UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA
UNAN-LEÓN
Facultad de Ciencias y Tecnología

"DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE EXÁMENES DEL LABORATORIO CLÍNICO TESMER"

Monografía para optar al Título de Ingeniero en Sistemas de Información

Autor:

Br. Ricardo Manuel Grijalba Urrutia.

Tutor:

Ing. Denis Leopoldo Espinoza Hernández.

León, Diciembre del 2010

AGRADECIMIENTOS

Le agradezco primeramente a ***Dios***; por darme la vida y por permitirme llegar a culminar mi trabajo**.**

A mis ***padres*** por su ayuda incondicional, dedicación y confianza, ya que sin ellos no estaría logrando este sueño.

A mi ***tutor*** Ing. Denis Espinoza, por brindarme su tiempo, paciencia y colaboración para terminar este trabajo.

A mis ***familiares*** y ***amigos*** por su apoyo y motivación para que siguiera adelante.

Y a todos aquellos que de una u otra manera me desearon éxitos en mi tesis.

DEDICATORIA

La presente tesis se la dedico con todo el amor, cariño y respeto a:

Dios: por haberme regalado la vida y la fuerza que me impulsó a seguir adelante a pesar de todas las dificultades.

Con mucho amor a mi **madre Daysi María Urrutia López** por haberme dado todo lo que necesitaba y estar a mi lado, brindándome su amor, apoyo y fortaleza.

A mi padre Luis Enrique Grijalba Ramírez por darme el apoyo que todo hijo necesita y brindarme el respaldo económico para cumplir esta meta.

A mis hermanos Jorge Luis, Luis Enrique y Luis Alberto que a pesar de todos los problemas siempre están conmigo brindándome su amor, respeto y respaldo.

A mi tutor por brindarme la ayuda, el tiempo y sobre todo la paciencia necesaria para lograr esta meta.

A mis compañeros de otras tesis, que me ayudaron ha aclarar mis dudas.

A mis amigos José Luis Vargas y José René Gutiérrez por ser gente que estuvo dispuesta gratuitamente a darme su apoyo y sus consejos.

A mi novia Hypatia Alexandra Rosales Valle, que estuvo conmigo siempre compartiendo los momentos de tristes y alegres en un recorrido de subidas y bajadas.

A todos ustedes se la dedico porque de alguna u otra forma llegaron a ser parte de esta carrera que hoy está llegando a su meta.

Y a todas las personas que aportaron un granito de arena a este trabajo.

"***GRACIAS SEÑOR POR DARME LA SALUD Y LAS FUERZAS EN ESTE CAMINO***"

INDICE

I. INTRODUCCIÓN 1
II. ANTECEDENTES 3
III. JUSTIFICACIÓN 5
IV. OBJETIVOS 7
V. DISEÑO METODOLÓGICO 9
VI. MARCO TEÓRICO 12
VII. ESPECIFICACIÓN DE REQUISITOS DEL SISTEMA 33
VIII. CASOS DE USO 66
IX. ROLES DE LOS USUARIOS 74
X. DIAGRAMA DE CLASES 76
XI. DIAGRAMAS DE SECUENCIA 78
XII. DIAGRAMA RELACIONAL 97
XIII. DISEÑO DE DATOS 99
XIV. MAPA DEL SITIO 103
XV. DISEÑO DE INTERFAZ 105
XVI. CONCLUSIONES 113
XVII. RECOMENDACIONES 115
XVIII. BIBLIOGRAFÍA 117
XIX. ANEXOS 119

ÍNDICE DE FIGURAS

Figura 1 Modelo en cascada 11
Figura 2: Modelo de funcionamiento 19
Figura 3: Interfaz de insertar paciente 23
Figura 4: Usuario Administrador 67
Figura 5: Usuario Administrador (Continuación) 68
Figura 6: Usuario Administrador (Continuación) 69
Figura 7: Usuario Administrador (Continuación) 70
Figura 8: Usuario Administrador (Continuación) 71
Figura 9: Usuario Operador 72
Figura 10: Usuario Operador (continuación) 73
Figura 11: Usuario Paciente 73
Figura 12: Roles de los usuarios 75
Figura 13: Diagrama de clases 77
Figura 14: Diagrama relacional 98
Figura 15: Mapa del sitio 104
Figura 16: Interfaz de Ingresar a la aplicación 106
Figura 17: Interfaz de administración 106
Figura 18: Interfaz de insertar paciente 107
Figura 19: Interfaz de menú de navegación 107
Figura 20: Interfaz de buscar paciente 108
Figura 21: Interfaz de resultado de exámenes 108
Figura 22: Interfaz de resultado de exámenes por grupo 109
Figura 23: Interfaz de crear examen 110
Figura 24: Interfaz de insertar datos de examen 110
Figura 25: Interfaz de exámenes por paciente 111
Figura 26: Interfaz de ver usuarios 111
Figura 27: Interfaz de eliminar paciente 112
Figura 28: Interfaz de restaurar base de datos 112
Figura 29: Instalador de WAMP 120
Figura 30: Probando WAMP 121
Figura 31: Clases del diagrama-parte1 122
Figura 32: Clases del diagrama-parte2 123
Figura 33: Clases del diagrama-parte3 123
Figura 34: Clases del diagrama-parte4 124

I. INTRODUCCIÓN

El área de la salud es vital para cada ciudadano en nuestro país y su rol es cada vez mayor debido al surgimiento de nuevas enfermedades que golpean el cuadro clínico de cada ser humano. Brindar un diagnóstico correcto es la misión de cada laboratorio.

Debido a que nos encontramos en una era llena de grandes avances tecnológicos es de gran importancia el desarrollo de una aplicación que permita gestionar de manera más ágil los diferentes procesos.

En el presente trabajo, se explica cómo se llevó a cabo la realización de este proyecto, el cual se encargará de gestionar los exámenes que se realizan los pacientes en el Laboratorio Clínico TESMER.

II. ANTECEDENTES

En el departamento de computación no se ha implementado ningún sistema que lleve a cabo este tipo de tareas. Por lo tanto, es la primera vez que se construye un proyecto de esta naturaleza tecnológica.

Actualmente en el Laboratorio Clínico TESMER la gestión tanto de los exámenes como de sus pacientes se realiza manualmente.

III. JUSTIFICACIÓN

La razón por la cual se decidió desarrollar este trabajo, es proporcionar al Laboratorio Clínico TESMER una aplicación que les ayude a controlar de una manera más rápida y eficiente la gestión de los pacientes y los exámenes que en dicho laboratorio se realizan.

IV. OBJETIVOS

4.1 Objetivo general:

- Desarrollar una aplicación Web que gestione los exámenes realizados a los pacientes en el Laboratorio Clínico TESMER.

4.2 Objetivos Específicos:

- Gestionar los diferentes exámenes que se realizan los pacientes en el Laboratorio Clínico TESMER.

- Respaldar y restaurar la base de datos de la aplicación siempre que lo soliciten usuarios con el Rol de Administrador.

- Generar reportes, de una manera ordenada y segura, de cada uno de los pacientes con sus respectivos exámenes.

- Definir roles de usuario que permitan la manipulación ordenada y segura del sistema.

V. DISEÑO METODOLÓGICO

Para llevar a cabo el desarrollo del presente trabajo se implemento la metodología del ciclo de vida clásico o en cascada.

Entiéndase por ciclo de vida todas aquellas fases que asocian una serie de tareas que se debieron realizar incluyendo los documentos que cada una de estas generan, y que sirvieron de entrada a las siguientes fases.

5.1 Modelo en Cascada

Descompone el proceso de desarrollo en diferentes fases, constituyendo la salida de cada una de ellas la entrada requerida por la siguiente. En este modelo se supone que todos los requisitos son conocidos y comprendidos perfectamente al iniciar el desarrollo del software.

5.2 Actividades del Ciclo de Vida en Cascada

- **Análisis:** Se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada ERS (documento de especificación de requisitos), que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos. Es importante señalar que en esta fase se debe reunir todo lo que se requiere del sistema y será aquello lo que seguirá en las siguientes fases, no pudiéndose requerir nuevos resultados a mitad del proceso de elaboración del software.

- **Diseño:** Se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el Documento de Diseño de Datos, que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como también la manera en que se combinan unas con otras. Otros documentos que surgen de esta fase son el diagrama de clase, el diagrama relacional, el diseño arquitectónico, y el mapa del sitio.

- **Codificación:** Es la fase de programación o implementación propiamente dicha. Aquí implementa el código fuente, haciendo uso de prototipos así como pruebas y ensayos para corregir errores. Dependiendo del lenguaje de programación y su versión se crean las bibliotecas y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido.

- **Prueba:** El software debe ser sometido a una serie de pruebas con el objetivo descubrir los defectos que puedan existir en la función, en la lógica y en la implementación del mismo.

- **Mantenimiento:** La fase de mantenimiento se centra en el cambio. Esta fase aplica lo pasos de las fases de definición y de desarrollo, pero en el contexto del software ya existe. Durante la fase de mantenimiento se centran tres tipos de cambios: corrección, adaptación y mejora.

El modelo de **ciclo de vida clásico** se representa gráficamente:

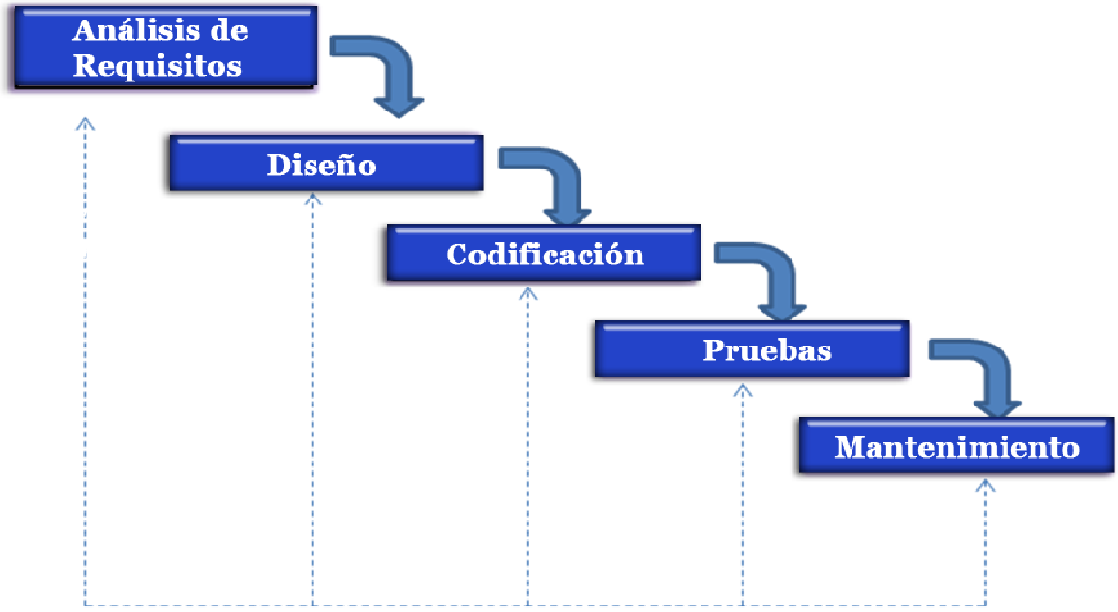


Figura 1 Modelo en cascada

5.3 Materiales:

a) **Hardware**: Para el desarrollo y la elaboración del software se utilizó 1 ordenador personal con las siguientes características:

- ✓ Microprocesador Intel Pentium IV 2.4GHz.
- ✓ Disco duro 120 Gb.
- ✓ Memoria RAM 1 GB.

b) **Software:** las herramientas software empleadas fueron:

- ✓ Rational rose 7.0.
- ✓ Win XP-SP2 y Win7 (S.O.).
- ✓ Adobe DreamWeaver CS3.
- ✓ Microsoft Office 2007.
- ✓ WAMP 5.0.
- ✓ Embarcadero ERStudio 8.0
- ✓ Adobe Photoshop CS3.
- ✓ Adobe Ilustrador CS3.

VI. MARCO TEÓRICO

Hoy en día, Internet es un medio de comunicación pública, cooperativa y autosuficiente en términos económicos, accesible por cientos de millones de personas en el mundo entero.
Físicamente, Internet emplea una gran parte del total de recursos actualmente existentes en redes de Telecomunicaciones.

Uno de los servicios que más éxito ha tenido en Internet ha sido la World Wide Web (WWW, o "la Web"), hasta tal punto que es habitual la confusión entre ambos términos.

World Wide Web (WWW) (Red Global Mundial**)** o simplemente Web, es el universo de Información accesible a través de Internet, una fuente inagotable del conocimiento humano.

El componente más usado en Internet es definitivamente la Web. En la mayoría de los Sitios Web, ciertas palabras aparecen en texto de otro color diferente al resto del documento. Por lo general, este texto es subrayado. Al seleccionar una palabra o frase, el visitante es transferido al sitio o página relacionada a esa frase. En algunas ocasiones hay botones, imágenes, o porciones de imágenes que pueden activarse mediante un clic.

6.1 Página Web

Una página de Internet o página Web es un documento electrónico que contiene información específica de un tema en particular y que es almacenado en algún sistema de Computo denominados como Servidores Web, que se encuentre conectado a la red mundial de información denominada Internet, de tal forma que este documento pueda ser consultado por cualquier persona que se conecte a esta red y que cuente con los permisos apropiados para hacerlo.

6.1.1 Una página Web es la unidad básica del World Wide Web

Una página Web tiene la característica peculiar de que el texto se combina con imágenes para hacer que el documento sea dinámico y permita que se puedan ejecutar diferentes acciones, una tras otra, a través de la selección de texto remarcado o de las imágenes, acción que puede conducir a otra sección dentro del mismo documento, abrir otra página Web, iniciar un mensaje de correo electrónico o transportarnos a otro Sitio Web totalmente distinto a través de sus hipervínculos.

6.2 Aplicación Web

En la ingeniería de software se denomina aplicación Web a aquellas aplicaciones sobre la cual los usuarios pueden utilizar accediendo a un servidor Web a través de Internet o de una intranet mediante un navegador.

Las aplicaciones Web son populares debido a lo práctico del navegador Web como cliente ligero, así como a la facilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los Web mails, wikis, weblogs, tiendas en línea que son ejemplos bien conocidos de aplicaciones Web.

Es importante mencionar que una aplicación Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

6.3 Lenguaje HTML

HTML**,** siglas de HyperText Markup Language (*Lenguaje de Marcas de Hipertexto*), es el lenguaje de marcado predominante para la construcción de páginas Web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un *script* (por ejemplo Java script), el cual puede afectar el comportamiento de navegadores Web y otros procesadores de HTML.

```
<!--Documento de Html-->
<HTML>  <!--inicio de html-->
<HEAD>  <!--Apertura de la cabecera del documento de html-->
<TITLE>   Aquí va el título de la página</TITLE>  <!--declaramos el título del documento-->
</HEAD>  <!--Cierre de la cabecera del documento de html-->
<BODY>    <!--Apertura del cuerpo del documento de html-->
        Aquí va el cuerpo de la página
</BODY>   <!--Cierre del cuerpo de html-->
</HTML>   <!--Cierre del documento de html-->
```

6.4 Javascript

Javascript es un lenguaje de programación interpretado, es decir, que no requiere Compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Al igual que Java, Javascript es un lenguaje orientado a objetos propiamente dicho, ya que dispone de Herencia, si bien esta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Java script (en contra de lo que se podría suponer) es totalmente distinto de java. Java crea programas totalmente independientes y operativos, Java Script es más sencillo porque lo único que permite es insertar código especial dentro del HTML de una página, su función es ampliar las posibilidades de HTML. Java Script no crea programas independientes, dependen por completo del código HTML de la página.

El siguiente ejemplo es un documento HTML que incorpora código Java Script.

```
<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
<!--
Programa JavaScript
//-->
</SCRIPT>

En este ejemplo se ha utilizado la variable nombre para guardar el nombre del usuario.
<HTML>
<HEAD><TITLE>Ejemplo: uso de una variable</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
var name = prompt("Introduce tu nombre:","Nombre")
//--> // comentario de una línea
</SCRIPT></HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">

<!—
//Para escribir la cadena podemos usar
document.write('<IMG SRC="welcome.gif">') Como el nombre de la imagen lleva comillas dobles,
el texto pasado al método write lleva comillas simples.
document.write("<H1>Hola " + name + ". !Bienvenido a mi página!</H1>")
//-->
</SCRIPT></BODY></HTML>
```

6.5 Hojas de estilo

Las hojas de estilo en cascada (*Cascading Style Sheets*, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). La idea que se encuentra detrás del desarrollo de CSS es separar la *estructura* de un documento de su *presentación*.

Se denominan "hojas de estilo en cascada" porque se pueden definir múltiples hojas y los estilos pueden aplicarse a todas las paginas (con un sistema predefinido para resolver conflictos).

El principio de Las hojas de estilo consiste en la utilización de un solo documento para almacenar las características de presentación de las páginas asociadas a grupos de elementos. Esto implica nombrar un conjunto de definiciones y características de presentación de las páginas, y activar esos nombres para aplicarlos a una parte del texto. En el siguiente ejemplo, se muestra un documento donde no se aplica CSS luego mostraremos más adelante otro documento en el que se aplicará CSS:

```
index.html
<html>
<head>
<title>Bienvenidos a las hojas de estilo </title>
</head>

<body>
<br><br>
<center><p > Sobre las hojas de estilo
<br><div >
<br>
<!--Aquí se crea el formulario-->
<form>
Nombre: <input name="nombre" type="text"><br><br>
Clave: <input name="nombre"  type="password"><br><br>
<center><button name="Enviar">Entrar</button></center>
</form><!--Cierre del formulario-->
<p> Si introduces tu nombre y tu clave accederás al sistema
<br>
</div>
<div >
<p>Esta es una prueba de Hojas de Estilo Puedes adornarla como quieras.
</div>
</center></body><!--Cierre del cuerpo de html--></html><!--Cierre del documento de html-->
```

6.6 PHP

PHP es un lenguaje de programación que se ejecuta en servidores Web (que incluye un servidor de aplicaciones) y permite crear paginas HTML de forma dinámica. Las siglas PHP provienen de Personal Home Page, que podemos traducir como Procesador personal de páginas Web o de Hipertexto.

