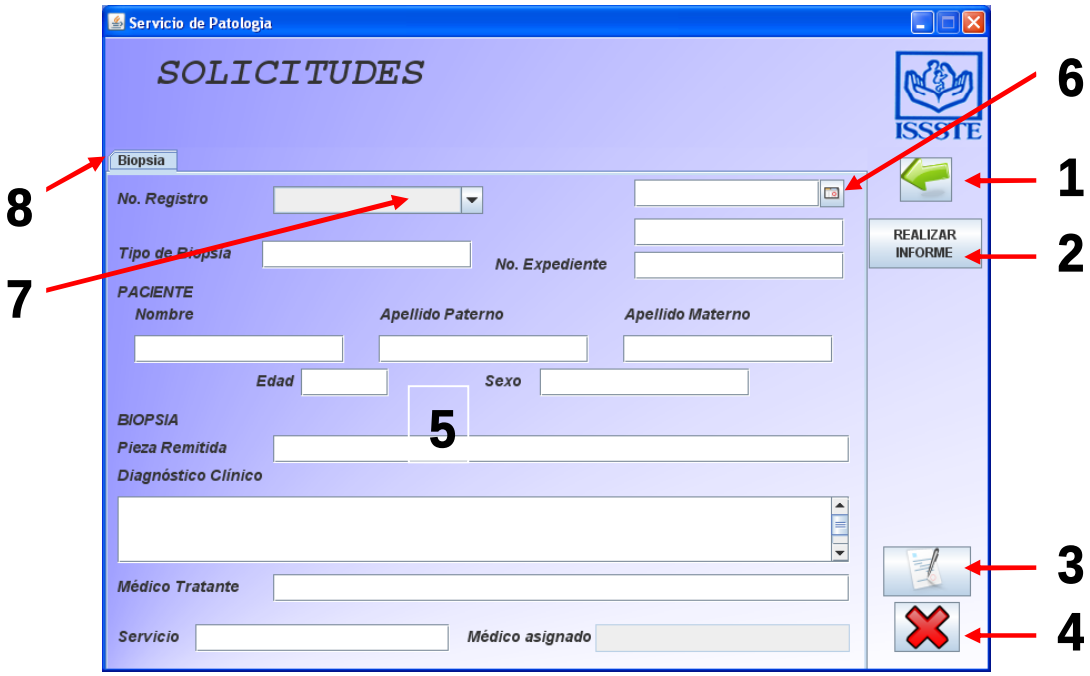


3. INFORMES

Para generar los informes de cada solicitud se manejaran dos pantallas:

- **PANTALLA DE SELECCIÓN DE SOLICITUD**

La **pantalla selección de solicitud** está compuesta por:

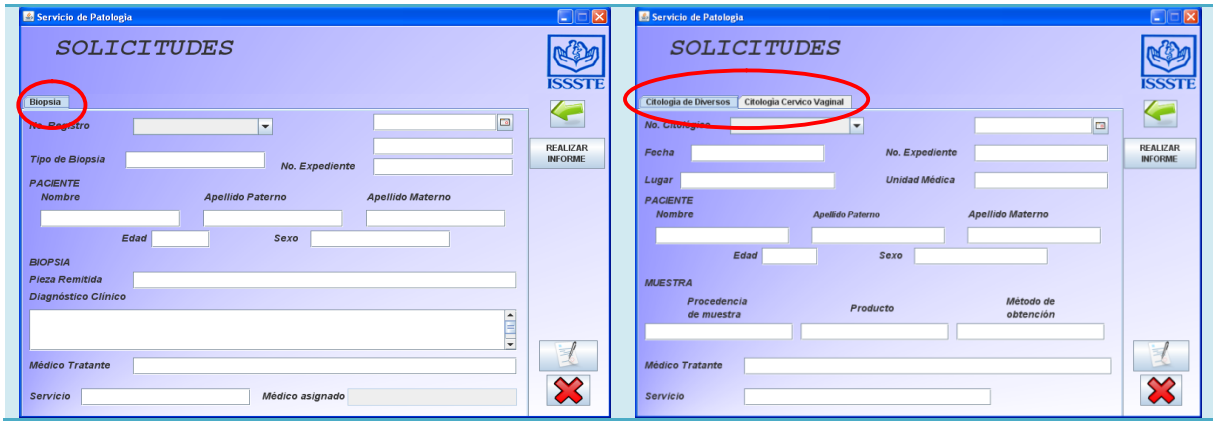


Descripción

1. **BOTÓN REGRESA (fecha hacia la izquierda de color verde):** Este botón solo está activo en la sesión de Jefatura, tiene la finalidad de regresar al usuario a un menú principal.
2. **BOTÓN REALIZAR INFORME:** Lleva a la **pantalla de informe**.
3. **BOTÓN LIMPIA (icono de una hoja con un lápiz):** Después de finalizar la visualización de una solicitud, limpia todos los campos para mostrar otra solicitud.
4. **BOTÓN CERRAR SESIÓN (X):** Finaliza la sesión del usuario.
5. **SECCIÓN DE CAMPOS:** Son los campos que se llenan automáticamente para visualizar la información correspondiente de cada solicitud.
6. **BOTÓN CALENDARIO:** Muestra un calendario dinámico donde se puede seleccionar la fecha específica para localizar las solicitudes.
7. **BOTÓN SELECCIÓN:** Este botón mostrará una lista de solicitudes, las cuales no tienen un informe relacionado, al igual que dependerán de la fecha seleccionada.
8. **PESTAÑA:** Permite seleccionar el tipo de solicitud a la cual se desea realizar un informe.

La **pantalla de selección de solicitud** varía en la pestaña y en la sección de campos de acuerdo al tipo de informes que se desee realizar.

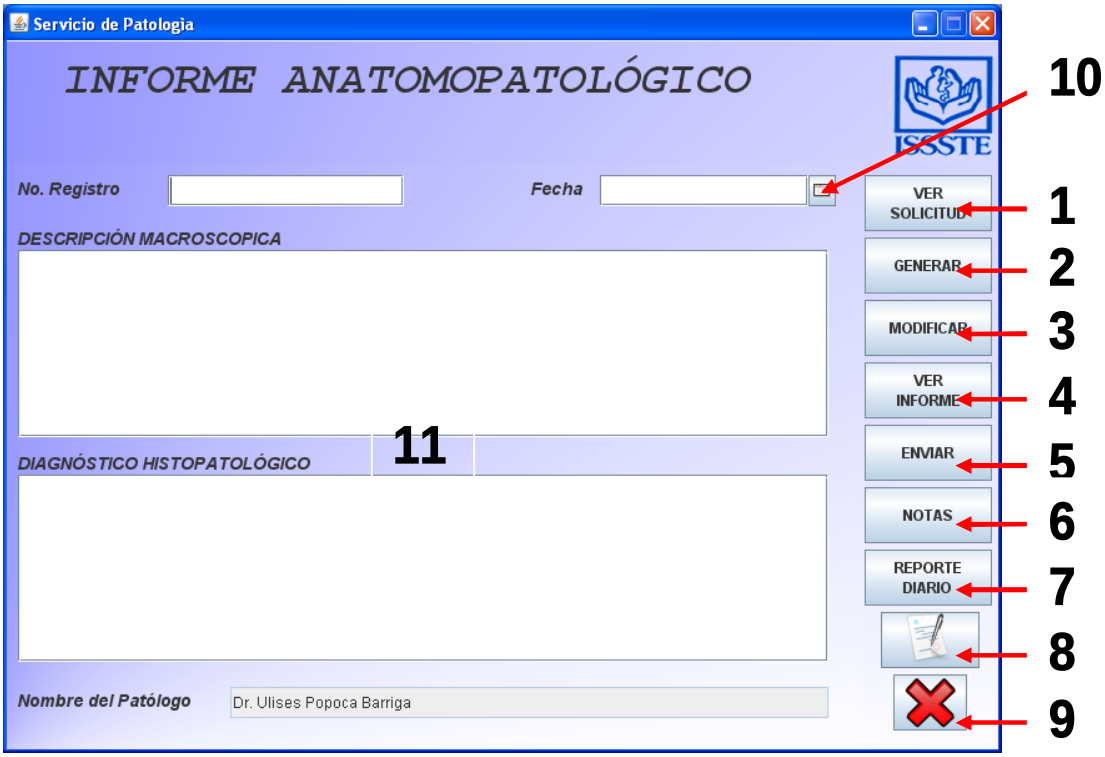
ANATOMOPATOLÓGICO ***CITOLÓGICO***



La funcionalidad de los botones es la misma en ambos tipos de informe.