El código de PHP está incluido en tags especiales "<?php,?>". Veamos un ejemplo:

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <?php echo "Hi, Esto es un Script PHP";?>
  </body>
</html>
```

6.7 AJAX

AJAX, acrónimo de **A**synchronous **J**ava Script **A**nd **X**ML (Java Script asíncrono y XML), es una técnica de desarrollo Web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador Web de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre ciertas secciones de las páginas sin necesidad de recargarlas por completo, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

6.8 MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario, es la base de datos Open Source más popular y su continuo desarrollo y su creciente popularidad está haciendo de MySQL un competidor cada vez más directo de gigantes en la materia de las bases de datos como Oracle. MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales.

6.9 JQUERY

JQuery es una biblioteca o framework de Javascript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la

tecnología AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC.

JQuery es software libre y de código abierto, posee un doble licenciamiento bajo la licencia MIT y de la GNU General Public License, Versión 2.1 JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en Javascript que de otra manera requerirían de mucho más código. Es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Veamos un ejemplo de cómo usar JQUERY:

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
<input type=”text” id=”enviar” value=”Hola mundo”/>
  </body>
</html>


$(document).ready(function(){
alert($(‘#enviar’).val());
});
```

El símbolo de dólar “$” es el sinónimo de llamar a la función JQUERY, es decir es lo mismo escribir $(document)… que JQUERY(document)…

$(document).ready(Nombre de la function o anonima) es similar a llamar al evento onload dentro de la etiqueta body.

Para acceder al contenido del input se utiliza siempre la función JQUERY “$” y colocando entre paréntesis el id del atributo input, en nuestro ejemplo se llama enviar. Si queremos leer el valor del input simplemente llamamos a la función “val()”. El símbolo de numeral (#) se coloca porque es un selector de tipo id. Si fuera de clase o de etiqueta no se le coloca.

Dentro de las llaves de la función podemos escribir todo el código javascript deseado.

6.10 Modelo de funcionamiento

Figura 2: Modelo de funcionamiento

6.10.1 Interfaz de usuario (HTML)

Es la capa más alta del modelo, la cual define la operación que desea realizar el usuario, además es la encargada de mostrar sus respectivos resultados en pantalla.

6.10.2 Javascript

Esta capa tiene como función primaria validar las entradas que proporciona el usuario mediante el uso del framework del lado del cliente llamado JQuery y como función secundaria realizar llamadas al servidor mediante AJAX que permita una respuesta más rápida a la solicitud del usuario.

6.10.3 Receptor de petición

Esta capa es la responsable de recibir las peticiones que realice el usuario y de invocar a la interface necesaria para realizar su tarea respectiva.

6.10.4 Interfaces

Esta capa es la responsable de realizar las depuraciones del lado del servidor de entradas que proporciona el usuario, así mismo de invocar a la operación que muestre su interfaz.

Además, esta capa maneja la Validación de Sesión del usuario.

6.10.5 Operaciones

Esta capa es el más baja del modelo y es la responsable realmente de ejecutar las operaciones elegidas por el usuario. Además es la que establece conexión a la base de datos y opera sobre ella.

6.11 Ejemplo del modelo de funcionamiento

Para tener una visión mucho más clara y entender cómo funciona nuestra aplicación, explicaré paso a paso los procesos que se llevan a cabo para insertar un paciente y su correcto uso.

6.11. 1 Interfaz de Usuario (HTML)

Antes de explicar el primer paso, suponemos que el usuario ya esta identificado para realizar esta labor.

El primer paso es simplemente solicitar la petición que permita mostrar la interfaz que se utilizará para ingresar los datos. Tal interfaz es un formulario cuyo código es el siguiente:

```
<div id="contenedorPaciente">
   <div id="formulario">
<form action="<?php echo $editFormAction;?>" id="form_paciente" name="form_paciente"
method="POST">

<label for="nombres">Nombres:<cite class='error'>  <?php if ($error)   echo
$mostrar_error[$i++] ?></cite></label>
          <input value="<?php  if(isset($_POST['nombres'])){
          echo htmlentities($_POST['nombres']);}
          ?>" name="nombres" type="text" class="textInput" id="nombres" />

<label for="apellidos">Apellidos:<cite class='error'>  <?php if ($error) echo
$mostrar_error[$i++] ?></cite></label>
          <input value="<?php  if(isset($_POST['apellidos'])){
          echo htmlentities($_POST['apellidos']);}
          ?>" name="apellidos" type="text" class="textInput" id="apellidos" />

<label>Fecha de Nacimiento:</label>
          <input  readonly="readonly" value="<?php
          if(!$_POST) {
          echo 'martes';
          }
          else      echo htmlentities($_POST['edad']);
           ?>" name="edad" type="text" class="textInput" id="edad" />

<label for="sexo">Sexo:</label>
                    <div class="sexo">
          <input <?php if ($_POST && !(strcmp($_POST['sexo'],"m"))) {echo
"checked=\"checked\"";} ?> type="radio" name="sexo" value="m"  /> Masculino
<input <?php if (($_POST && !(strcmp($_POST['sexo'],"f"))) || !$_POST) {echo
"checked=\"checked\"";} ?> type="radio" name="sexo" value="f" />
```

```
        Femenino
 </div>

<label for="lugar">Lugar:<cite class='error'>  <?php if ($error) echo
$mostrar_error[$i++] ?></cite></label>
        <input value="<?php  if(isset($_POST['lugar'])){
        echo htmlentities($_POST['lugar']);}
        ?>" name="lugar" type="text" class="textInput" id="lugar" />

<label for="direccion">Direcci&oacute;n:<cite class='error'>  
        <?php if ($error) echo $mostrar_error[$i++] ?></cite></label>
        <textarea name="direccion" cols="5" rows="5" id="direccion">
        <?php if(isset($_POST['direccion'])){
        echo $_POST['direccion'];} ?></textarea>

<label for="telefono">Tel&eacute;fono:<cite class='error'>  
        <?php if ($error) echo $mostrar_error[$i++] ?></cite></label>
        <input name="telefono" type="text" class="textInput" id="telefono"   value="<?php
if(isset($_POST['telefono'])){
        echo htmlentities($_POST['telefono']);}
        ?>"/>

<input style="background-color:#006699; color:#00FFFF; border:2px outset #0099FF; margin-
left:30px;" name="enviar"
type="submit" class="enviar1" id="enviar" value="Ingresar" />

<input style="background-color:#006699; color:#00FFFF; border:2px outset #0099FF;margin-
left:10px" type="button"
        value="Cancelar"  onclick="
        <?php if(isset($_SESSION['admin']) && $_SESSION['admin_priv']=='s'){?>
        window.location='../Paciente/peticion.php?opcion=180';
        <?php
        }
        else{
        ?>
        window.location='../Paciente/peticion.php?opcion=306';
        <?php
        }
        ?>" />
```

```
<input type="hidden" name="MM_insert" value="form_paciente" />
</form>
</div><!--Fin del div#formulario--></div>
```

Al final el usuario ve la siguiente interfaz:

Figura 3: Interfaz de insertar paciente

6.11. 2 Validación Javascript

El segundo paso es la validación javascript. Su forma de realizarlo es:

El código Javascript se ejecuta cuando un elemento de formulario pierde el foco, inmediatamente valida su entrada, de ser correcta continuará la misma tarea con el resto de componentes. Si un dato es inválido, simplemente se le muestra un mensaje de error. Además, el botón ingresar solamente está disponible si todas las entradas tienen datos. El código javascript de la función de validación es el siguiente:

```
//funciones de validacion
function validar_string(index,actual){
//Eliminar los espacios en Blanco
actual.val($.trim(actual.val()));
            if(actual.val().length <4){
                $('cite').eq(index).text('');
               $('cite').eq(index).html(' M&iacute;nimo de 4 caracteres.');
              actual.css('backgroundColor','#CC6699');
                enviar.attr('disabled','disabled');
              return false;
                                        }
           else if(!isNaN(actual.val())){
                $('cite').eq(index).text('');
                $('cite').eq(index).html(' Error de escritura.');
                 actual.css('backgroundColor','#CC6699');//nombres.addClass("error");
                 enviar.attr('disabled','disabled');
                 return false;
                                          }
           // SI longitud, SI caracteres A-z
            else{      $('cite').eq(index).text('');
                       actual.css('backgroundColor','white');
if(nombres.val().length >0 && apellidos.val().length >0 && lugar.val().length >0 &&
telefono.val().length >0 && direccion.val().length >0)
                                enviar.removeAttr('disabled');
 else
                                enviar.attr('disabled','disabled');
                               return true;
                  }
}

function validar_telefono(){      //Eliminar los espacios en Blanco
            telefono.val($.trim(telefono.val()));
                  if(telefono.val().length < 8 )
                    {  $('cite').eq(5).text('');
                       $('cite').eq(5).html(' M&iacute;nimo de 8 n&uacute;meros.');
                     telefono.css('backgroundColor','#CC6699');          return false;
                    }
            //SI longitud pero NO solo caracteres A-z
            else if(!telefono.val().match(/^[0-9]{4}\-[0-9]{4}$/)){
                      $('cite').eq(5).text('');
                      $('cite').eq(5).html(' Error. Formato Correcto:####-####.');
```

```
            telefono.css('backgroundColor','#CC6699');
            return false;
        }
        // SI longitud, SI caracteres A-z
        else{
             $('cite').eq(5).text('');
            telefono.css('backgroundColor','white');
             return true;
        }
        }
        nombres.blur(function(){
            validar_string(0,$(this));
        });
        apellidos.blur(function(){
            validar_numeros(1,$(this));
        });
        lugar.blur(function(){
             validar_numeros(2,$(this));
        });
        direccion.blur(function(){
           validar_numeros(3,$(this));
        });
        telefono.blur(function(){
           validar_numeros(4,$(this));
        });
```

6.11. 3 Receptor de petición

El tercer paso consiste en llamar a la petición que muestra la interfaz para ingresar el paciente. Su código es el siguiente:

```
        <?php
        require_once('Interface.php');
        $nuevo=new Interface1();//Invoca a la interfaz de Pacientes
          if(isset($_GET['opcion']))
                {
                        $opcion=$_GET['opcion'];
                          if($opcion==2)
                              $nuevo->formInsertPaciente();
                }?>
```

6.11. 4 Interfaces

El cuarto paso: Debemos asegurarnos que solamente los encargados de la aplicación puedan trabajar sobre ella legítimamente, esto lo haremos comprobando si es un usuario registrado validando la sesión:

```
if (!isset($_SESSION))
 session_start();
        if(!isset($_SESSION['admin']))
        {
           $updateGoTo="../Paciente/peticion.php?opcion=400&tipo_error=3";
          header(sprintf("Location: %s", $updateGoTo));
        }
```

Una vez seguros de que el usuario es legítimo, debemos por seguridad de la Base de datos realizar validaciones en el servidor que nos garanticen que los datos son correctos. Esto se logra con el siguiente código:

```
public function validar_nombres($nombres)
{
        $nombres=trim($nombres);
        if(empty($nombres)|| strlen($nombres) < 4 || !preg_match("/^[a-z A-Z]+$/", $nombres))
        {
           $this->error[$this->contador++]="M&iacute;nimo de 4 letras (No use
n&uacute;meros).";
        }
        else
        $this->error[$this->contador++]="";
}//Fin de la Fn validar_nombres

public function validar_ apellidos ($apellidos)
{
    $ apellidos=trim($apellidos);
        if(empty($apellidos)|| strlen($apellidos) < 4 || !preg_match("/^[a-z A-Z]+$/", $nombres))
        {
        $this->error[$this->contador++]="M&iacute;nimo de 4 letras (No use   n&uacute;meros).";
        }
        else
        $this->error[$this->contador++]="";
}//Fin de la Fn validar_ apellidos
```

```
public function validar_nick($nick)//Tambien valida el lugar
{
  $nick=trim($nick);
   if(empty($nick)|| strlen($nick) < 3 || !preg_match("/^[a-z A-Z0-9]+$/", $nick))
  {
     $this->error[$this->contador++]="1)M&iacute;nimo de 3 caracteres. 2)S&oacute;lo   caracteres
alfanum&eacute;ricos";
  }
   else
     $this->error[$this->contador++]="";
}//Fin de la Fn validar_nick

public function validar_direccion($direccion)
{
            if(empty($direccion)|| strlen($direccion) < 4 )
            {
                $this->error[$this->contador++]="M&iacute;nimo de 4 caracteres.";
            }
            elseif(!is_string($direccion) || is_numeric($direccion))
                 $this->error[$this->contador++]="Error de escritura.";
            else
                 $this->error[$this->contador++]="";

}//Fin de la Fn validar_direccion

public function validar_telefono($telefono)
{
              $telefono=trim($telefono);
            if(empty($telefono)|| strlen($telefono) < 4 || !preg_match("/^[0-9]{4}\-[0-9]{4}$/",
$telefono))
            {
               $this->error[$this->contador++]="Error. Formato Correcto: ####-####";
            }
            else
                $this->error[$this->contador++]="";

}//Fin de la Fn validar_telefono
```

Luego de estar seguros de que los datos son los correctos, simplemente llamamos a la función que recibirá los datos para ser procesados. Su código es el siguiente:

```
$paciente=new Paciente();
$result = $paciente->insertar_paciente($_POST['nombres'],$_POST['apellidos'], $paciente, $clave
,$edad,$_POST['sexo'], $_POST['lugar'], $_POST['direccion'],$_POST['telefono']);
```

6.11. 5 Operaciones:

El quinto paso: Con los datos procedentes de la interfaz, simplemente realizamos el proceso de inserción en la Base de datos. Su código es el siguiente:

```
<?php require_once("../Connections/conexion_tesmer.php"); ?>
public function insertar_paciente($nombres, $apellidos, $paciente, $clave, $edad, $sexo, $lugar,
$direccion, $telefono)
        {
                $insertSQL = sprintf("INSERT INTO tb_paciente (nombres, apellidos, paciente,
clave, edad, sexo, lugar, direccion, telefono)
                VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)",
              Conectar::GetSQLValueString($nombres, "text"),
              Conectar::GetSQLValueString($apellidos, "text"),
             Conectar::GetSQLValueString($paciente, "text"),
             Conectar::GetSQLValueString($clave, "text"),
              Conectar::GetSQLValueString($edad, "date"),
              Conectar::GetSQLValueString($sexo, "text"),
               Conectar::GetSQLValueString($lugar, "text"),
              Conectar::GetSQLValueString($direccion, "text"),
              Conectar::GetSQLValueString($telefono, "text"));
          //La Fn Conectar::GetSQLValueString() evita la inyección SQL
              $id_link = Conectar::con();  $result = @mysql_query($insertSQL, $id_link);
            if (!$result)    {
              echo 'No se puede ejecutar la consulta insertar_paciente: '  . mysql_error();
                                    Conectar::cerrar_conexion($result);
                                    return false;
                              }
            return $result;}//Fin de la Funcion insertar_paciente
```

Antes de insertar los datos, éstos son preparados para evitar la posible inyección SQL. Este tema se abordará en el apartado de **Seguridad de la Aplicación**

Para realizar la tarea de inserción es necesario que se conecte a la Base de datos. Para hacerlo veamos su código:

```
public  static function con()
 {
          $id_link = mysql_pconnect("localhost","tesmer","tesmer");
         mysql_query("SET NAMES 'utf8'");
          if (!$id_link)
          {
             die('No conectado al Servidor : ' . mysql_error());
             return false;//Si la conexion  al Servidor falla devolvemos False
          }
          $db_selected = @mysql_select_db("db_termer",$id_link);
          if(!$db_selected)
          {
              die ('No puede usar la DB db_termer :' .mysql_error());
              return false;//Si la conexion a la Base de datos falla devolvemos False
          }
          return $id_link;
 }
```

6.12 Seguridad en la aplicación

6.12.1 Evitar Inyección SQL.

Inyección SQL es una vulnerabilidad informática en el nivel de la validación de las entradas a la base de datos de una aplicación. El origen es el filtrado incorrecto de las variables utilizadas en las partes del programa con código SQL. Es, de hecho, un error de una clase más general de vulnerabilidades que puede ocurrir en cualquier lenguaje de programación o de script que esté incrustado dentro de otro.

Una vez visto en qué consiste la Inyección SQL, vamos a explicar el método que utilizamos para evitar dicho peligro en esta aplicación Web.

En la función de ingresar los datos de un usuario encontramos embebida la Fn Conectar::GetSQLValueString, la cual permite evitar la inyección SQL, es decir siempre se prepara la consulta justo antes de ejecutar una instrucción mysql_query. Su código es el siguiente:

```
public static function GetSQLValueString($theValue, $theType, $theDefinedValue = "", $theNotDefinedValue = "")
```

```
{
$conectar=Conectar::con();

//Creo de nuevo un objeto de la clase Conectar, porque las funciones:
//mysql_real_escape_string y mysql_real_escape_string necesitan comunicarse
//con el Server, ya que al llamarlas de otros script
//se pierde el ámbito de la primera conexión, por eso es necesaria otra conexión.

$theValue = get_magic_quotes_gpc() ? stripslashes($theValue) : $theValue;
  if(function_exists("mysql_real_escape_string"))
       $theValue = mysql_real_escape_string($theValue);
   else
       $theValue = mysql_escape_string($theValue);
  switch ($theType) {
    case "text":
      $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
      break;
    case "long":
    case "int":
      $theValue = ($theValue != "") ? intval($theValue) : "NULL";
      break;
    case "double":
      $theValue = ($theValue != "") ? "'" . doubleval($theValue) . "'" : "NULL";
      break;
    case "date":
      $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
      break;
    case "defined":
      $theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue;
      break;
   }
   return $theValue;
}
```

Para llamarla veamos este ejemplo sólo con 2 parámetros:

```
$insertSQL = sprintf("INSERT INTO tb_paciente (nombres, apellidos)      VALUES (%s, %s)",
                Conectar::GetSQLValueString($nombres, "text"),
```

```
Conectar::GetSQLValueString($apellidos, "text");
//La Fn Conectar::GetSQLValueString() evita la inyección SQL
 $id_link = Conectar::con();
 $result = @mysql_query($insertSQL, $id_link);
```

En mi caso esta función es estática de la clase Conectar.

6.12.2 Encriptación SHA1.

SHA-1 produce un 160-bits para digerir un mensaje con una longitud máxima de (264 - 1) bits. Este se basa en principios similares a los utilizados por Ronald L. Rivest del MIT, en el diseño del mensaje MD4 y MD5 algoritmos, pero tiene un diseño más conservador.

Hoy en día la mayoría de las páginas web utilizan bases de datos para poder desarrollar portales dinámicos y así hacerlos más atractivos a la vez que útiles. Pero esta información que se guarda en la base de datos tiene que tener algún tipo de protección. Es por ello que algunos campos se guardan encriptados en la base de datos, principalmente cuando una página requiere el nombre de usuario y contraseña, esta última se encripta y se guarda en la Base de datos.

En PHP se utiliza la función SHA1, que es una función hash irreversible (de un sólo sentido), es decir, encripta la contraseña tecleada por el usuario y es imposible que partiendo desde la cadena encriptada se vuelva a la contraseña origen. Por esto mismo no hay problema de que alguien pueda acceder al campo encriptado de la base de datos.

En el caso específico de mi aplicación, en la base de datos se guarda la contraseña encriptado, cuando un usuario quiere acceder, habrá que realizar una comparación entre la contraseña que introduce encriptado en SHA1, y lo que tenemos en la base de datos, (que es la contraseña encriptada en SHA1), si coincide se le permite el acceso, si no, se rechaza.

Vemos un ejemplo práctico:

```
<?php
$contraseña = 'rolex';

if (sha1 ($contraseña) == '1f3870be274f6c49b3e31a0c6728957f') {
   echo "contraseña valida";
}

?>
```

6.12.3 DATOS GENERALES DEL LABORATORIO CLÍNICO TESMER

El laboratorio se encuentra ubicado en la ciudad de León-Nicaragua en el barrio el coyolar exactamente del Dr. Cayetano Munguía ½ C. al Norte. Es un empresa que lleva 4 años funcionando y que tiene un sólo propietario.

Por el momento brinda servicios de:

- Atención médica para adultos y niños.
- Realización de examen para determinar el estado clínico del paciente.
  - Parasitología.
  - Hematología.
  - Uroanálisis.

El Laboratorio lleva un control de los exámenes que se realizan por paciente mediante cuadernos de anotación, donde se le piden los datos al paciente y el tipo de examen a realizarse.

VII. ESPECIFICACIÓN DE REQUISITOS DEL SISTEMA

I. INTRODUCCIÓN

Propósito del documento
Describir el conjunto de funcionalidades, restricciones y requisitos que debe cumplir la aplicación.

Alcance
El nombre de la aplicación será "TESMER" (Tu Eres Señor Mi Escudo y Refugio)", el cual realizará las siguientes funciones:

- Ingresar a la Aplicación.
- Insertar Paciente.
- Editar Paciente.
- Eliminar Paciente.
- Buscar paciente.

- Insertar Examen Parasitología.
- Editar Examen Parasitología.
- Borrar Examen Parasitología.
- Imprimir Examen Parasitología.
- Insertar Examen Hematología.
- Editar Examen Hematología.
- Borrar Examen Hematología.
- Imprimir Examen Hematología.
- Insertar Examen Uroanálisis.
- Editar Examen Uroanálisis.
- Borrar Examen Uroanálisis.
- Imprimir Examen Uroanálisis.

- Editar Tipo Examen.
- Borrar Tipo Examen.
- Reportes Generales:
    - Ver resultados por Grupo.
    - Ver resultados Grupo-Fecha.
    - Exámenes según pacientes.

- Reportes Específicos.
    - Ver todos los pacientes.
    - Ver todos los Usuarios.
    - Ver resultados.
    - Ver resultados por fecha.
    - Ver resultados por Grupo.
    - Ver resultados Grupo-Fecha.

- Respaldar Base de datos.
- Restaurar Base de Datos.
- Insertar Usuario.
- Editar Usuario.
- Borrar Usuario.
- Salir del sistema.

Definiciones, acrónimos y abreviaturas usadas.

- BD: Base de Datos.
- RF: Requisitos Funcionales.
- UA: Usuario Administrador.
- UO: Usuario Operador.
- RE: Reporte Específico.
- UP: Usuario Paciente.

Referencias.

Ninguna.

Visión General.

Primeramente se hará una descripción general de la aplicación que se desea desarrollar para posteriormente estudiar cada una de las funciones que ésta realizará.

DESCRIPCIÓN GENERAL.

Relación con proyectos pasados

El software no interactúa con ninguna otra aplicación, ya que es la primera aplicación informativa que el Laboratorio utilizará para automatizar sus tareas.

Relación con proyectos actuales

Ninguna.

Funciones y propósitos del sistema

- Solamente los usuarios registrados pueden tener acceso a la aplicación.
- Una vez dentro del sistema el usuario puede listar reportes dependiendo de lo que quiera generar.
- El UA puede ver todos los registros de los pacientes y operar sobre ellos.
- El UA puede ver todos los registros de los exámenes de los pacientes y operar sobre ellos.
- El UA puede modificar, incluso eliminar registros de los usuarios cuando lo desee.
- El UA será capaz de disponer información exclusiva de su privilegio.
- El UA será capaz de realizar búsquedas de pacientes.

- El UO podrá realizar las mismas tareas del UA salvo las siguientes:
  - Eliminar Pacientes.
  - Eliminar Exámenes.
  - Informes Generales.
  - Modificar un Tipo Examen.
  - Eliminar un Tipo Examen.
  - Eliminar usuarios.
- El UP será capaz solamente de ver sus resultados.

Restricciones Generales

EL navegador que se utilizara será la versión Mozilla Firefox 3.0 y superiores. Únicamente podrán acceder al sistema usuarios registrados. Una vez dentro del sistema las funciones a las que se tendrán acceso dependerán de los permisos que se posean.

REQUISITOS ESPECÍFICOS.
Requisitos Funcionales

RF1 Ingresar a la aplicación.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario identificarse en la aplicación para tener acceso a las funciones según el privilegio que se le asignó en la BD.

1.1.2 Entradas:

Datos proporcionados por el usuario:

- Nombre del usuario.
- Contraseña del usuario.

1.1.3 Procesos:

Se validarán los datos introducidos y si son correctos se procederá a ingresar al sistema TESMER.

1.1.4 Salida:

Si el usuario ingresado es un UA, se le mostrará una página con el menú de administración. Si el usuario es un UO, se le mostrará la página de inicio y el menú paciente para llevar a cabo las operaciones deseadas. Si el usuario es UP se le mostrará una página con sus respectivos resultados.

1.2 Interfaces Externas:

Interfaz de Usuario:

La forma de introducir los datos será a través de un formulario que solicite al usuario su nombre y su clave para poder ingresar.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF2 Insertar paciente.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA o UO ingresar un paciente a la aplicación.

1.1.2 Entradas:

Datos Proporcionados por el Usuario:

- Nombres.
- Apellidos.
- Fecha de Nacimiento.
- Sexo.
- Lugar.
- Dirección.
- Teléfono.
- Paciente.

1.1.3 Procesos:

Se validarán los datos introducidos y luego serán insertados en la BD.

1.1.4 Salida:

Se le muestra un mensaje por pantalla que indica que el paciente ha sido insertado correctamente.

1.2 Interfaces Externas:

Interfaz de Usuario:

La forma de introducir los datos será de a través de un formulario que solicite al usuario sus datos.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF3 Editar paciente.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA o UO editar un paciente en la aplicación.

1.1.2 Entradas:

Datos Proporcionados por el Usuario:

- Número de identificación del paciente.
- Nombres.
- Apellidos.
- Fecha de Nacimiento.
- Sexo.
- Lugar.
- Dirección.
- Teléfono.
- Paciente.

1.1.3 Procesos:

Mediante una consulta los datos serán presentados, los datos serán verificados y luego actualizados en la BD.

1.1.4 Salida:

Se mostrará por pantalla un mensaje indicando que los datos se actualizaron correctamente.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón “Actualizar” que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizara cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF4 Eliminar paciente.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá solamente al usuario UA eliminar un paciente en la aplicación.

1.1.2 Entradas:

- Número de identificación del paciente.

1.1.3 Procesos:

Mediante una consulta los datos serán presentados, el usuario confirmará la operación y luego eliminados de la BD.

1.1.4 Salida:

Se actualizará la BD y se mostrará un mensaje que indica que el paciente ha sido eliminado correctamente.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón "Eliminar" que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizara cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF5 Buscar paciente.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA o UO buscar un paciente en la aplicación. El usuario puede introducir varias combinaciones de cadenas.

1.1.2 Entradas:

Datos proporcionados por el usuario:

- Nombres y/o Apellidos.

1.1.3 Procesos:

Se realizará una consulta en la BD obteniendo el registro del paciente.

1.1.4 Salida:

Se mostrará por pantalla los datos del paciente.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón "Buscar" que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF6 Insertar examen parasitología.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA o UO insertar un examen de parasitología en la aplicación.

1.1.2 Entradas:

- Color.
- Consistencia.
- Examen microscópico.
- Tinción de Wright.
- Otros.

1.1.3 Procesos:

Se validarán los datos introducidos y se insertarán en la BD.

1.1.4 Salida:

Se mostrará por pantalla un mensaje indicando que los datos del examen han sido guardados satisfactoriamente.

1.2 Interfaces Externas:

Interfaz de Usuario:

La forma de introducir los datos será de a través de un formulario que solicite al usuario sus datos.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF7 Editar examen parasitología.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA o UO editar un examen de parasitología en la aplicación.

1.1.2 Entradas:

- Número de examen.
- Color.
- Consistencia.
- Examen microscópico.
- Tinción de Wright.
- Otros.

1.1.3 Procesos:

Se realizará la consulta con los datos introducidos obteniéndose la coincidencia y se modificará el registro y se actualizará en la BD.

1.1.4 Salida:

Se mostrará por pantalla un mensaje indicando que los datos del examen han sido actualizados satisfactoriamente.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón “Actualizar” que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF8 Borrar examen parasitología.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá únicamente al usuario UA borrar un examen de parasitología en la aplicación. Se le mostrará un mensaje de confirmación.

1.1.2 Entradas:

- Número de examen.

1.1.3 Procesos:

Se realizará una consulta obteniendo la coincidencia y luego se procederá a ejecutar la eliminación del examen y actualización de la BD.

1.1.4 Salida:

Se mostrará por pantalla un mensaje indicando que los datos del examen han sido eliminados satisfactoriamente.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón “Eliminar” que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF9 Imprimir examen parasitología.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA o UO imprimir examen de parasitología en la aplicación.

1.1.2 Entradas:

- Número del examen.
- Color.
- Consistencia.
- Examen microscópico.
- Tinción de Wright.
- Otros.

1.1.3 Procesos:

Se realizará una consulta que muestre los datos del examen para ser impresos.

1.1.4 Salida:

Se mostrará por pantalla los datos obtenidos en un formulario listo para imprimir.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón “Imprimir” que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF10 Insertar examen hematología.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA o UO insertar examen de hematología en la aplicación.

1.1.2 Entradas:

- Hematócrito.
- Hemoglobina.
- Malaria.
- Recto de Plaquetas.
- Recto glóbulos Blancos.
- Neutrófilos segmentados.
- Linfocitos.
- Eosinófilos.
- Basófilos.
- Monócitos.
- Neutrófilos en banda.

1.1.3 Procesos:

Se validarán los datos introducidos y se insertarán en la BD.

1.1.4 Salida:

Se mostrará por pantalla un mensaje indicando que los datos del examen han sido guardados satisfactoriamente.

1.2 Interfaces Externas:

Interfaz de Usuario:

La forma de introducir los datos será de a través de un formulario que solicite al usuario sus datos.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF11 Editar examen hematología.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA o UO editar examen de Hematología en la aplicación.

1.1.2 Entradas:

- Número de examen.
- Hematócrito.
- Hemoglobina.
- Malaria.
- Recto de Plaquetas.
- Recto glóbulos Blancos.
- Neutrófilos segmentados.
- Linfocitos.
- Eosinófilos.
- Basófilos.
- Monócitos.
- Neutrófilos en banda.

1.1.3 Procesos:

Se realizará la consulta con los datos introducidos obteniéndose la coincidencia y se modificará el registro y se actualizará en la BD.

1.1.4 Salida:

Se mostrará por pantalla un mensaje indicando que los datos del examen han sido actualizados satisfactoriamente.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón “Actualizar” que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF12 Borrar examen hematología.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá únicamente al usuario UA borrar examen de hematología en la aplicación. Se le mostrará un mensaje de confirmación.

1.1.2 Entradas:

- Número de examen.

1.1.3 Procesos:

Se realizará una consulta obteniendo la coincidencia y luego se procederá a ejecutar la eliminación del examen y actualización de la BD.

1.1.4 Salida:

Se mostrará por pantalla un mensaje indicando que los datos del examen han sido eliminados satisfactoriamente.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón “Eliminar” que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF13 Imprimir examen hematología.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA o UO imprimir examen de hematología en la aplicación.

1.1.2 Entradas:

- Número del examen.
- Hematócrito.
- Hemoglobina.
- Malaria.
- Recto de Plaquetas.
- Recto glóbulos Blancos.
- Neutrófilos segmentados.
- Linfocitos.
- Eosinófilos.
- Basófilos.
- Monócitos.
- Neutrófilos en banda.

1.1.3 Procesos:

Se realizará una consulta que muestre los datos del examen para ser impresos.

1.1.4 Salida:

Se mostrará por pantalla los datos obtenidos en un formulario listo para imprimir.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón “Imprimir” que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF14 Insertar examen uroanálisis.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA o UO insertar examen de uroanálisis en la aplicación.

1.1.2 Entradas:

- Color.
- Aspecto.
- Densidad.
- PH.
- Nitritos.
- Sangre.
- Cetonas.
- Glucosa.
- Proteínas.
- Urobilinogeno.
- Leucoestereza.
- Bilirrubina.
- Glóbulos Blancos.
- Glóbulos Rojos.
- Células epiteliales.
- Urato amorfo.
- Fosfato amorfo.
- Hilos mucosos.
- Otros.

1.1.3 Procesos:

Se validarán los datos introducidos y se insertarán en la BD.

1.1.4 Salida:

Se mostrará por pantalla un mensaje indicando que los datos del examen han sido guardados satisfactoriamente.

1.2 Interfaces Externas:

Interfaz de Usuario:

La forma de introducir los datos será de a través de un formulario que solicite al usuario sus datos.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF15 Editar examen uroanálisis.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA o UO editar examen de uroanálisis en la aplicación.

1.1.2 Entradas:

- Número de examen.
- Color.
- Aspecto.
- Densidad.
- PH.
- Nitritos.
- Sangre.
- Cetonas.
- Glucosa.
- Proteínas.
- Urobilinogeno.
- Leucoestereza.
- Bilirrubina.
- Glóbulos Blancos.
- Glóbulos Rojos.
- Células epiteliales.
- Urato amorfo.
- Fosfato amorfo.
- Hilos mucosos.
- Otros.

1.1.3 Procesos:

Se realizará la consulta con los datos introducidos obteniéndose la coincidencia y se modificará el registro y se actualizará en la BD.

1.1.4 Salida:

Se mostrará por pantalla un mensaje indicando que los datos del examen han sido actualizados satisfactoriamente.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón "Actualizar" que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF16 Borrar examen uroanálisis.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá únicamente al usuario UA borrar examen de uroanálisis en la aplicación. Se le mostrará un mensaje de confirmación.

1.1.2 Entradas:

- Número de examen.

1.1.3 Procesos:

Se realizará una consulta obteniendo la coincidencia y luego se procederá a ejecutar la eliminación del examen y actualización de la BD.

1.1.4 Salida:

Se mostrará por pantalla un mensaje indicando que los datos del examen han sido eliminados satisfactoriamente.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón "Eliminar" que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF17 Imprimir examen uroanálisis.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA o UO imprimir examen de uroanálisis en la aplicación.

1.1.2 Entradas:

- Número del examen.
- Color.
- Aspecto.
- Densidad.
- PH.
- Nitritos.
- Sangre.
- Cetonas.
- Glucosa.
- Proteínas.
- Urobilinogeno.
- Leucoestereza.
- Bilirrubina.
- Glóbulos Blancos.
- Glóbulos Rojos.
- Células epiteliales.
- Urato amorfo.
- Fosfato amorfo.
- Hilos mucosos.
- Otros.

1.1.3 Procesos:

Se realizará una consulta que muestre los datos del examen para ser impresos.

1.1.4 Salida:

Se mostrará por pantalla los datos obtenidos en un formulario listo para imprimir.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón “Imprimir” que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF18 Editar tipo examen.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá únicamente al usuario UA editar un tipo de examen.

1.1.2 Entradas:

- Número del tipo de examen.
- Nombre del tipo de examen.

1.1.3 Procesos:

Se validarán los datos y serán introducidos en la BD.

1.1.4 Salida:

Se mostrará por pantalla un mensaje indicando que el tipo de examen ha sido actualizado satisfactoriamente en la BD.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón “Actualizar” que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF19 Borrar tipo examen.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá únicamente al usuario UA eliminar un tipo de examen en la aplicación.

1.1.2 Entradas:

- Número tipo de examen.

1.1.3 Procesos:

Se realizará la consulta con el dato pasado obteniéndose la coincidencia y se eliminará el registro y se actualizará en la BD.

1.1.4 Salida:

Se mostrará en pantalla un mensaje que indica que los datos están eliminados.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón “Eliminar” que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF20 Ver resultados por grupo.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá únicamente al usuario UA ver un informe de los exámenes en la aplicación.

1.1.2 Entradas:

Ninguna.

1.1.3 Procesos:

Se realizará una consulta en la BD, obteniéndose todos los registros por grupo de los exámenes realizados hasta el día actual.