- **PANTALLA DE INFORME**

Esta pantalla sirve para generar informes de resultados, está compuesta por los siguientes elementos:

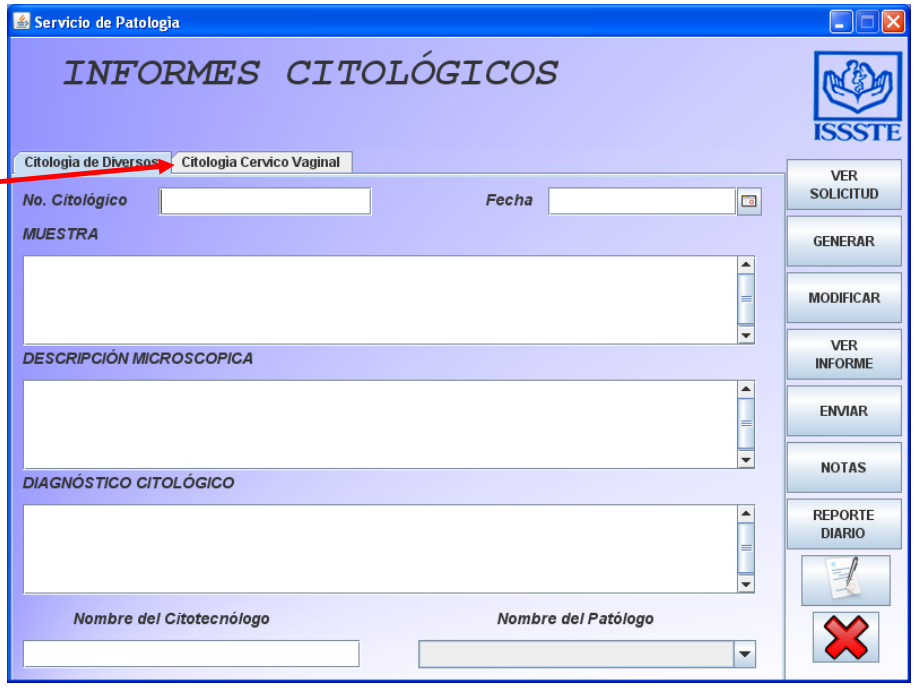


Descripción

1. **BOTÓN VER SOLICITUD:** Tiene la función de regresar a la **pantalla selección de solicitud**.
2. **BOTÓN GENERAR:** Este botón guarda en la base de datos el registro del informe creado para dicha solicitud.
3. **BOTÓN MODIFICAR:** Permite guardar los cambios realizados al informe.
4. **BOTÓN VER INFORME:** Muestra en los campos correspondientes los valores que dicho informe tiene en la base de datos.
5. **BOTÓN ENVIAR:** Cambia el estado del informe (de pendiente a finalizado).
6. **BOTÓN NOTAS:** Abre la **pantalla notas** donde se realizan anotaciones personales.
7. **BOTÓN REPORTE DIARIO:** Crea un archivo, que contiene los datos más relevantes de los informes generados durante el día.
8. **BOTÓN LIMPIA (icono de una hoja con un lápiz):** Limpia todos los campos de la **pantalla de informe**.
9. **BOTÓN CERRAR SESIÓN (X):** Finaliza la sesión del usuario.
10. **BOTÓN CALENDARIO:** Muestra un calendario dinámico donde se puede seleccionar una fecha específica.
11. **SECCIÓN DE CAMPOS:** Área donde se encuentran los campos donde se ingresa la información del informe.

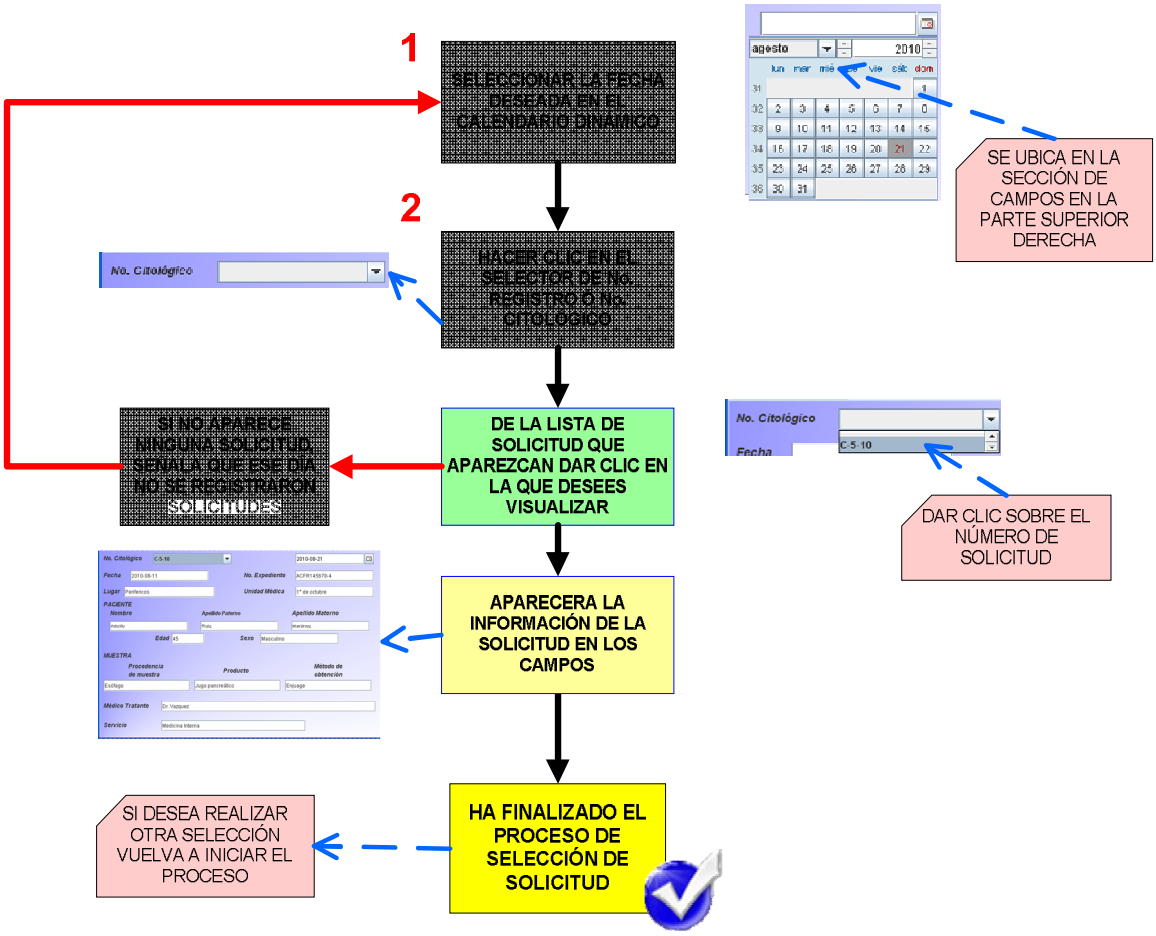
En el caso de informes citológicos se agrega un elemento más a la **pantalla de informe**, los demás componentes son iguales en diseño y funcionalidad a los del otro tipo de informe.

PESTAÑA: Seleccionar el tipo de informe citológico que se desea realizar.



3.1 SELECCIÓN DE SOLICITUD PARA GENERAR INFORME

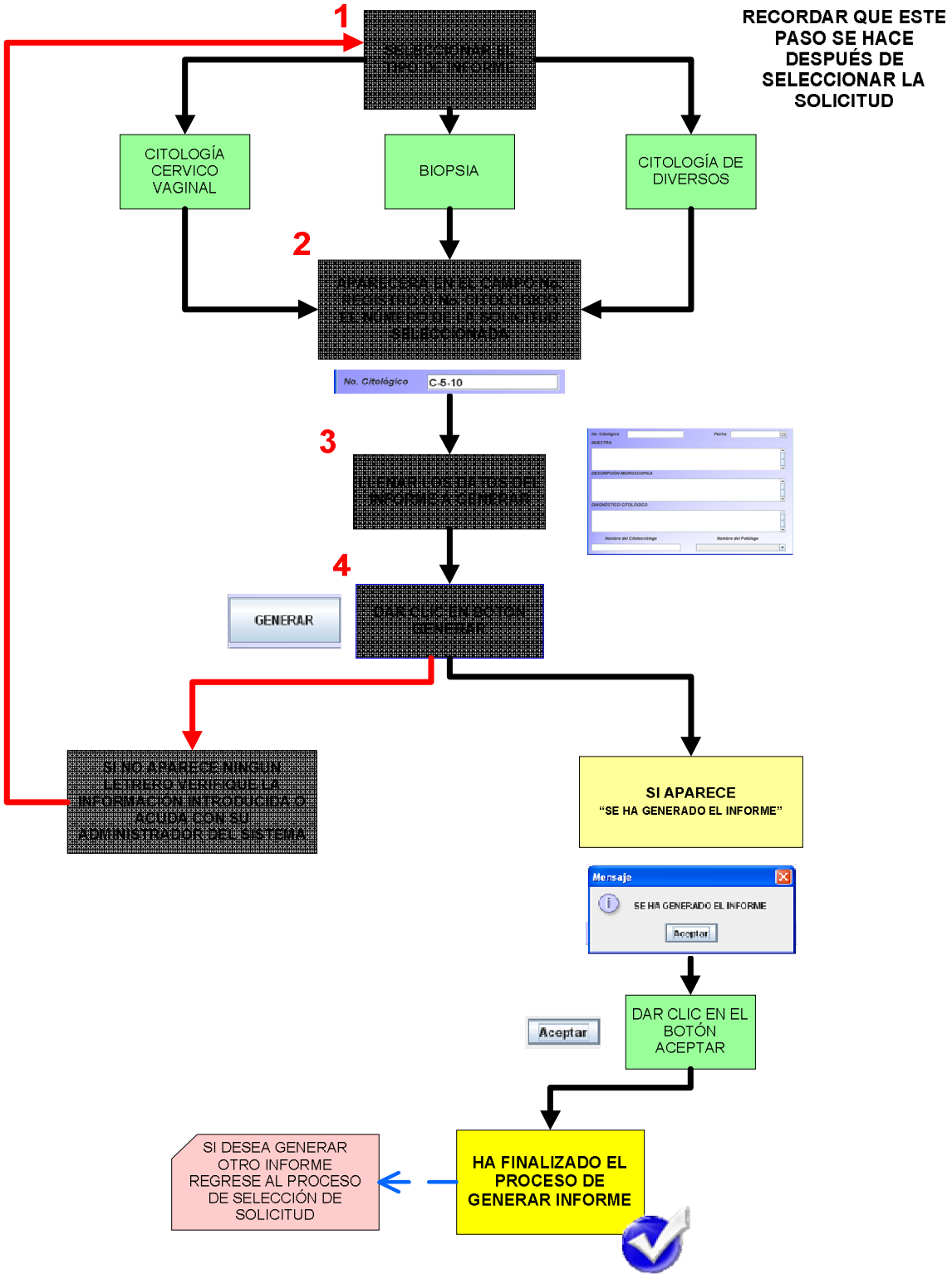
Para poder realizar el informe de resultado de alguna solicitud primero se debe seleccionar dicha solicitud a través de los pasos siguientes:



SI DESEA IR A LA PANTALLA DONDE PODRA GENERAR EL INFORME DE LA SOLICITUD SELECCIONADA SIGA LOS SIGUENTES PASOS

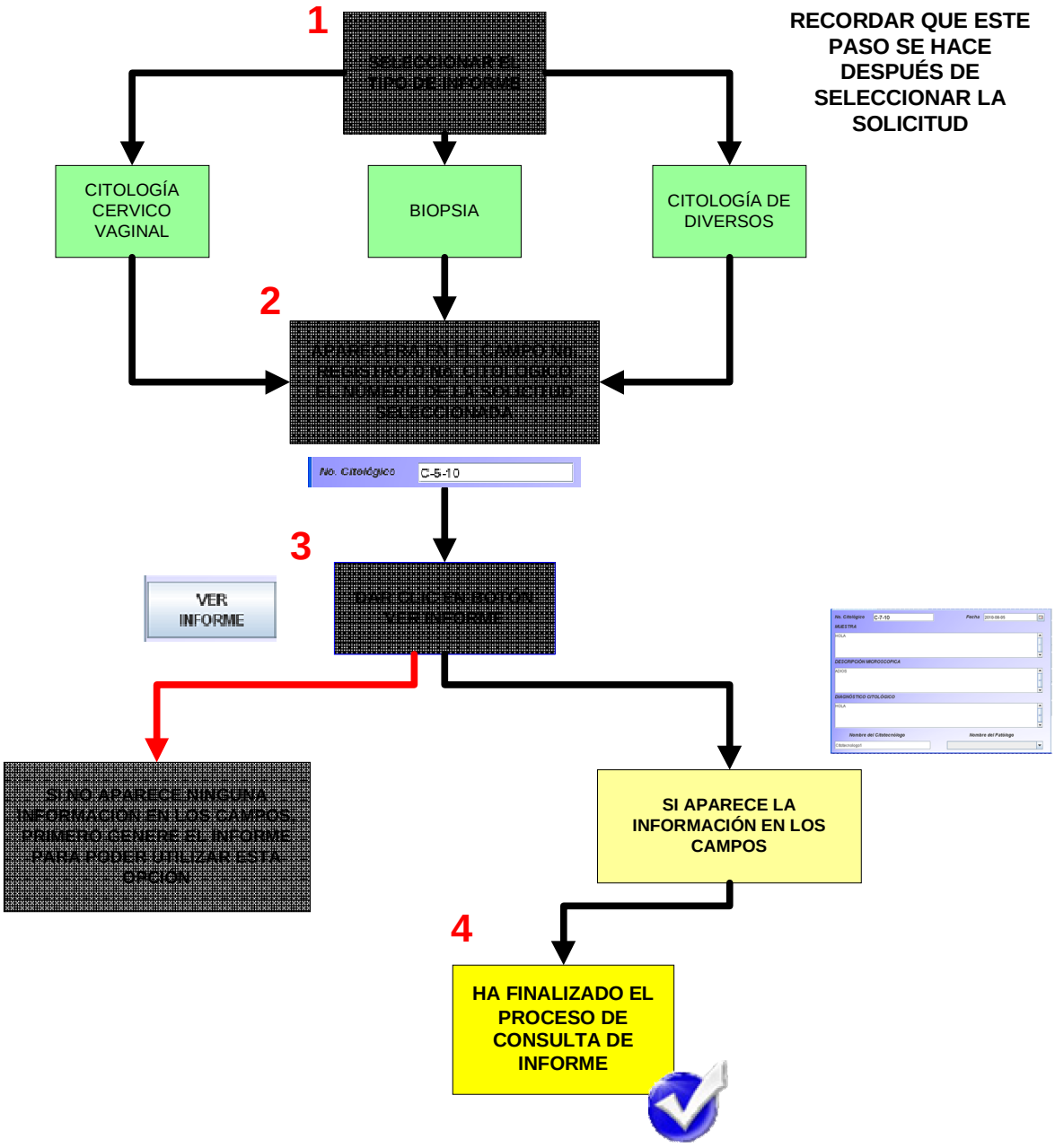
3.2 GENERAR INFORME

Para generar un informe de resultado de una solicitud que fue seleccionada en la **pantalla de selección de solicitud** se ejecuta el procedimiento que a continuación se presenta:



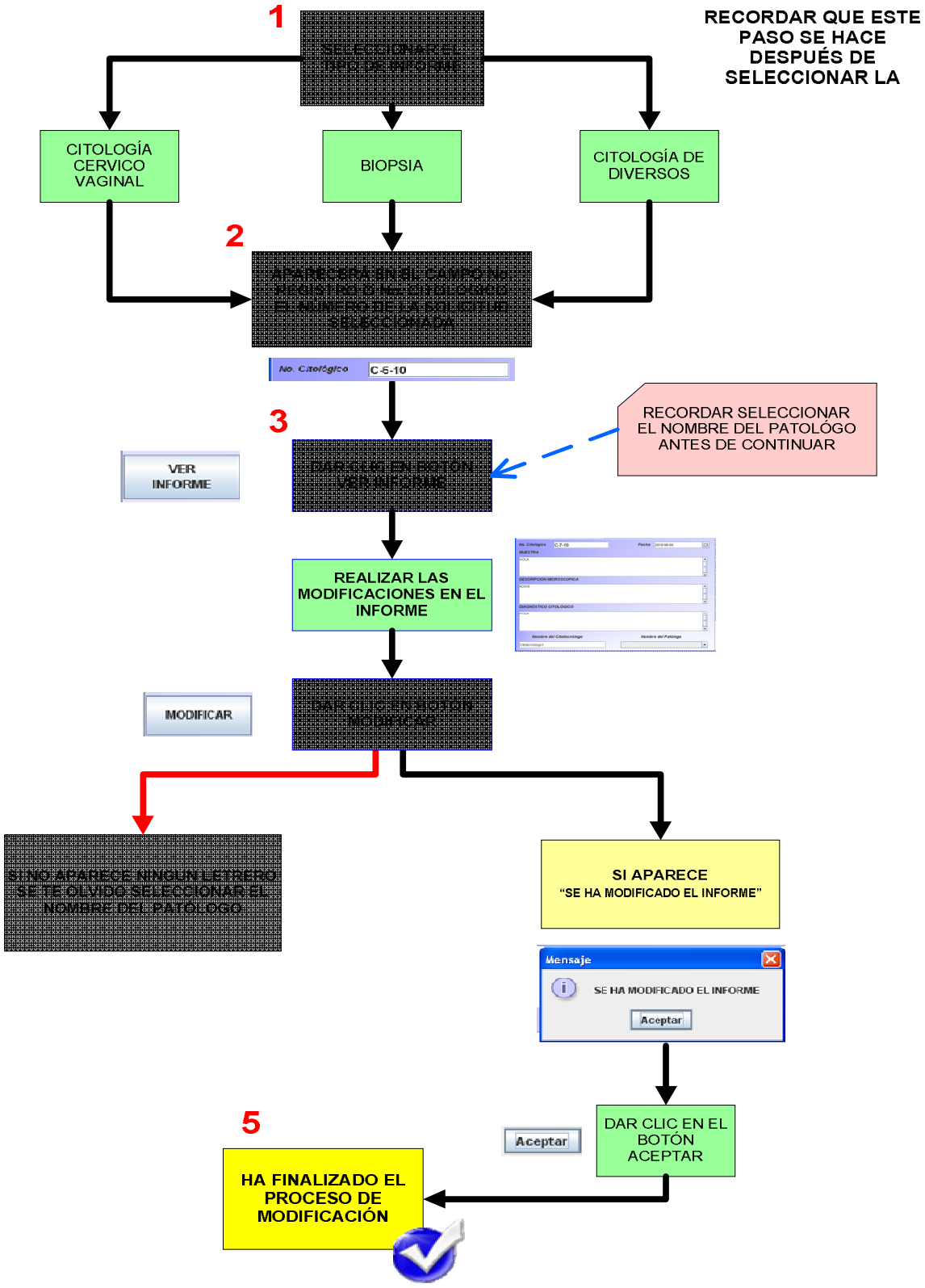
3.3 VER INFORME

Los cuatro pasos que se muestran en el siguiente diagrama permiten visualizar el contenido de un informe de resultado específico.