1.1.4 Salida:

Se mostrará por pantalla los registros generados por la consulta.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón "Ver resultados por grupo" que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF21. Ver resultados grupo-fecha.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá únicamente al usuario UA ver un informe de los exámenes en la aplicación.

1.1.2 Entradas:

- Fecha Inicial.
- Fecha Final.

1.1.3 Procesos:

Se realizará una consulta en la BD según el rango de fechas, obteniéndose todos los registros por grupo-fecha de los exámenes realizados.

1.1.4 Salida:

Se mostrará por pantalla todos los registros generados por la consulta.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón "Ver resultados por grupo-fecha" que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF22 Exámenes según pacientes.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá únicamente al usuario UA ver un informe de los exámenes en la aplicación.

1.1.2 Entradas:

Ninguna.

1.1.3 Procesos:

Se realizará una consulta en la BD, obteniéndose todos los registros de los exámenes realizados.

1.1.4 Salida:

Se mostrará en pantalla una tabla que organiza la vista de los exámenes clasificados por paciente.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón "Exámenes según pacientes" que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF23 Ver todos los pacientes.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA o UO ver un informe de los pacientes en la aplicación.

1.1.2 Entradas:

Ninguna.

1.1.3 Procesos:

Se realizará una consulta en la BD, obteniéndose todos los registros de los exámenes realizados por el paciente.

1.1.4 Salida:

Se mostrará por pantalla todos los registros generados por la consulta.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón "Ver pacientes" que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF24 Ver todos los usuarios.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá únicamente al usuario UA ver a los usuarios que gestionan la aplicación.

1.1.2 Entradas:

Ninguna.

1.1.3 Procesos:

Se realizará una consulta en la BD, obteniéndose todos los registros de los usuarios existentes en el Sistema.

1.1.4 Salida:

Se mostrará por pantalla todos los registros generados por la consulta.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón "Ver usuarios" que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF25 Ver resultados.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA, UO o UP ver un informe de los exámenes según el paciente que se haya seleccionado o ingresado en la aplicación.

1.1.2 Entradas:

- Número de paciente.

1.1.3 Procesos:

Se realizará una consulta en la BD con el número del paciente, obteniéndose todos los datos existentes en el Sistema.

1.1.4 Salida:

Se mostrará por pantalla todos los registros generados por la consulta.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón "Ver resultados" que ejecutará la funcionalidad si es UA o UO. Si el usuario es UP se le envía directamente a sus resultados.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF26 Ver resultados por fecha.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA o UO ver un informe de los exámenes según el paciente seleccionado en la aplicación.

1.1.2 Entradas:

- Fecha Inicial.
- Fecha Final.
- Número de paciente.

1.1.3 Procesos:

Se realizará una consulta en la BD con el número del paciente y un rango de fechas, obteniéndose todos los datos existentes en el Sistema.

1.1.4 Salida:

Se mostrará por pantalla todos los registros generados por la consulta.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón "Ver resultados por Fecha" que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF27 Ver resultados por grupo (RE).

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA o UO ver un informe de los exámenes según el paciente seleccionado en la aplicación.

1.1.2 Entradas:

- Número de paciente.

1.1.3 Procesos:

Se realizará la consulta con el dato proporcionado, obteniéndose todos los registros que existan en el Sistema.

1.1.4 Salida:

Se mostrará por pantalla todos los registros generados por la consulta.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón “Ver resultados por grupo” que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF28 Ver resultados grupo-fecha (RE).

1.1 Especificación:

1.1.1 Introducción:

Se permitirá al usuario UA o UO ver un informe de los exámenes según el paciente seleccionado en la aplicación.

1.1.2 Entradas:

- Número de paciente.
- Fecha inicial.
- Fecha Final.

1.1.3 Procesos:

Se realizará la consulta con los datos proporcionados, obteniéndose todos los registros que existan en el Sistema.

1.1.4 Salida:

Se mostrará por pantalla todos los registros generados por la consulta.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón “Ver resultados grupo-fecha” que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF29 Respaldar base de datos.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá únicamente a los usuarios de tipo UA crear un respaldo de los datos contenidos en la BD.

1.1.2 Entradas:

Ninguna.

1.1.3 Procesos:

Se realizará una consulta que tendrá todas las tablas con sus respectivos datos y luego le indicara al usuario el lugar en el cual desea almacenar el respaldo.

1.1.4 Salida:

Ventana de abrir/guardar del navegador.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón “Respaldar Base de Datos” que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF30 Restaurar base de datos.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá únicamente a los usuarios de tipo UA crear una Recuperación de la BD, mediante un archivo creado por la aplicación en curso.

1.1.2 Entradas:

- Nombre y localización del archivo de recuperación.

1.1.3 Procesos:

Se realizará una consulta que ejecutará el archivo de recuperación.

1.1.4 Salida:

Se mostrará por pantalla una serie de mensajes según los resultados de la consulta.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón "Examinar". Luego aparecerá una nueva ventana que permitirá localizar el archivo que posteriormente se realizará al pulsar el botón "Restaurar" para ejecutar la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF31 Insertar usuario.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá únicamente a los usuarios de tipo UA crear usuarios en la aplicación.

1.1.2 Entradas:

- Nombres.
- Apellidos.
- Nick.
- Clave.
- Tipo de usuario.

1.1.3 Procesos:

Se validarán los datos introducidos por el UA que serán validados e insertados en la BD.

1.1.4 Salida:

Se mostrará por pantalla todos los usuarios existentes.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón "Ingresar" que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF32 Editar usuario.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá únicamente a los usuarios de tipo UA editar usuarios en la aplicación.

1.1.2 Entradas:

- Número de usuario.
- Nombres.
- Apellidos.
- Nick.
- Clave.
- Tipo de usuario.

1.1.3 Procesos:

Mediante una consulta los datos serán presentados, verificados y luego actualizados en la BD.

1.1.4 Salida:

Se mostrará por pantalla todos los usuarios existentes.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón “Actualizar” que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF33 Borrar usuario.

1.1 Especificación:

1.1.1 Introducción:

Se permitirá únicamente a los usuarios de tipo UA eliminar usuarios en la aplicación.

1.1.2 Entradas:

- Número de usuario.

1.1.3 Procesos:

Mediante una consulta los datos serán presentados antes de eliminarse y luego borrados de la BD.

1.1.4 Salida:

Se actualizará la BD y se enviará a todos los usuarios.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón "Eliminar" que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

RF34 Salir del sistema.

1.1 Especificación:

1.1.1 Introducción:

Todos los usuarios registrados podrán acceder a esta funcionalidad.

1.1.2 Entradas:

No hay ninguna.

1.1.3 Procesos:

Se procederá a salir la aplicación.

1.1.4 Salida:

El usuario será redireccionado a la página de inicio.

1.2 Interfaces Externas:

Interfaz de Usuario:

El usuario pulsará un botón “Desconectarse” que ejecutará la funcionalidad.

Interfaz de Hardware:

Se utilizará cualquier ordenador que tenga acceso al servidor donde esté instalado el Sistema.

Interfaz de Software:

El proceso interactúa con la BD en la cual se encuentra almacenada toda la información.

Interfaz de Comunicación:

Se comunicará únicamente con el servidor de BD.

VIII.CASOS DE USO

8.1 Usuario Administrador

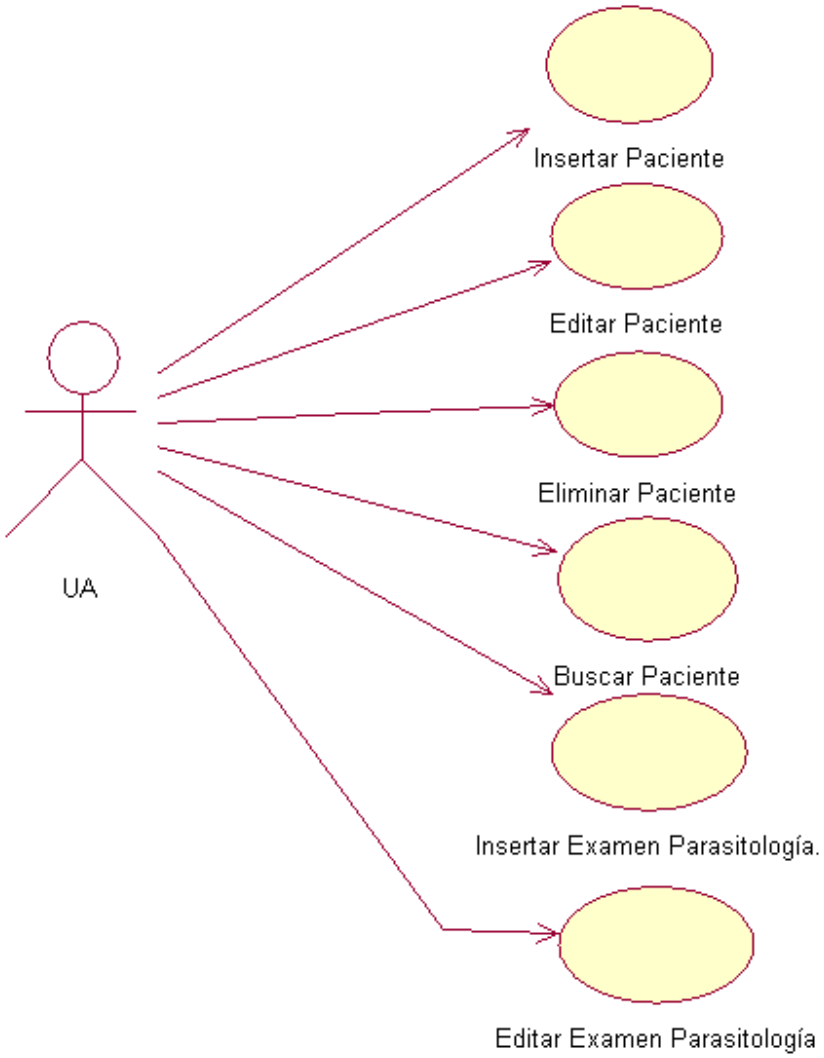


Figura 4: Usuario Administrador

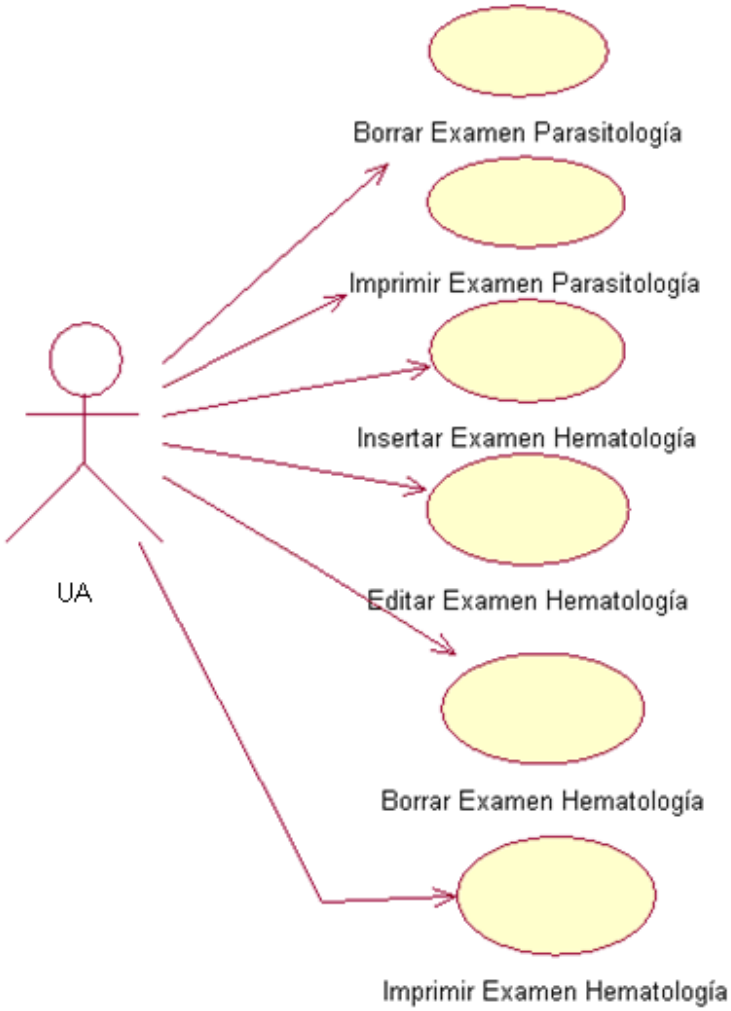


Figura 5: Usuario Administrador (Continuación)

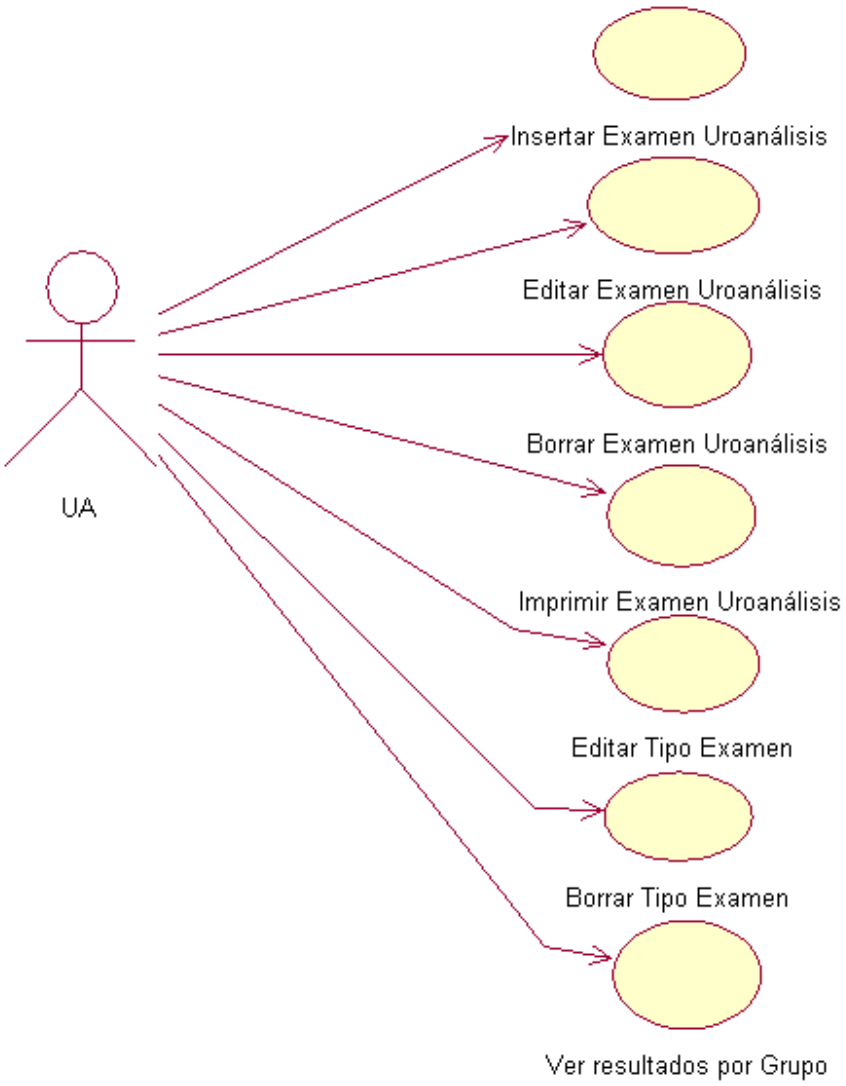


Figura 6: Usuario Administrador (Continuación)

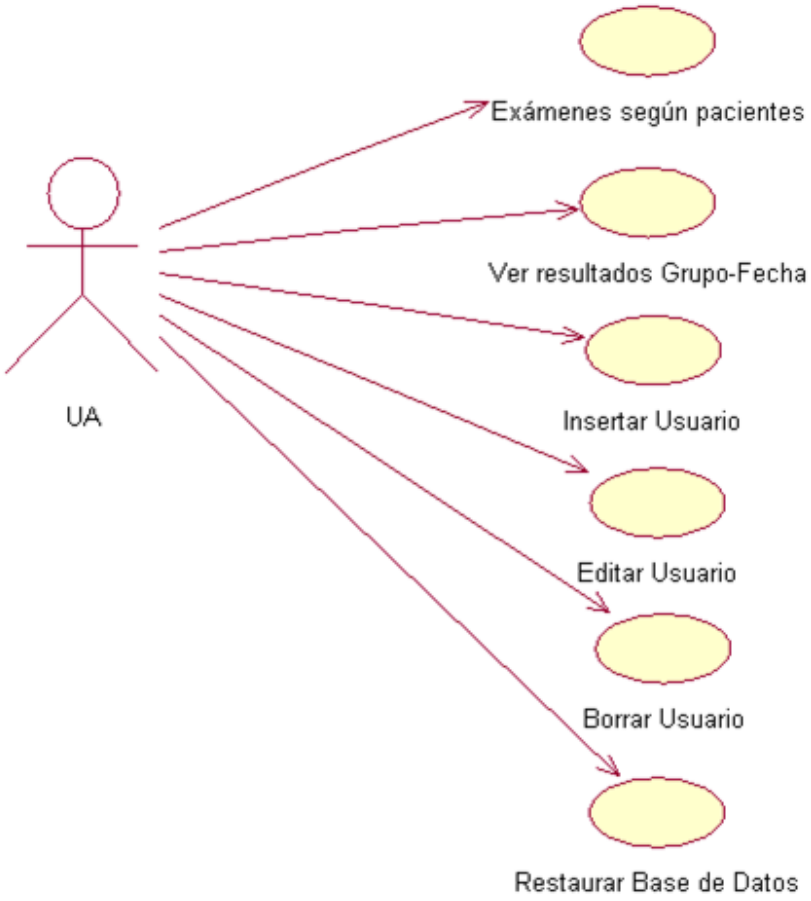


Figura 7: Usuario Administrador (Continuación)