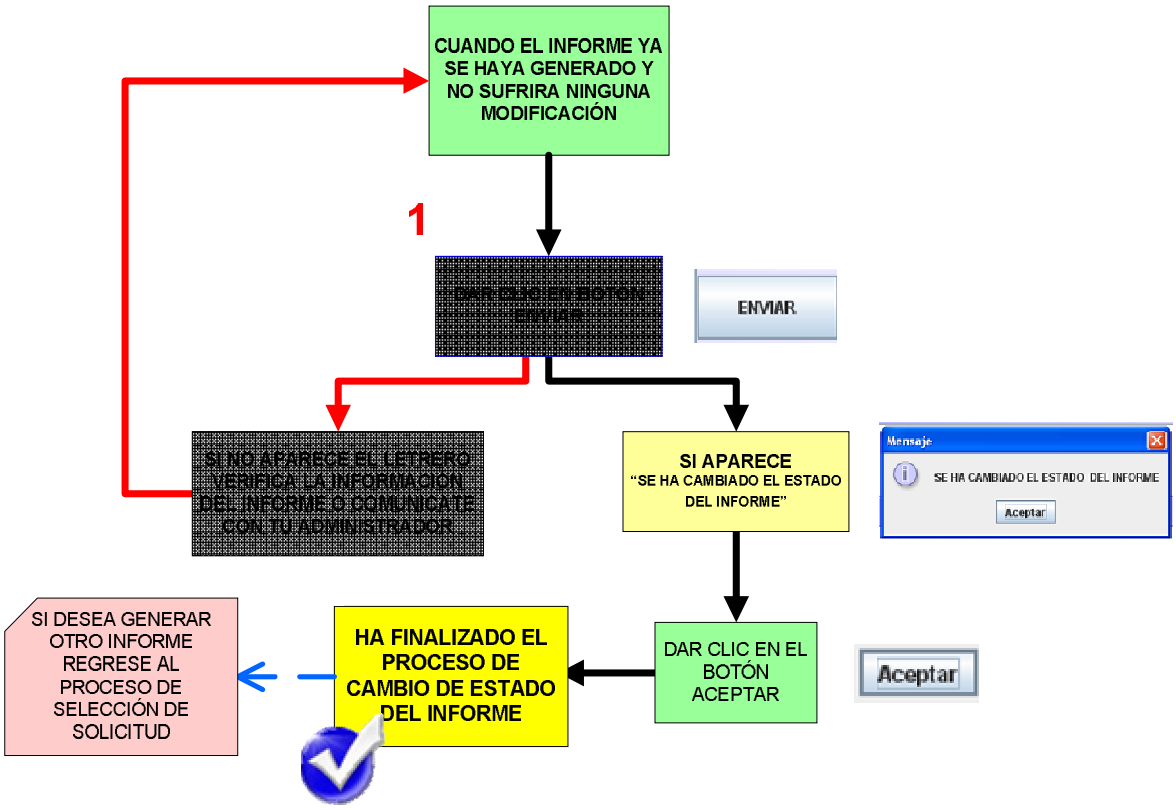
3.4 MODIFICAR INFORME

En el caso de que se realice algún cambio en el contenido del informe de resultado se siguen los pasos que se mencionan a continuación:



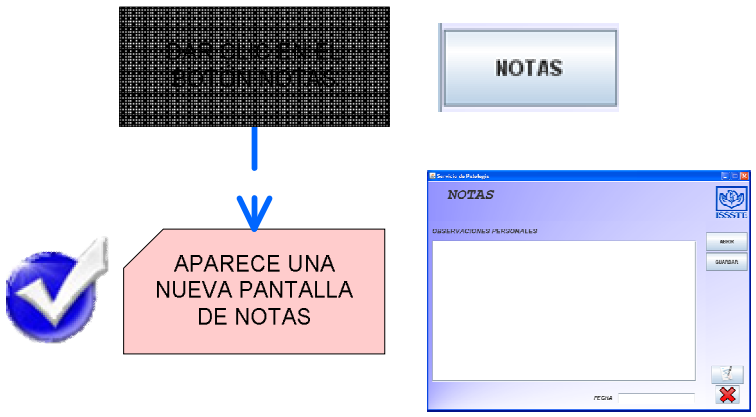
3.5 ENVIAR INFORME

Para cambiar el status de un informe de resultado de pendiente a finalizado se deben ejecutar los siguientes dos pasos:

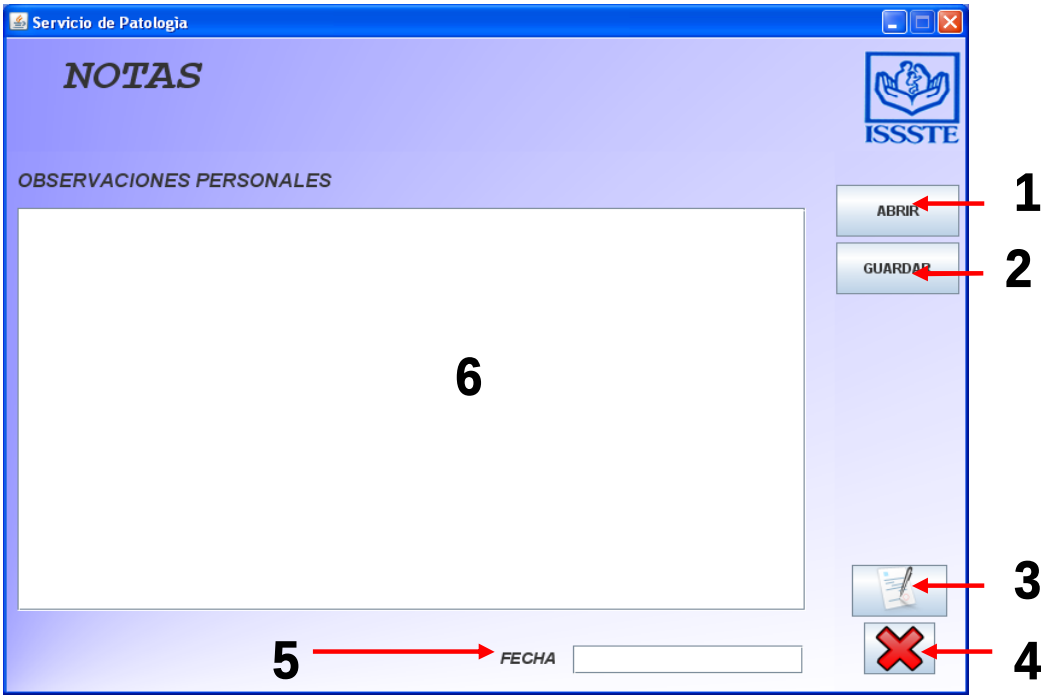


3.6 ACCESO A NOTAS PERSONALES PARA PATÓLOGOS Y CITOTECNÓLOGOS

Los patólogos y citotecnólogos en ocasiones realizan ciertas anotaciones acerca de los diagnósticos de las biopsias o muestras, por lo que se introduce un área específica para realizar esta acción siguiendo los pasos que se mencionan en el diagrama:



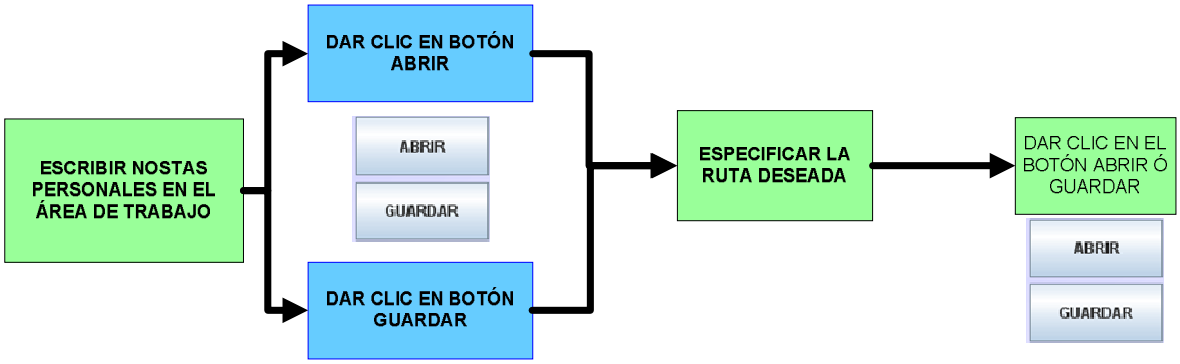
A continuación se explican los componentes de la **pantalla notas**, al igual que su funcionalidad



Descripción

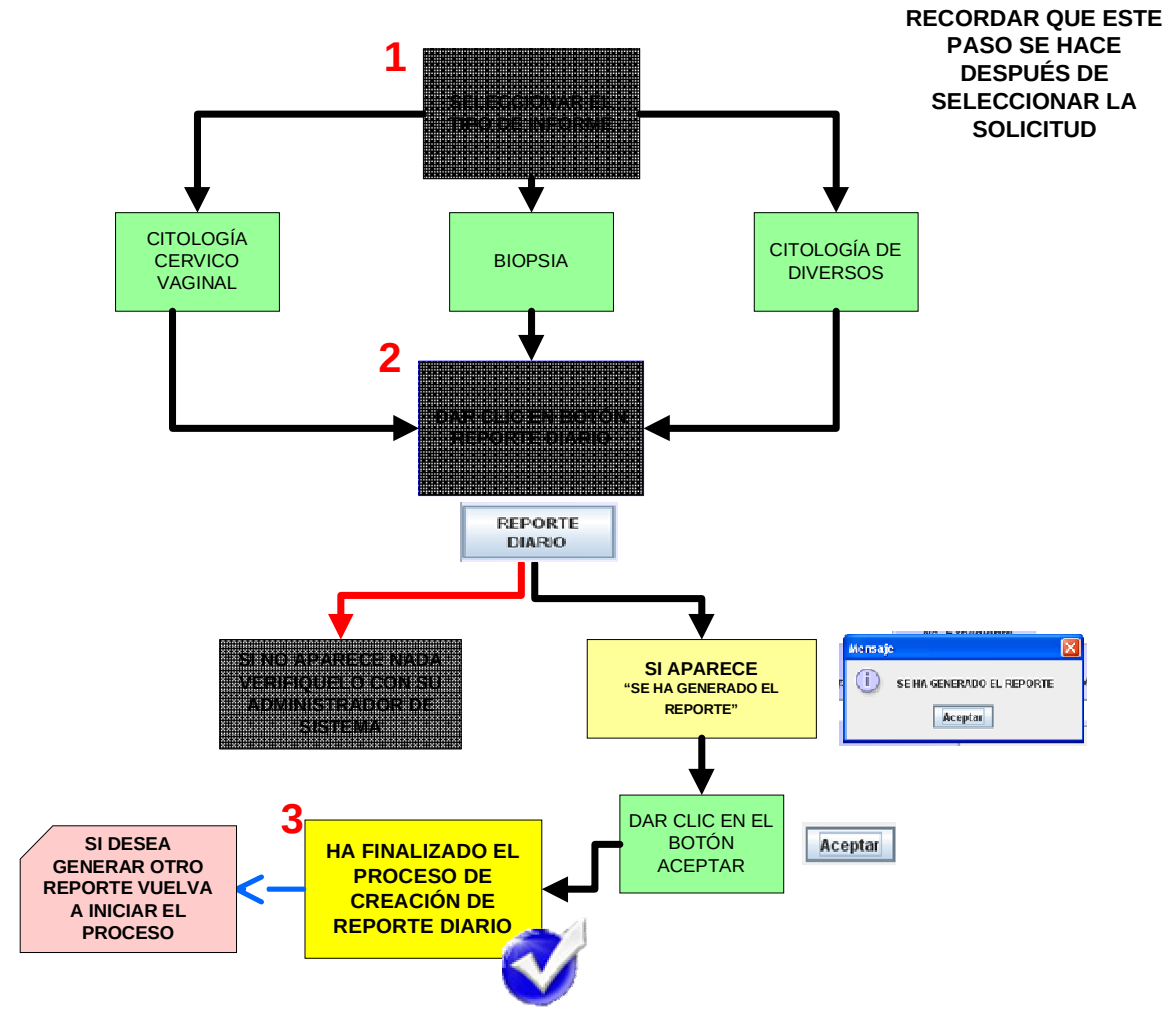
1. **BOTÓN ABRIR:** Muestra un menú donde se puede elegir un archivo de notas que se mostrará en el área de trabajo.
2. **BOTÓN GUARDAR:** Muestra un menú donde se elije la ruta para guardar la información existente en el área de trabajo.
3. **BOTÓN LIMPIA (icono de una hoja con un lápiz):** Limpia el área de trabajo.
4. **BOTÓN CERRAR SESIÓN (X):** Regresa a la pantalla de informe.
5. **FECHA:** Escribe la fecha en la que se realiza la nota teniendo el formato AAAA-MM-DD.
6. **AREA DE TRABAJO:** Zona de escritura.