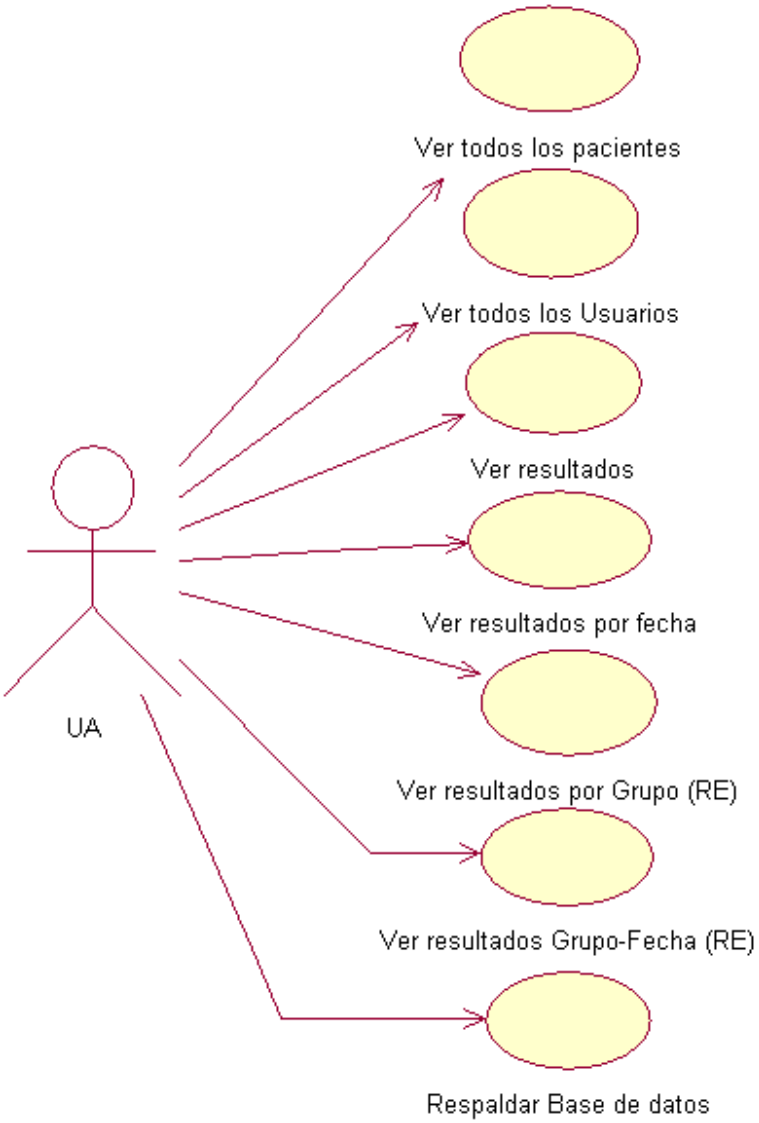


Figura 8: Usuario Administrador (Continuación)

8.2 Usuario Operador

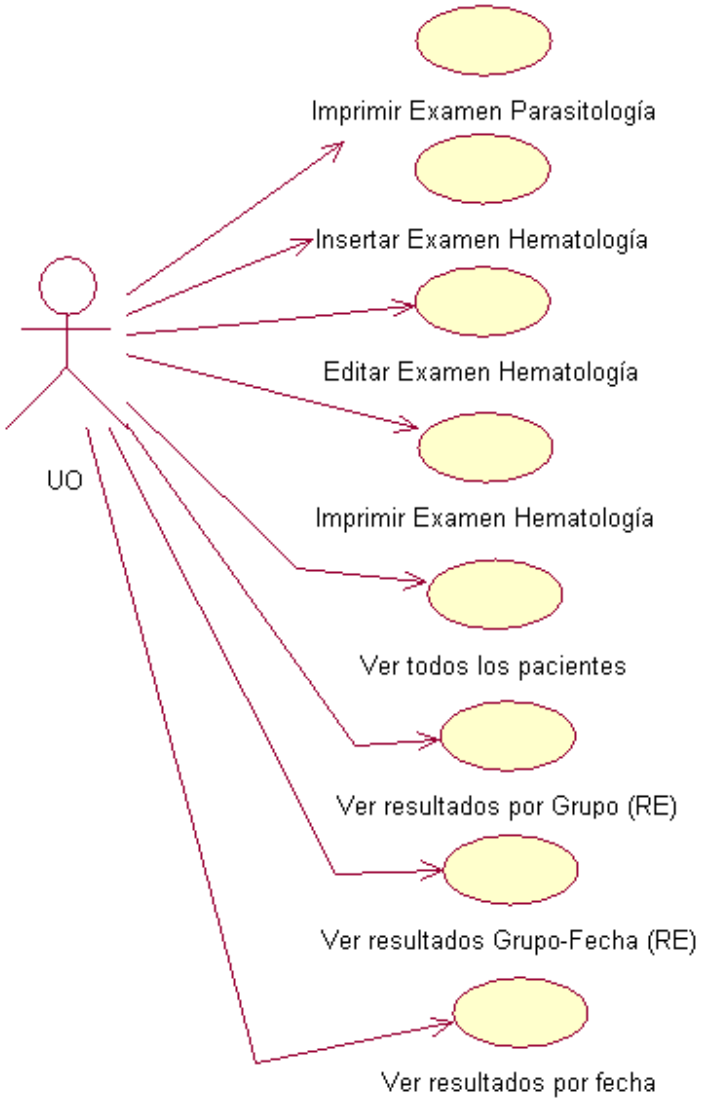


Figura 9: Usuario Operador

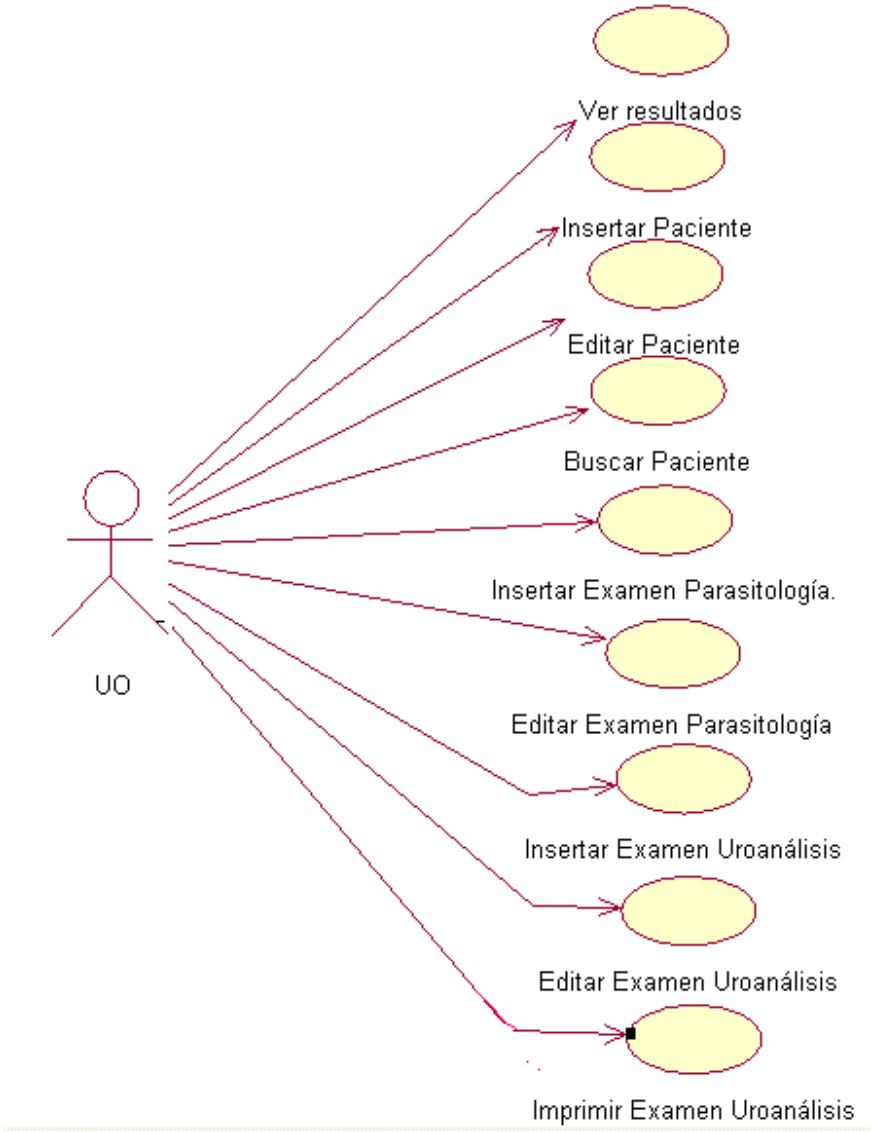


Figura 10: Usuario Operador (continuación)

8.3 Usuario Paciente

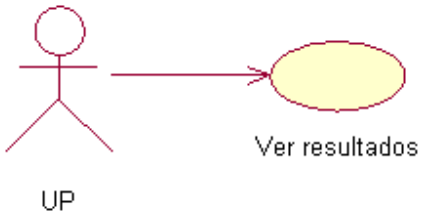


Figura 11: Usuario Paciente

IX. ROLES DE LOS USUARIOS

	Paciente	Examen	Usuario	Tipo de Examen	Base de Datos
Administrador	➢ Insertar. ➢ Editar. ➢ Eliminar. ➢ Buscar. ➢ Ver Pacientes. ➢ Resultados. ➢ Resultados por grupo (RE). ➢ Resultados por fecha. ➢ Resultados grupo-fecha.	➢ Insertar. ➢ Editar. ➢ Eliminar. ➢ Imprimir. ➢ Resultados por grupo. ➢ Resultados grupo-fecha ➢ Exámenes según pacientes.	➢ Insertar ➢ Editar. ➢ Borrar. ➢ Ver.	➢ Editar. ➢ Eliminar.	➢ Respaldar ➢ Restaurar.
Operador	➢ Insertar. ➢ Editar. ➢ Buscar. ➢ Ver Pacientes. ➢ Resultados. ➢ Resultados por grupo (RE). ➢ Resultados por fecha. ➢ Resultados grupo-fecha (RE).	➢ Insertar. ➢ Editar. ➢ Imprimir.			
Paciente	➢ Resultados.				

Figura 12: Roles de los usuarios

- ✓ El Usuario Administrador es capaz de realizar todas las tareas señaladas anteriormente.
- ✓ El Usuario Operador solamente realiza las tareas indicadas en la tabla.
- ✓ El Usuario Paciente solamente realiza la tarea señalada.

X. DIAGRAMA DE CLASES

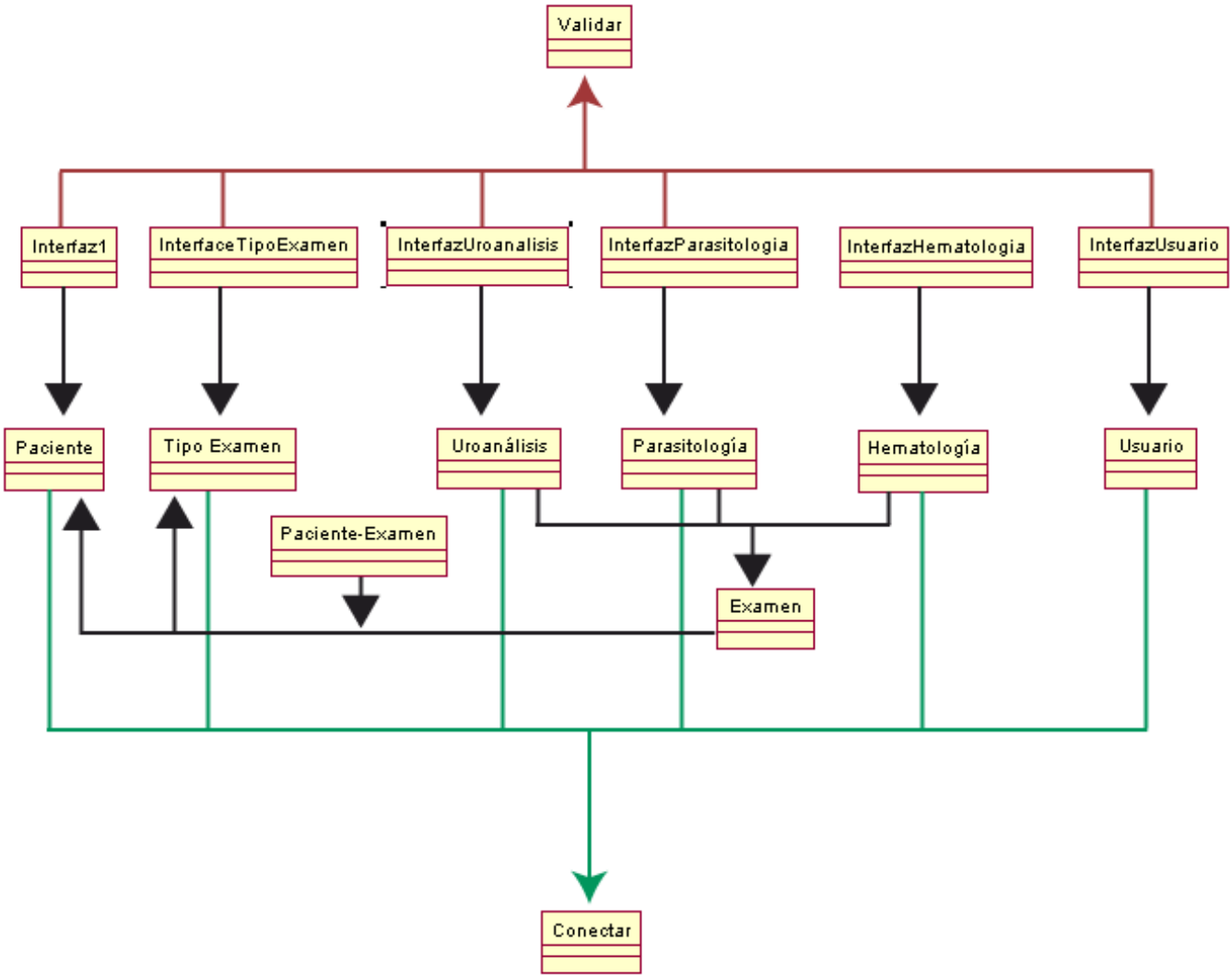
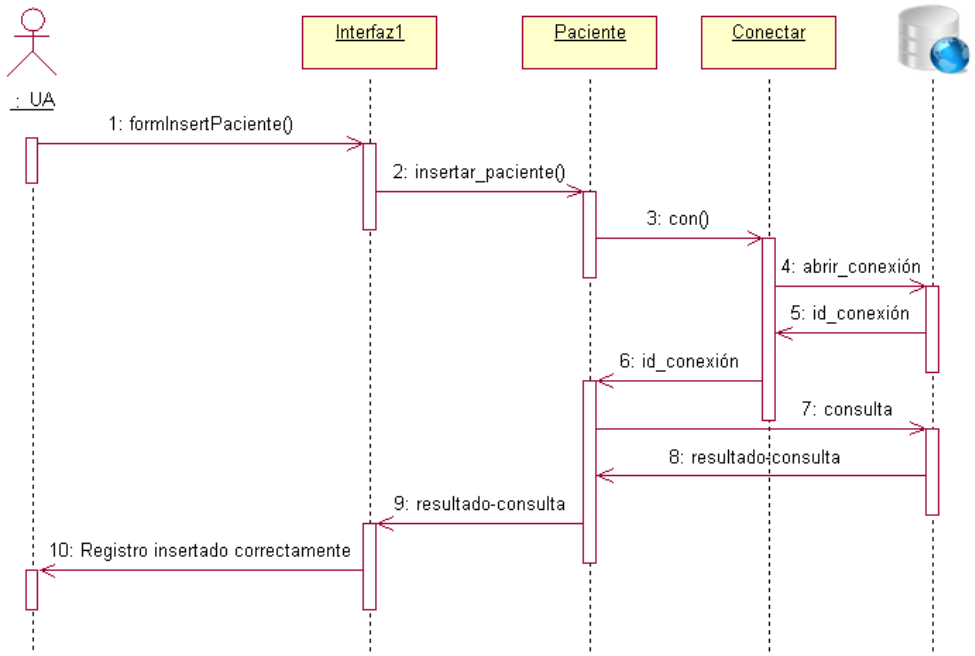