El diagrama que a continuación se presenta describe el procedimiento para abrir o guardar algún archivo de notas:



3.7 REPORTE DIARIO DE INFORMES

El siguiente procedimiento se ejecuta para crear un reporte diario de los informes de resultados que fueron generados sin importar el status en el que se encuentren:



PARA VER EL CONTENIDO DEL REPORTE GENERADO REALIZAMOS LO SIGUIENTE

4. RESULTADOS

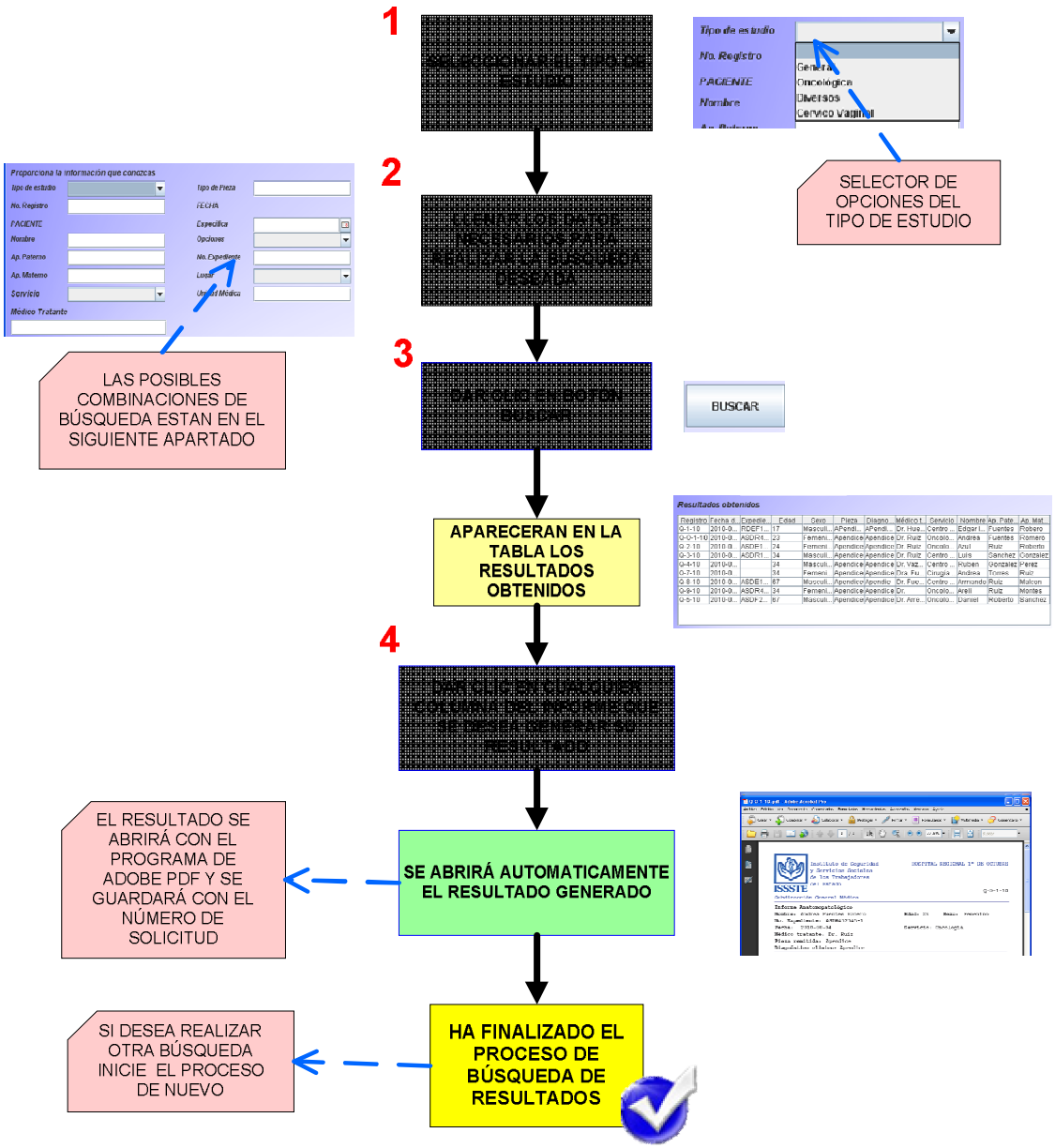
Para obtener los informes de resultados que tengan status finalizado se utiliza la **pantalla resultados** que contempla los siguientes elementos:

Descripción:

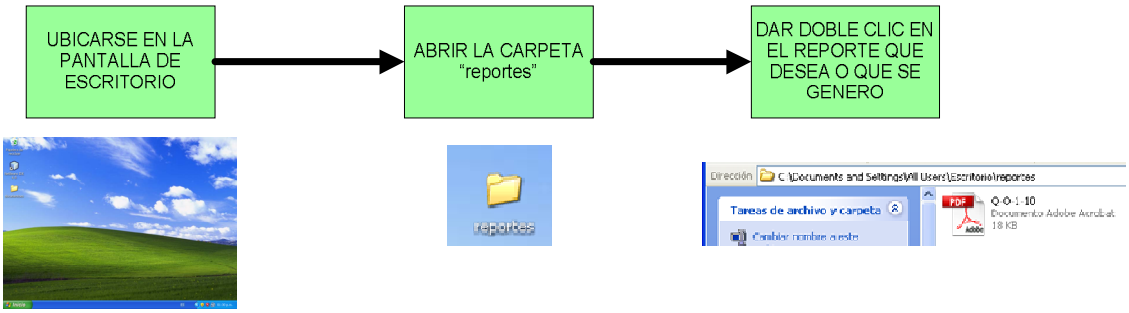
1. **BOTÓN DE BÚSQUEDA:** Realiza la búsqueda en la base de datos de acuerdo a la información proporcionada.
2. **BOTÓN REGRESAR (fecha hacia la izquierda de color verde):** Este botón solo está activo en la sesión de Jefatura y Secretaria, tiene la finalidad de regresar al usuario a un menú principal.
3. **BOTÓN LIMPIA (icono de una hoja con un lápiz):** Después de finalizar la búsqueda limpia todos los campos.
4. **BOTÓN CERRAR SESIÓN (X):** Finaliza sesión del usuario.
5. **SECCIÓN DE CAMPOS:** Son los campos que se llenaran para poder realizar la búsqueda correspondiente.
6. **SECCIÓN DE RESULTADOS:** Tabla donde se muestran los resultados de la búsqueda.

4.1 OBTENER EL RESULTADO FINAL DE UNA SOLICITUD

El procedimiento siguiente tiene la finalidad de ayudar a obtener los informes de resultados finalizados:



PARA VER POSTERIORMENTE LOS RESULTADOS SIGA LOS SIGUIENTES PASOS:



COMBINACIONES DE BUSQUEDAS POSIBLES PARA RESULTADOS

Las combinaciones posibles para realizar la búsqueda de informes de resultados se encuentran en la tabla D.4. Se ocupa la siguiente simbología para saber en qué tipo de informes de resultados se puede utilizar dicha combinación:

SIMBOLOGIA **G** = GENERAL **O** = ONCOLOGICA **D** = DIVERSOS **C** = CERVICO VAGINAL

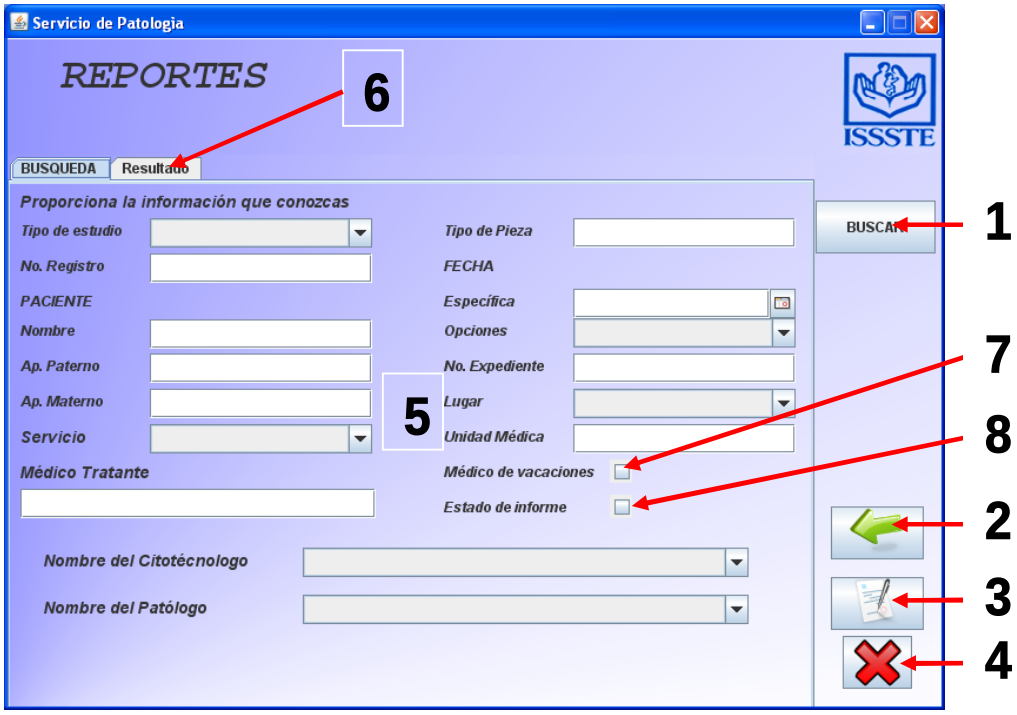
	TIPO DE ESTUDIO

COMBINACIONES	▪ Nombre y apellido materno (GODC)
	▪ Nombre y apellido paterno (GODC)
	▪ Nombre, apellido materno y paterno (GODC)
	▪ Servicio (GOD)
	▪ Pieza (GO)
	▪ Lugar (D)
	▪ Unidad (C)
	▪ Médico Tratante (GODC)
	▪ Fecha específica (GODC)
	▪ Opciones de fecha (GODC)
	  ▪ 3 días atrás
	  ▪ Mes actual
	  ▪ 3 meses atrás
	  ▪ Año en curso
	  ▪ 1 año atrás
	▪ Registro (GODC)
	  ▪ Junto con las combinaciones de nombre, apellido materno y paterno.
	▪ Expediente (GODC)
	  ▪ Junto con las combinaciones de nombre, apellido materno y paterno.

TABLA D.4 Combinaciones para la búsqueda de informes de resultados.

5. REPORTES

Para obtener diferentes reportes se utiliza la **pantalla reportes,** la cual es utilizada únicamente por **JEFATURA**. Los componentes de dicha pantalla son los siguientes:

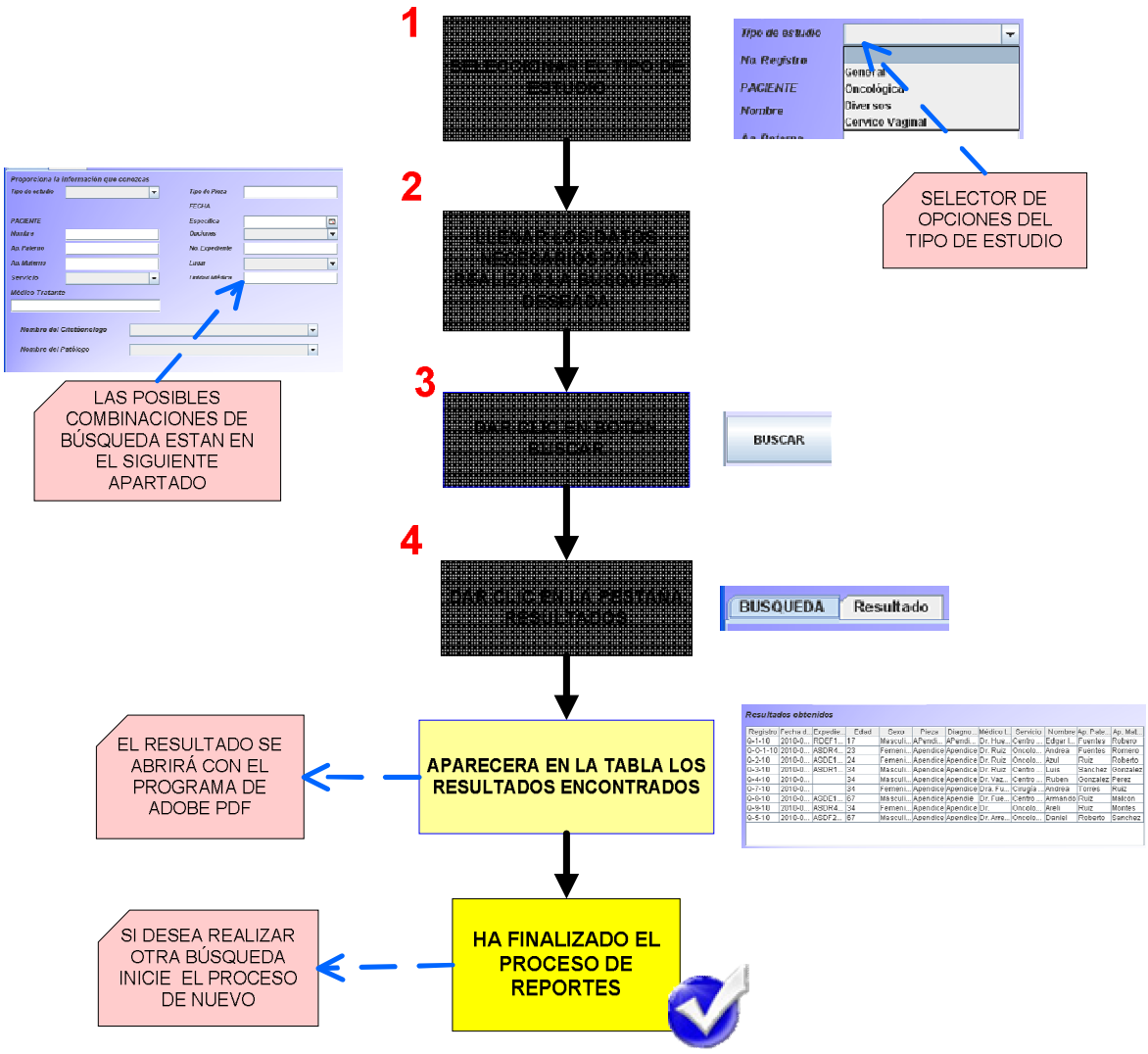


Descripción:

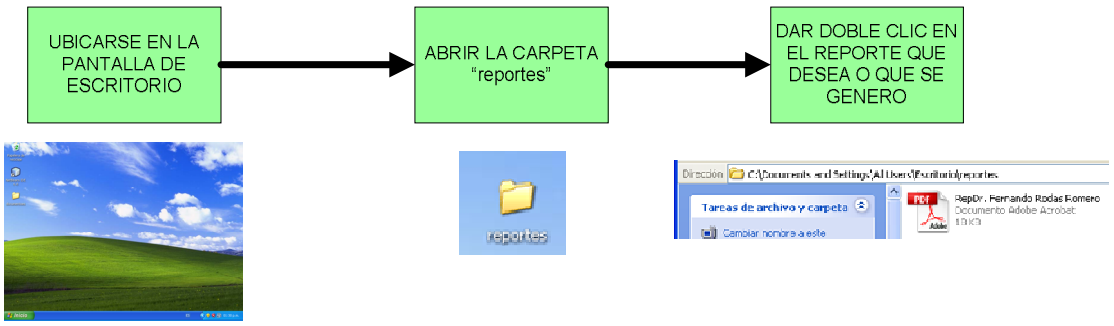
1. **BOTÓN DE BÚSQUEDA:** Realiza la búsqueda en la base de datos de acuerdo a la información proporcionada para generar reportes.
2. **BOTÓN REGRESAR (fecha hacia la izquierda de color verde):** Este botón solo está activo en la sesión de Jefatura, tiene la finalidad de regresar al usuario a un menú principal.
3. **BOTÓN LIMPIA (icono de una hoja con un lápiz):** Después de finalizar la búsqueda limpia todos los campos.
4. **BOTÓN CERRAR SESIÓN (X):** Finaliza sesión del usuario.
5. **SECCIÓN DE CAMPOS:** Son los campos que se llenaran para poder realizar la búsqueda correspondiente.
6. **PESTAÑA DE RESULTADOS:** Pestaña que contiene una tabla donde se muestran los resultados de la búsqueda.
7. **BOTÓN VACACIONES:** Tiene la funcionalidad de poder cambiar el estado del patólogo (activo o de vacaciones).
8. **BOTÓN INFORME:** Permite cambiar el estado del informe de finalizado a pendiente para poder realizar alguna modificación.