Figura 13: Diagrama de clases

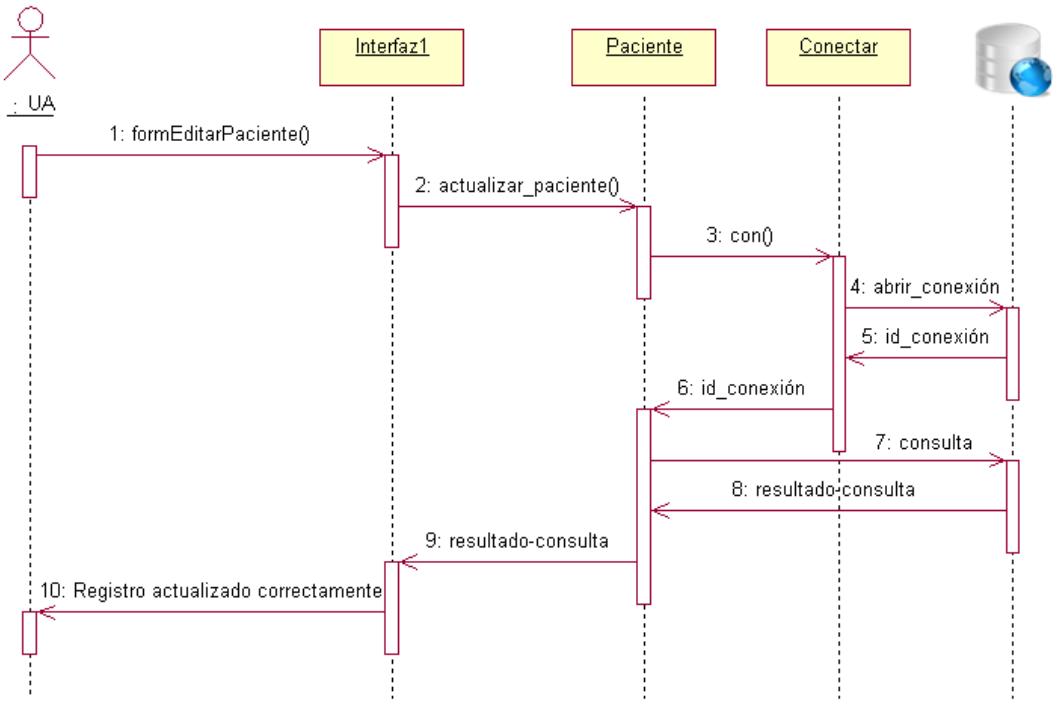
XI. DIAGRAMAS DE SECUENCIA

PACIENTE

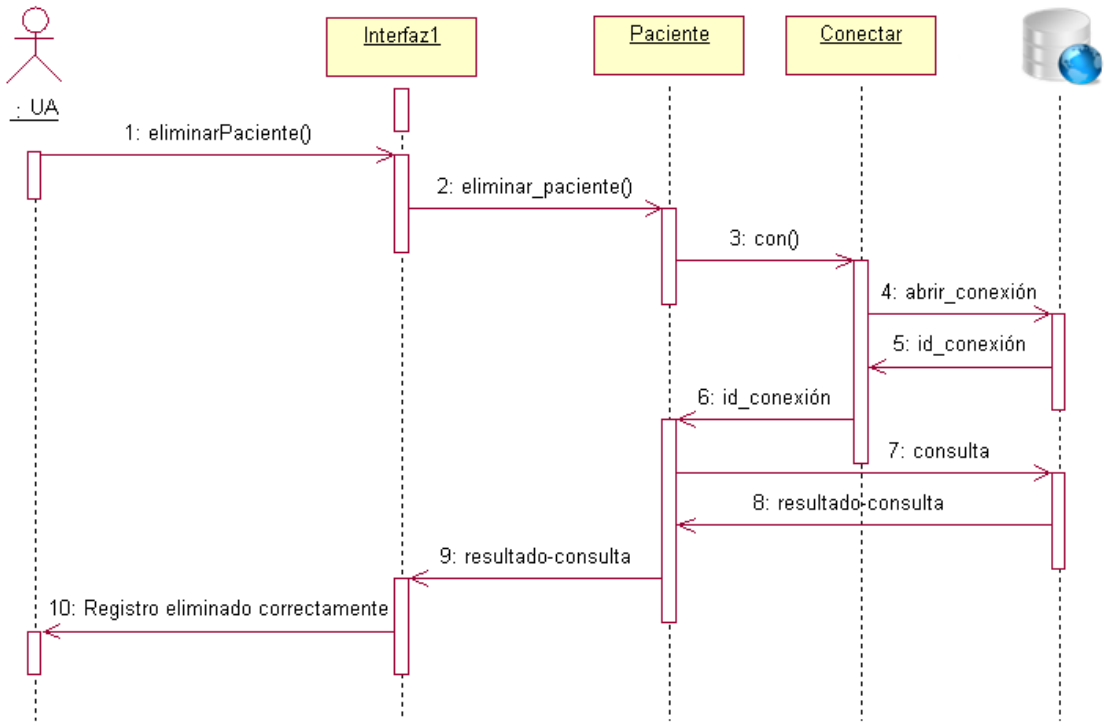
Insertar paciente



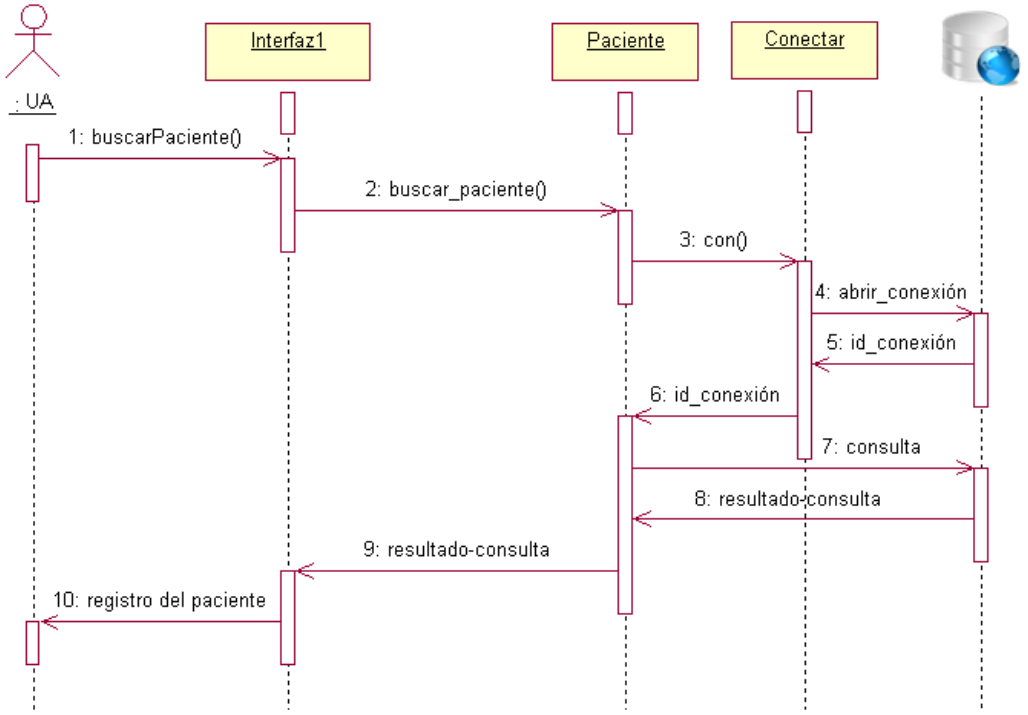
Editar paciente



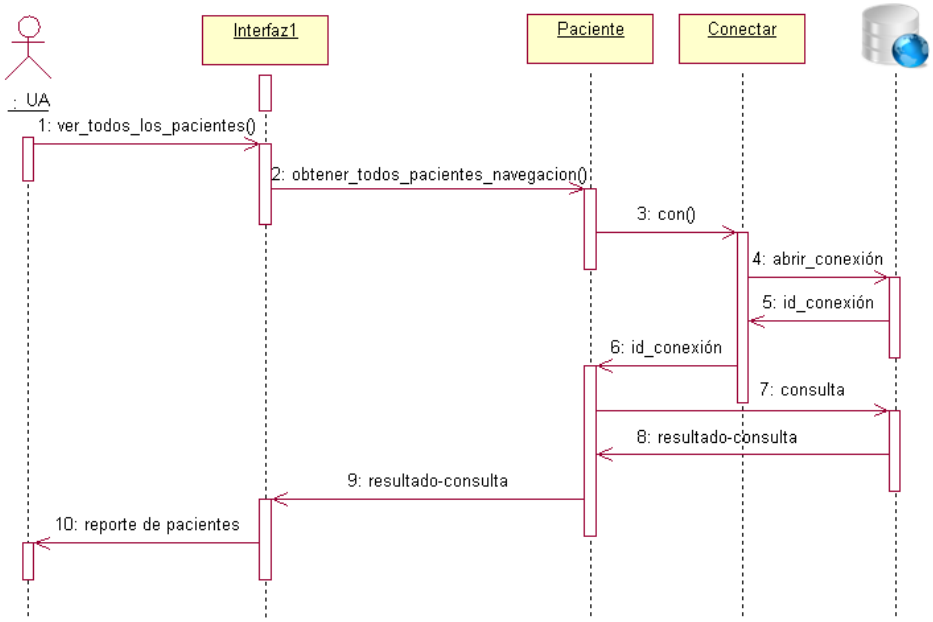
Eliminar paciente



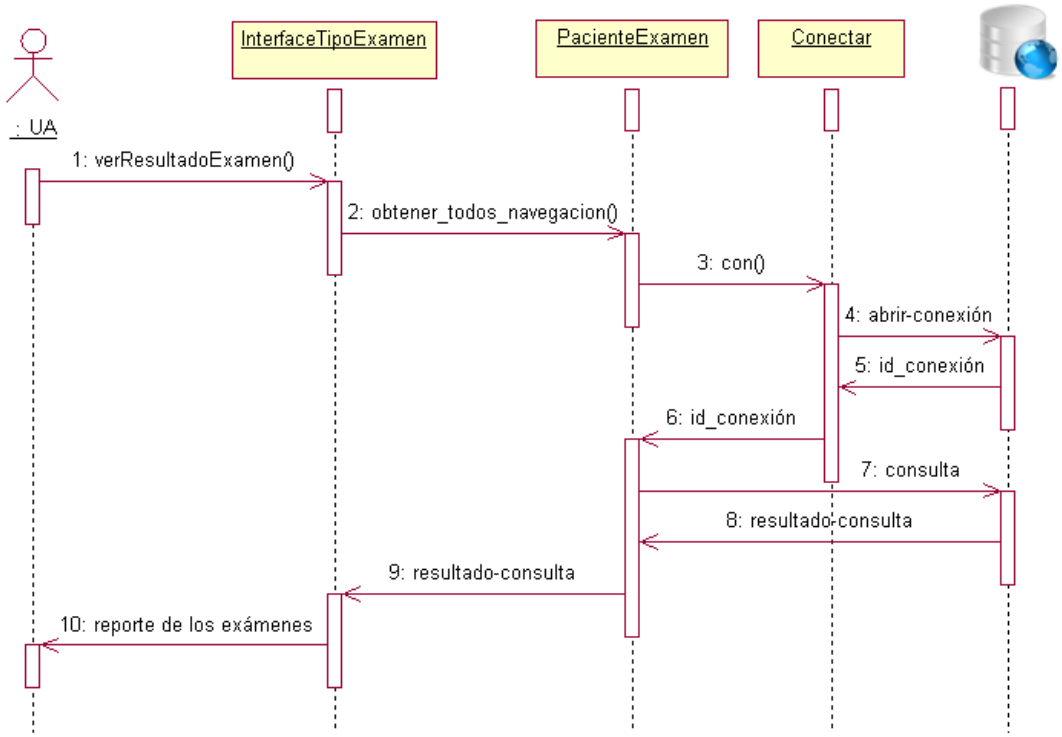
Buscar paciente



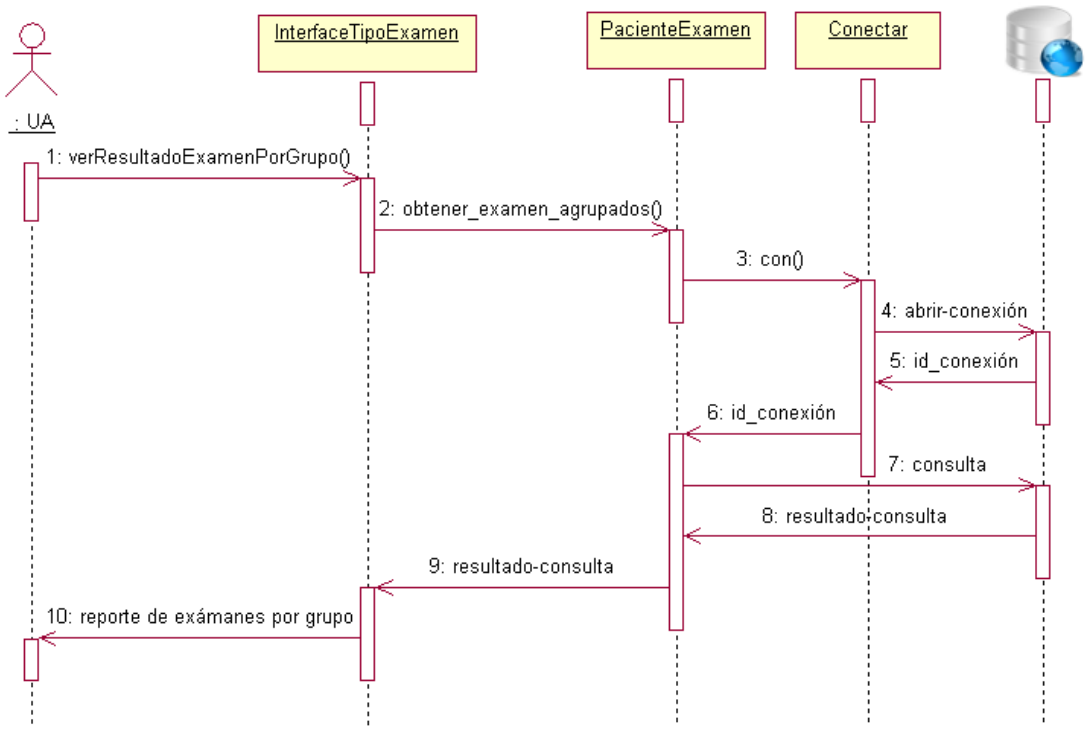
Ver todos los pacientes



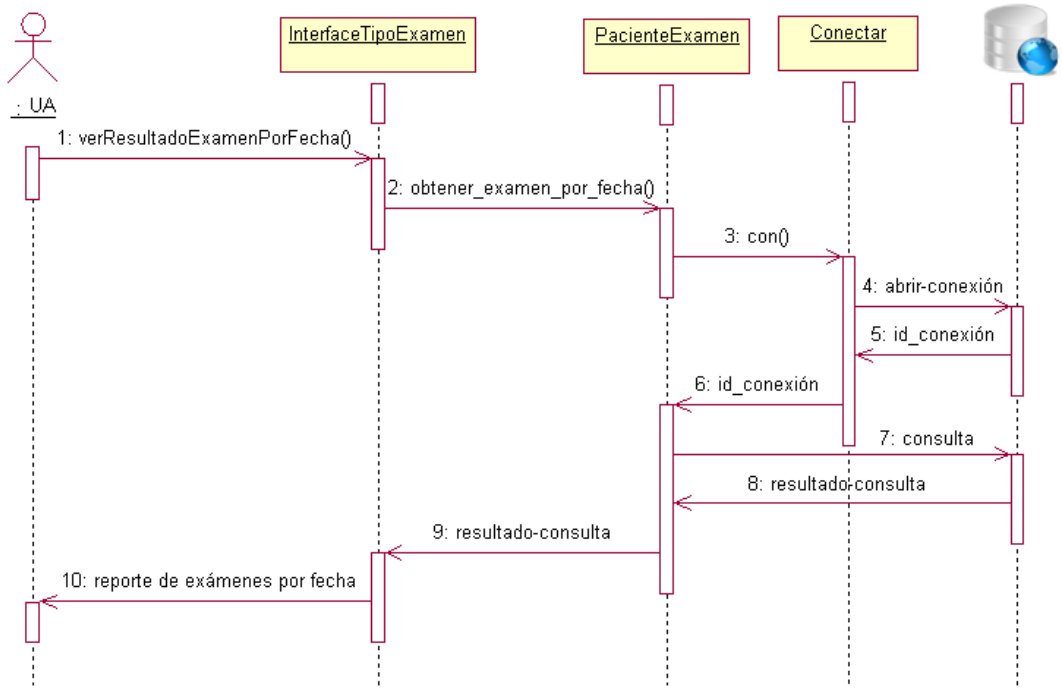
Ver resultados



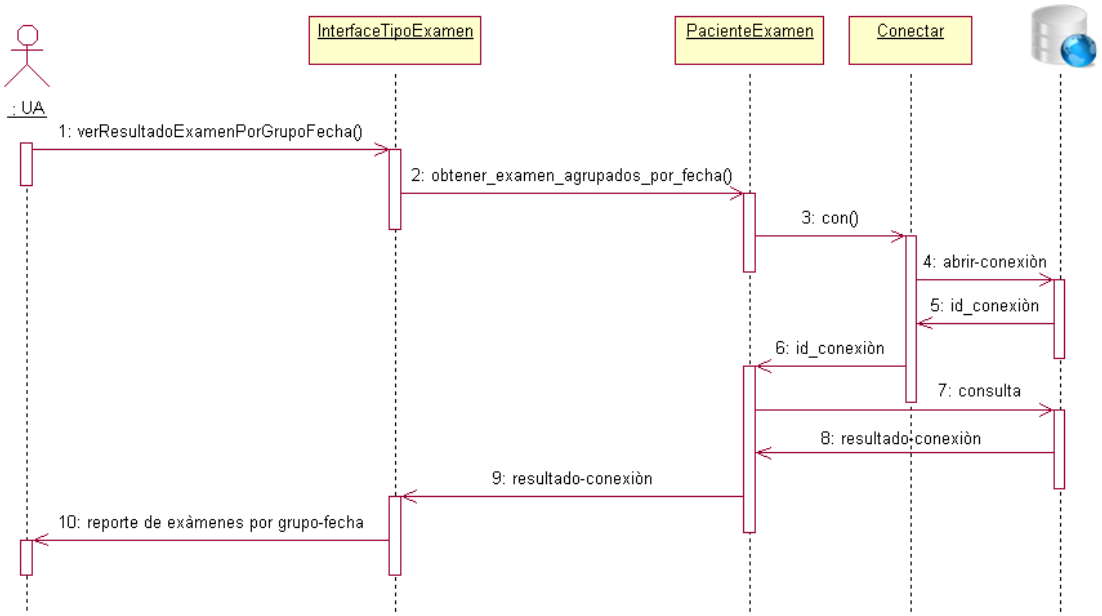
Ver resultados por grupo (RE)



Ver resultados por fecha

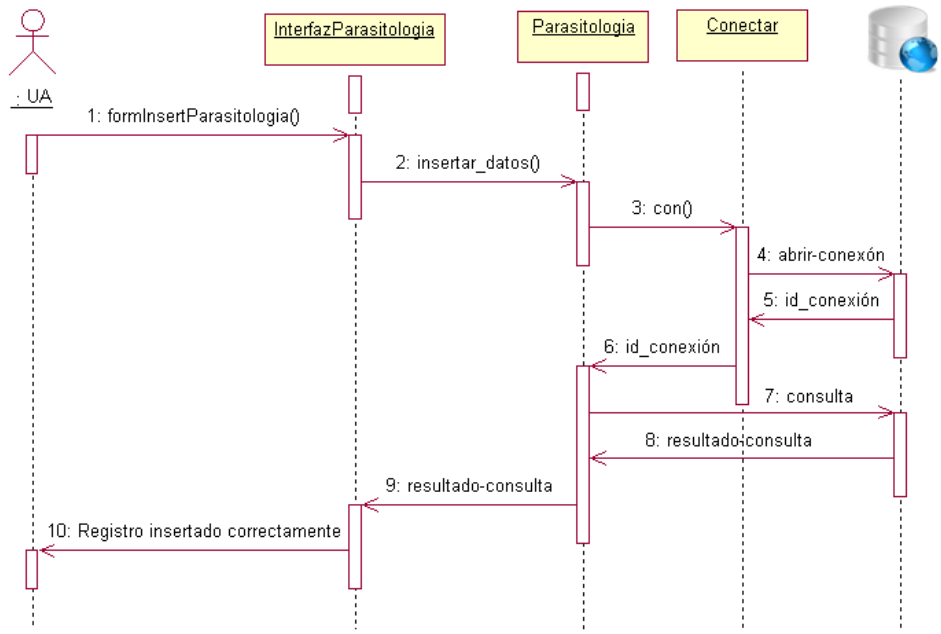


Ver resultados por grupo-fecha (RE)