5.1 REALIZAR UN REPORTE

Los pasos siguientes señalan el procedimiento para generar los diferentes reportes:



PARA ABRIR LOS REPORTES GENERADOS REALIZA LOS SIGUIENTES PASOS



COMBINACIONES DE BUSQUEDAS POSIBLES PARA REPORTES

Las combinaciones posibles para obtener un reporte se muestran en la tabla D.5. Se ocupa la siguiente simbología para saber en qué tipo de informes de resultados ó solicitudes se puede utilizar dicha combinación:

SIMBOLOGIA **G** = GENERAL **O** = ONCOLOGICA **D** = DIVERSOS **C** = CERVICO VAGINAL

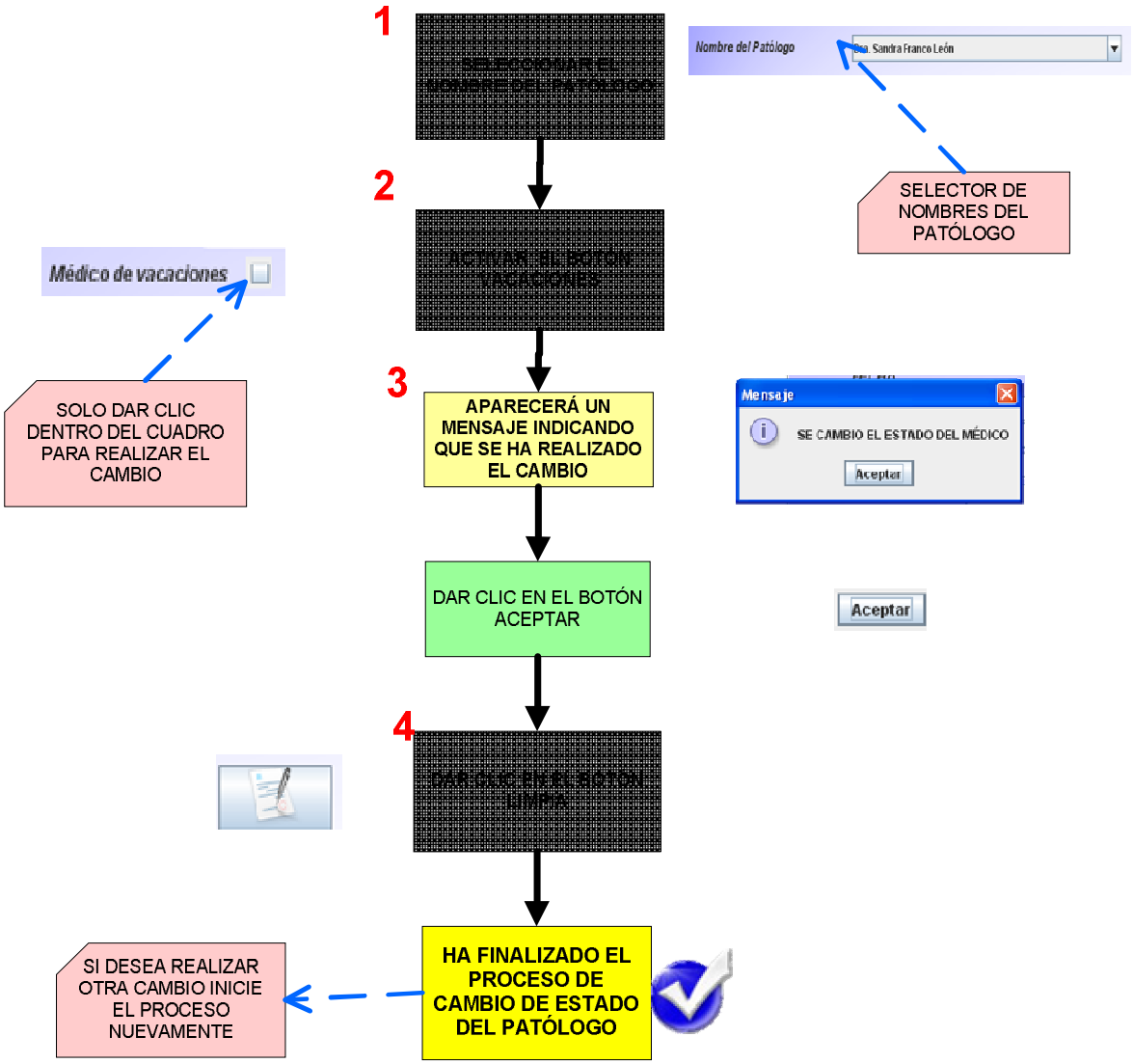
	TIPO DE ESTUDIO

COMBINACIONES	▪ Nombre y apellido materno (GODC)
	▪ Nombre y apellido paterno (GODC)
	▪ Nombre, apellido materno y paterno (GODC)
	▪ Servicio (GOD)
	▪ Pieza (GO)
	▪ Lugar (D)
	▪ Unidad (C)
	▪ Médico Tratante (GODC)
	▪ Fecha específica (GODC)
	▪ Opciones de fecha (GODC)
	  ▪ 3 días atrás
	  ▪ Mes actual
	  ▪ 3 meses atrás
	  ▪ Año en curso
	  ▪ 1 año atrás
	▪ Expediente (GODC)
	  ▪ Junto con las combinaciones de nombre, apellido materno y paterno.
	▪ Patólogo (No se tiene que seleccionar ningún tipo de estudio)
	  ▪ Fecha especifica
	  ▪ Opciones de fecha
	▪ Citotecnólogo (No se tiene que seleccionar ningún tipo de estudio)
	  ▪ Fecha especifica
	  ▪ Opciones de fecha

TABLA D.5 Combinaciones posibles para generar reportes.

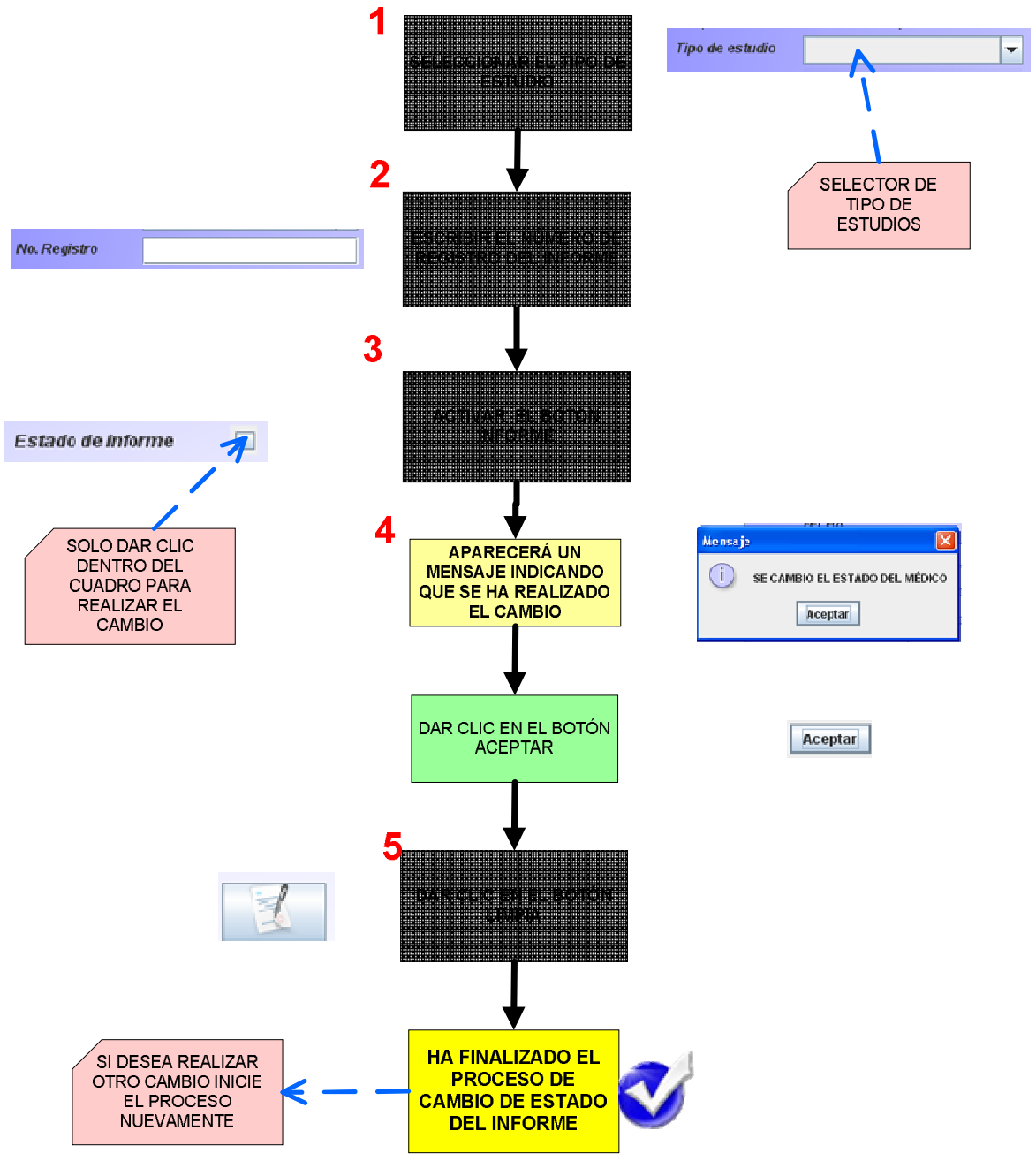
5.2 CAMBIAR EL ESTADO DE UN PATÓLOGO

Hay periodos de vacaciones que tiene un patólogo para ello se requiere cambiar el estado de activo a vacaciones, los siguientes pasos ayudan a realizar dicha función:



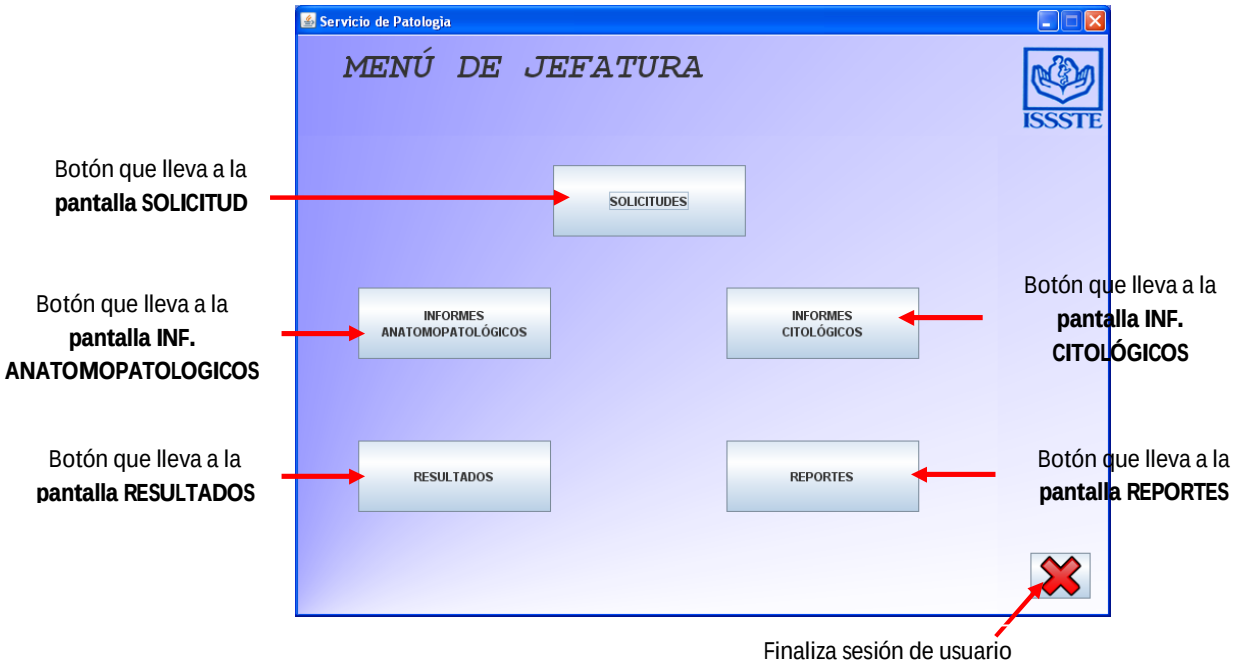
5.3 CAMBIAR EL ESTADO DE UN INFORME

Después de finalizar un informe de resultado ya no se pueden realizar modificaciones, para tener la posibilidad de corregir el jefe del Departamento debe cambiar el estado del informe de resultado de finalizado a pendiente, los pasos que a continuación se presentan ayudarán a cambiar dicho estado:



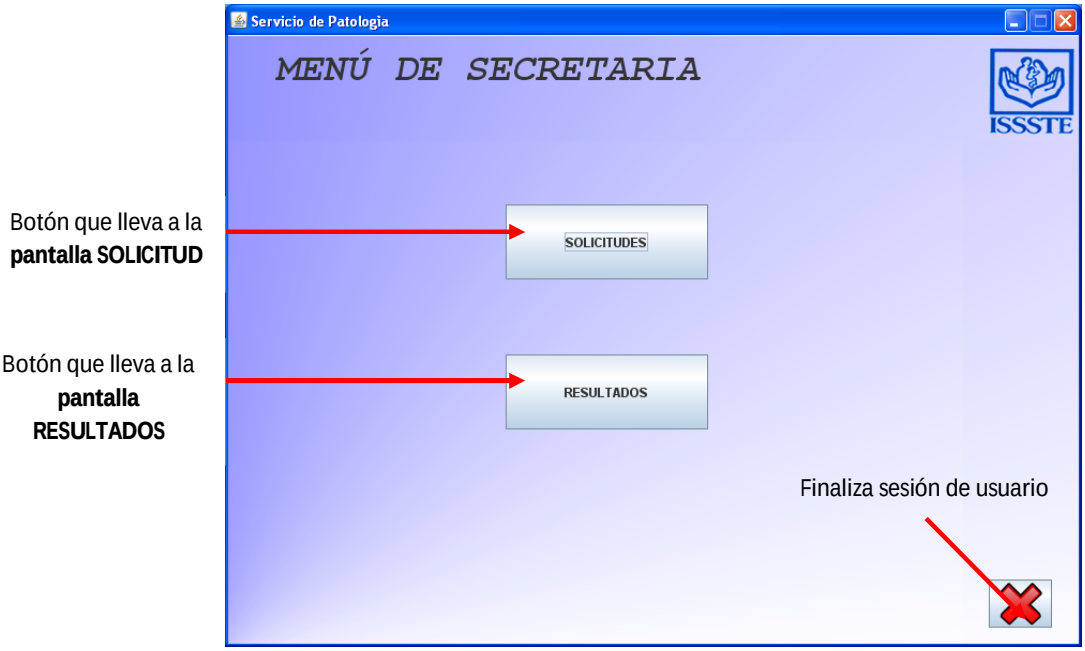
6. PANTALLA DE JEFATURA

El jefe de Departamento tiene una pantalla especial que está compuesta por los siguientes elementos:



7. PANTALLA DE SECRETARIA

El personal administrativo tendrá una pantalla especial para realizar el registro de solicitudes y obtener los informes de resultados finalizados. Está pantalla contiene los elementos que a continuación se mencionan:



BIBLIOGRAFÍA Y MESOGRAFÍA

No es la posesión de la verdad, sino el éxito
que llega luego de la búsqueda, donde
el buscador se enriquece con ella.
Max Planck

BIBLIOGRAFÍA Y MESOGRAFÍA

Andrew S. Tanenbaum (2003). *Computer Networks* (4ta ed.). Upper Saddle River, New Jersey: Prentice Hall.

Cifuentes Francisco. (n.d.). *Comparativa plataformas JAVA y .NET.* Consultado el 11 de marzo del 2010 de http://openfactory.opendigital.cl/archivo/fecha/comparativa-plataformas-java-y-net/.

Daniel Pecos. (n.d.). *PostGreSQL vs MySQL.* Consultado el 12 de marzo del 2010 de http://danielpecos.com/docs/mysql_postgres/index.html.

Dataprix. (n.d.). *Dataprix*. Consultado el 12 de marzo de 2010 de http://www.dataprix.com/.

Heikki Koivo, Mohammed Elmusrati (2009). *Systems Engineering in Wireless Communications*. Finland: Wiley.

Herb Shildt (2008). *Herb Schildt's Java: Programming Cookbook*. USA: Mc GRAW-HILL.

I. T. Hawryszkiewycz (1991). *Database analysis and design* (2da ed.). Universidad de Michigan: Macmillan Pub. Co.

Ian Gilfillan. (n.d.). *La Biblia MYSQL*. SYBEX – ANAYA multimedia.

iText in Action. Consultado el 8 de abril del 2010 de http://api.itextpdf.com/.

i*Text PDF: your Java-PDF library*. Consultado el 8 de abril del 2010 de http://itextpdf.com/.

JCalendar An flib Component. (2007). Consultado el 3 de julio del 2010 de http://flib.sourceforge.net/JCalendar/doc/tutorial.html.

JCalendar. (2008). Consultado el 30 de junio del 2010 de http://www.toedter.com/en/jcalendar/api.html.

Lockhart Thomas. (n.d.). *Guía del Programador de PostgreSQL. Consultado el 17 de marzo de 2010 de https://forja.rediris.es/docman/view.php/312/.../Postgres-Programmer.pdf.*

M. Tamer Özsu, Patrick Valduriez (1991). *Principles of Distributed Database Systems* (2da ed.). Upper Saddle River, New Jersey: Prentice Hall.

Marin, Alvaro. (n.d.). *MySQL*. Consultado el 12 de marzo del 2010 de http://www.slideshare.net/alvmarin/mysql-2467107.

Martínez Rafael. (2009). *PostgreSQL-es.org*. Consultado el 13 de marzo de 2010 de http://www.postgresql-es.org/

Mercedes Marqués. (2002). *Apuntes de Archivos y Bases de datos*. Consultado el 19 de marzo del 2010 de http://www3.uji.es/~mmarques/f47/apun/apun.html.

Mercedes Marqués. (2003).*Ciclo de vida de las aplicaciones de bases de datos*. Consultado el 24 de marzo del 2010 de http://www3.uji.es/~mmarques/f47/apun/node67.html.

Peter Rob, Carlos Coronel (2004). *Sistemas de Bases de Datos: diseño, implementación y administración* (5ta ed.). México: THOMSON.

PostgreSQL Global Development Group. (1996). *PostgreSQL*. Consultado el 12 de marzo de 2010 de http://www.postgresql.org/.

Ramez Elmasri, Shamkant B. Navathe. (2002). *Fundamentos de sistemas de bases de datos* (3ra ed.). Addison Wesley.

Roger S. Pressman (2002). *Ingeniería del Software. Un enfoque práctico* (5ta ed.). Madrid: Mc GRAW-HILL.

Segura Salazar Juan. (n.d.). *Curso de JAVA.* Consultado el 11 de marzo del 2010 http://tikal.cifn.unam.mx/~jsegura/LCGII/java1.htm#.

Sicilia Miguel Angel. (2008). *Visión general de la arquitectua de MySQL 5.1*. Consultado el 11 de marzo del 2010 de http://cnx.org/content/m18938/latest/.

Silberschatz, Korth, Sudarshan (2007). *Fundamentos de Diseño de Bases de Datos* (5ta ed.). Aravaca, Madrid: Mc GRAW-HILL.

Tutorial de Redes WIFI. (n.d.) Consultado el 17 de marzo del 2010 de http://www.scribd.com/doc/3265274/Tutorial-Redes-WiFi-telefonica.

W. Clay Richardson, Donald Avondolio, Scot Schrager, Mark W. Mitchell, and Jeff Scanlon (2007). *Professional Java, JDH 6 Edition*. Crosspoint Boulevard Indianapolis: Wiley Publishing, Inc.