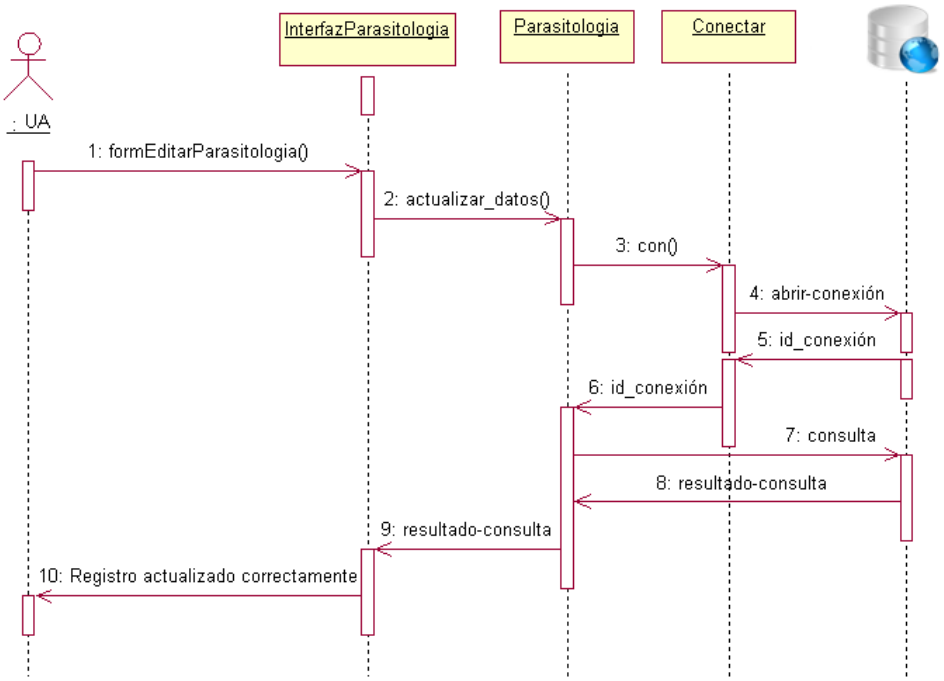


EXAMEN

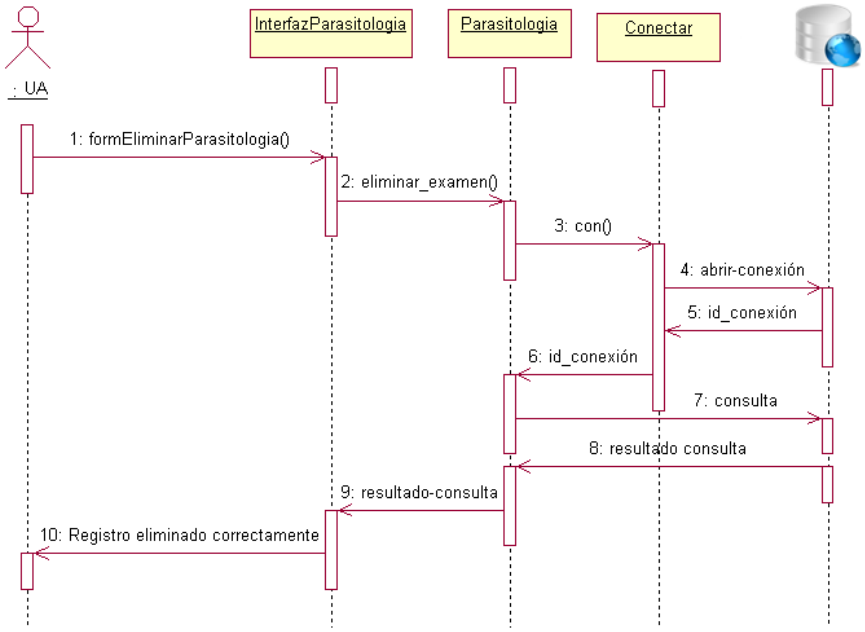
Insertar examen de parasitología



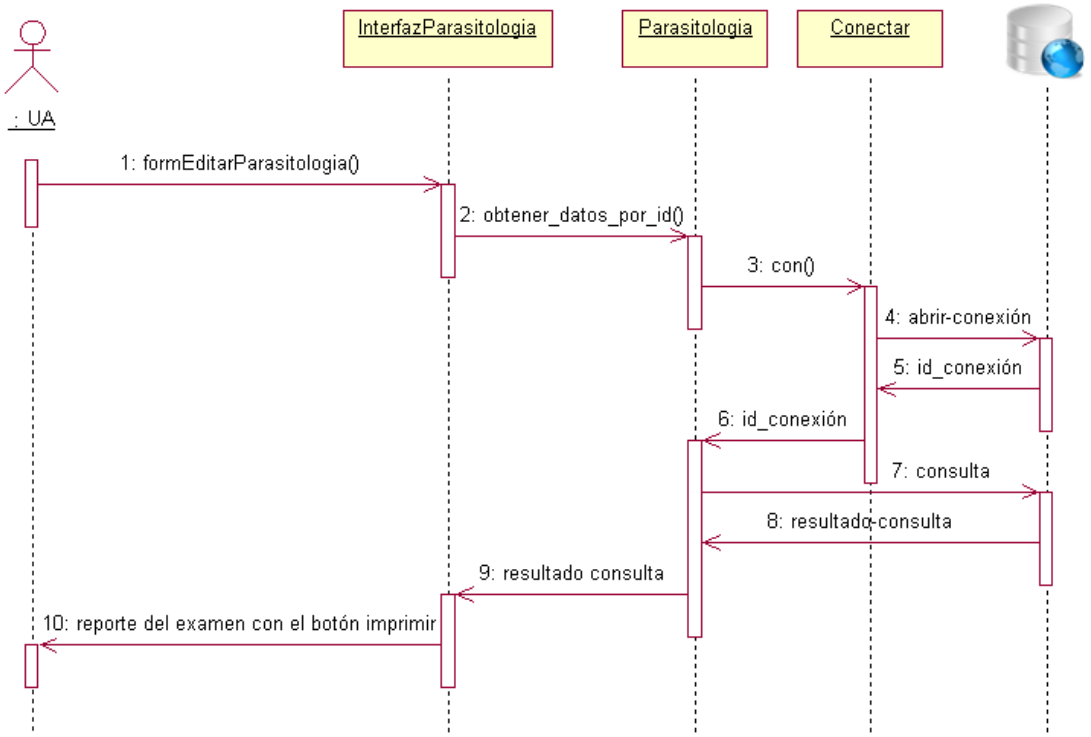
Editar examen de parasitología



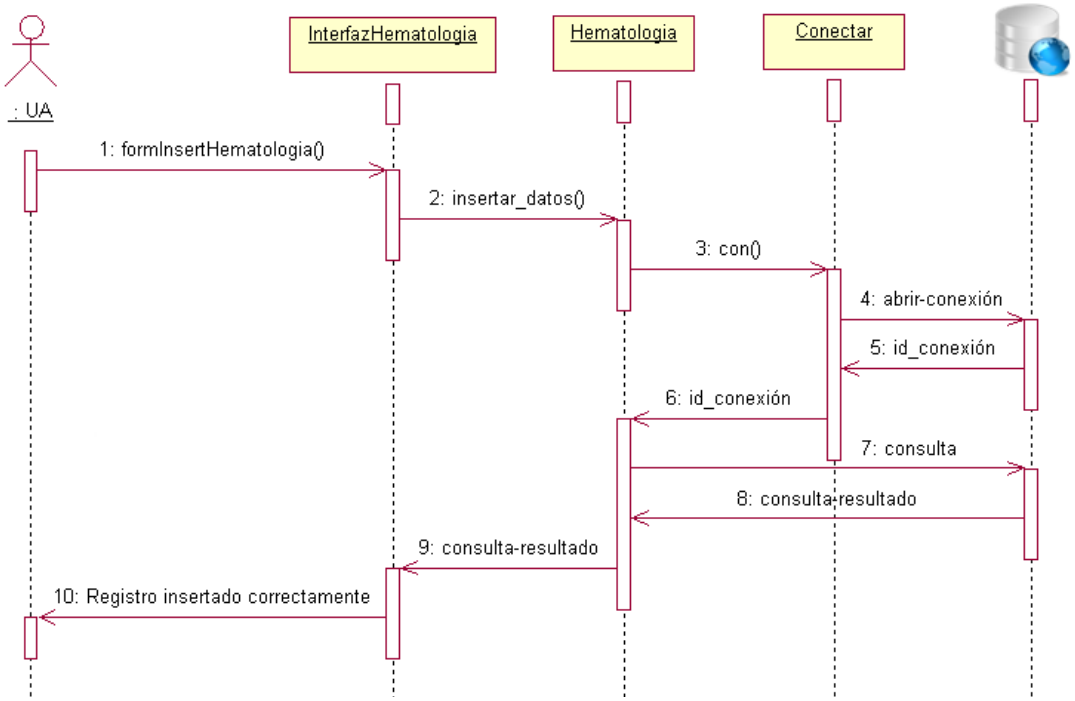
Eliminar examen de parasitología



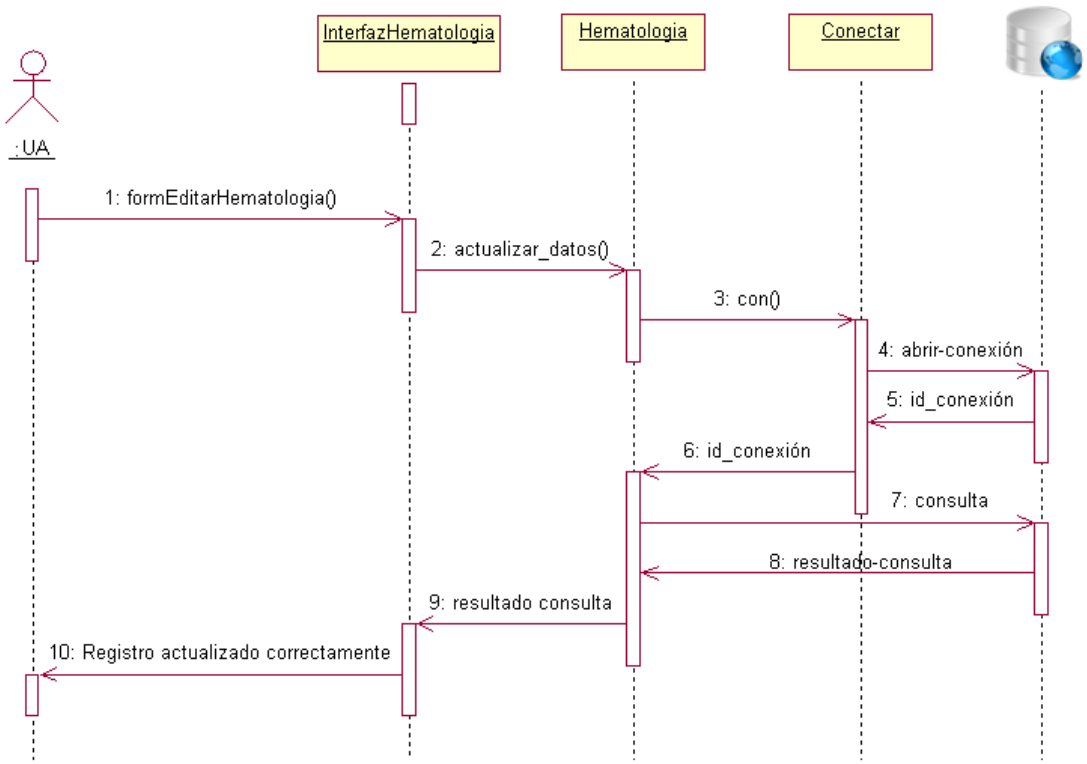
Imprimir examen de parasitología



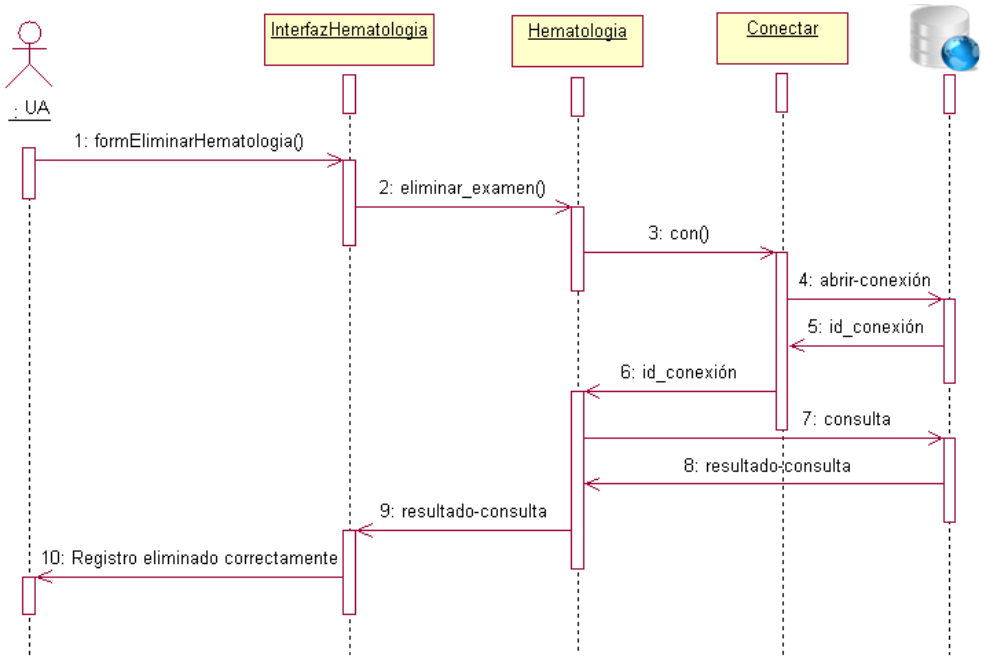
Insertar examen de hematología



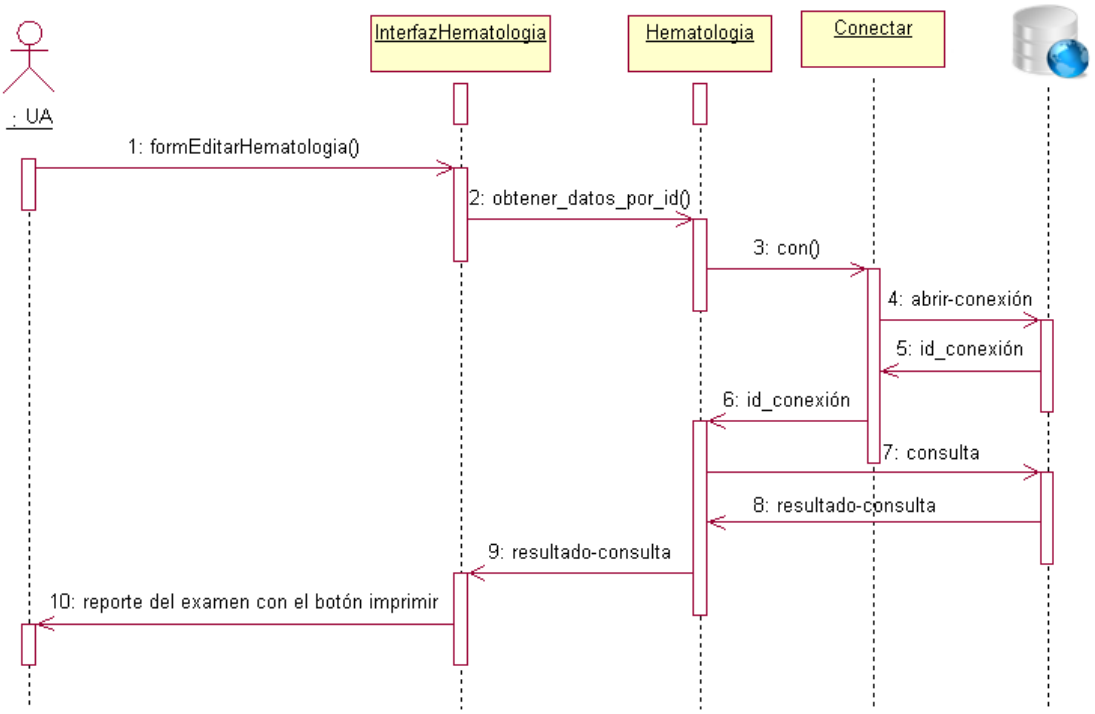
Editar examen de hematología



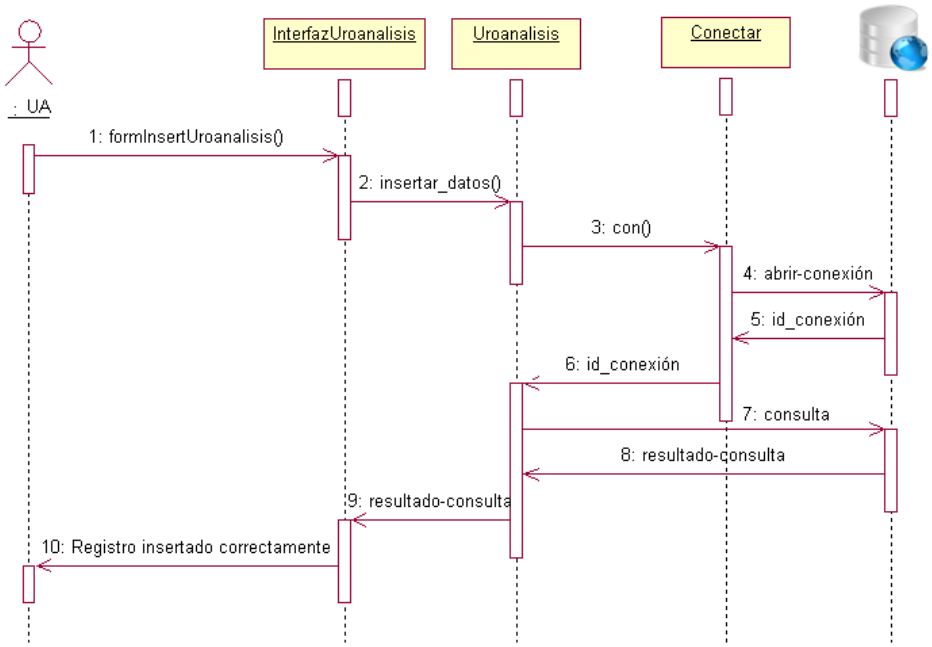
Eliminar examen de hematología



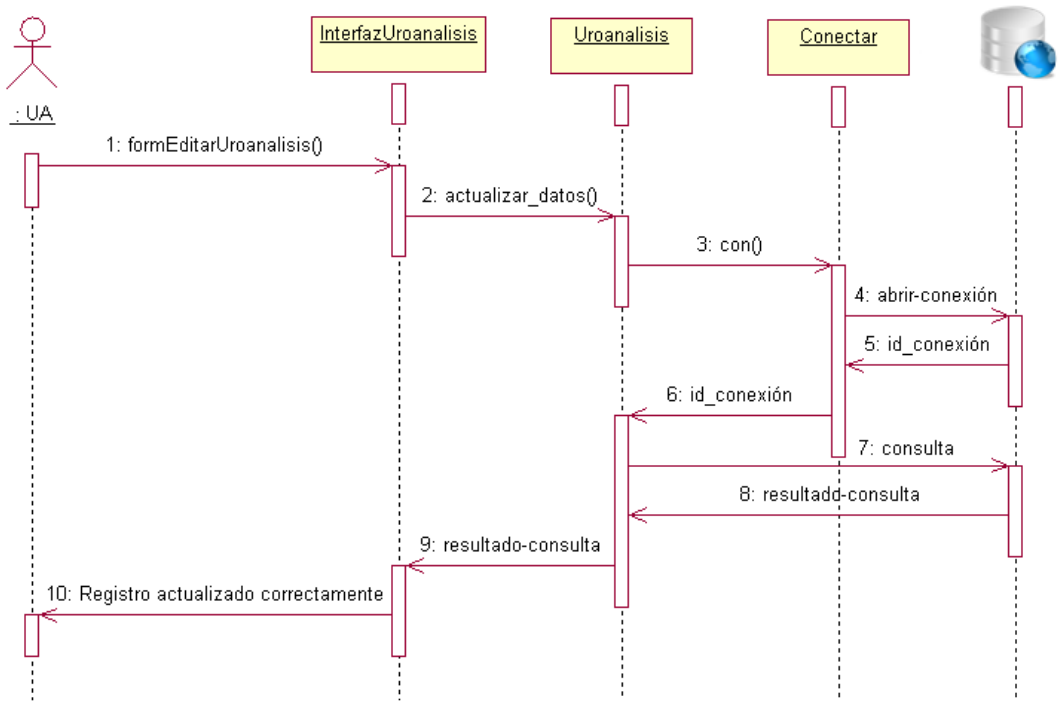
Imprimir examen de hematología



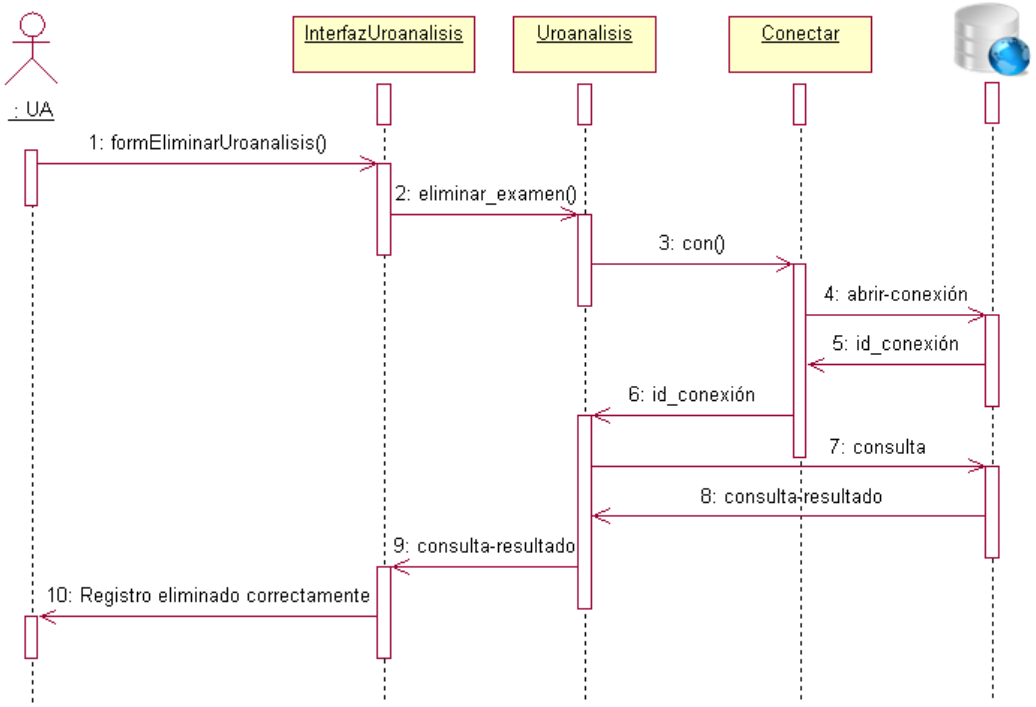
Insertar examen de uroanálisis



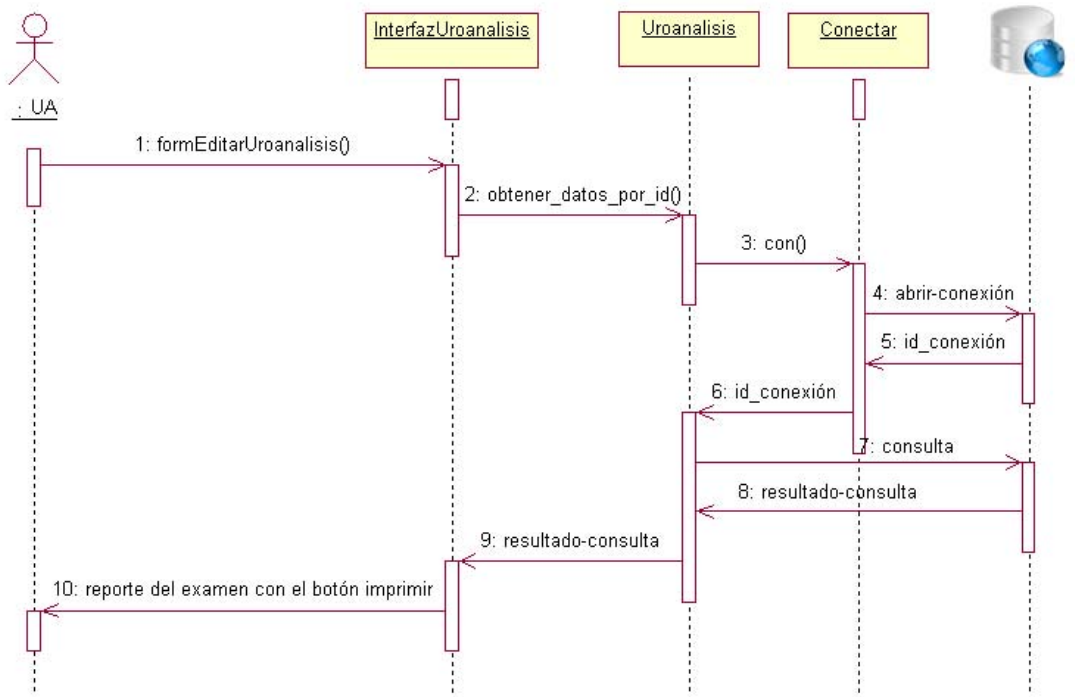
Editar examen de uroanálisis



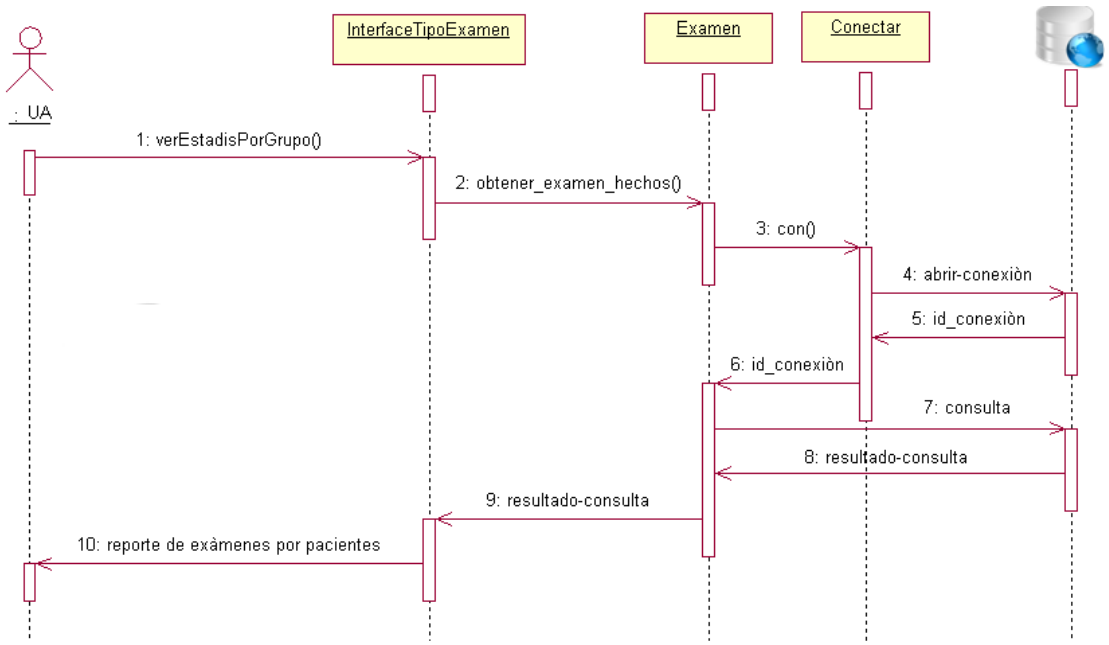
Eliminar examen de uroanálisis



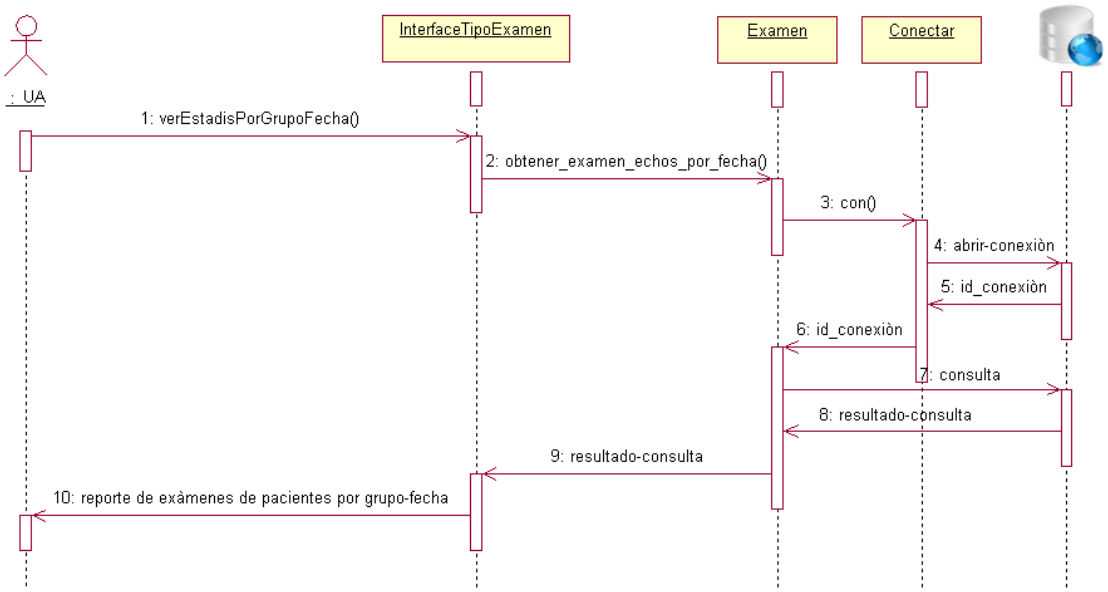
Imprimir examen de uroanálisis



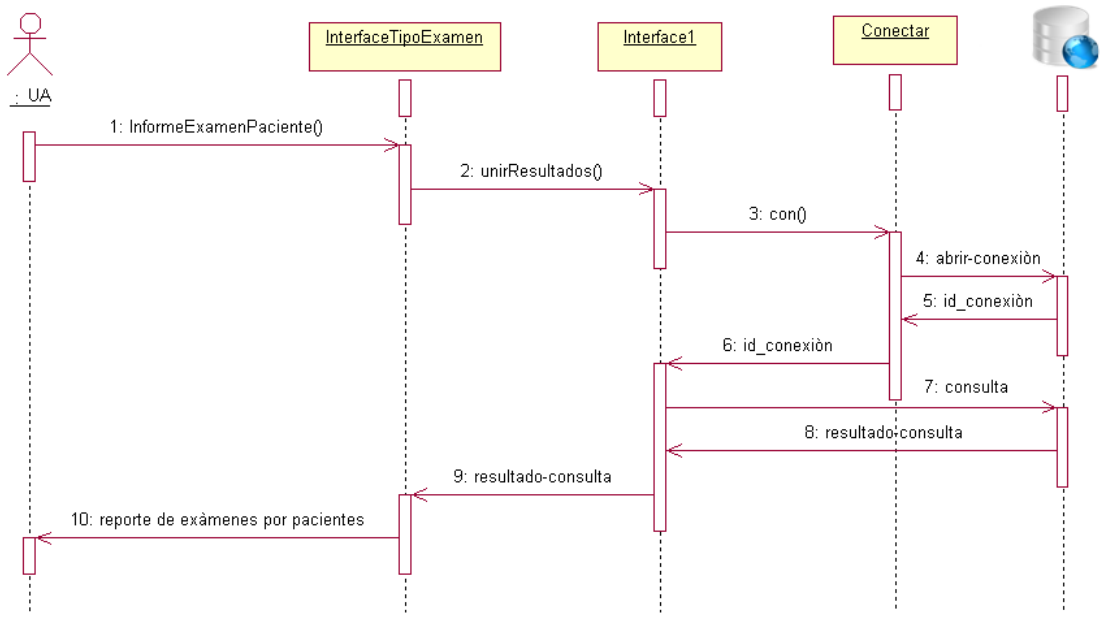
Ver resultado por grupo



Ver resultado por grupo-fecha

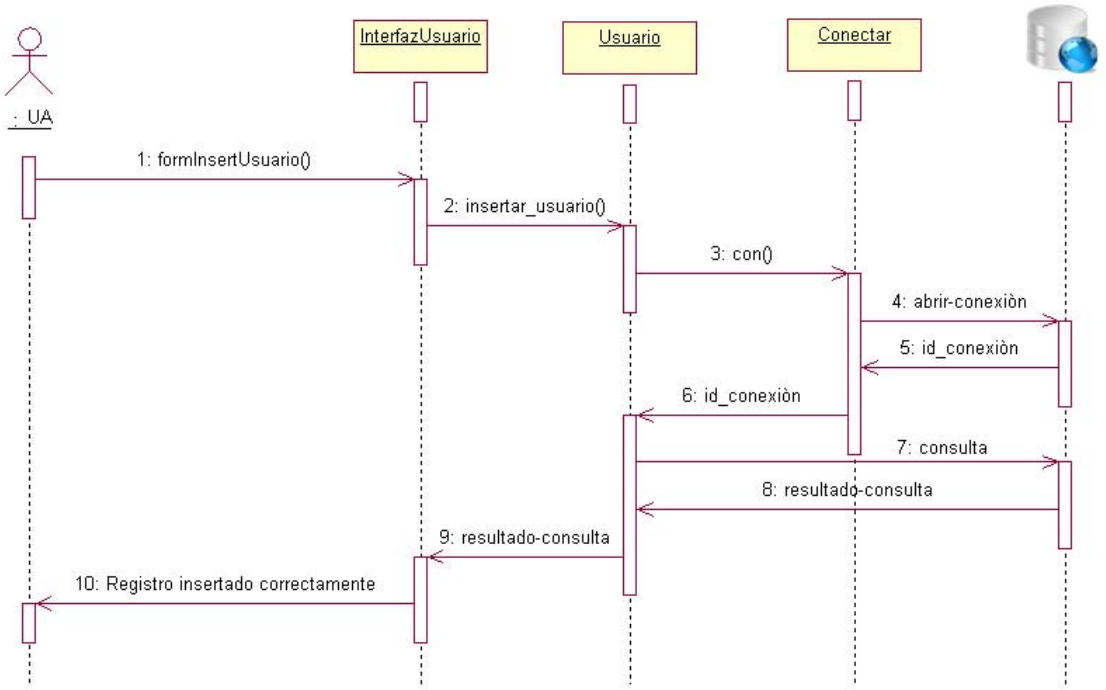


Exámenes según pacientes

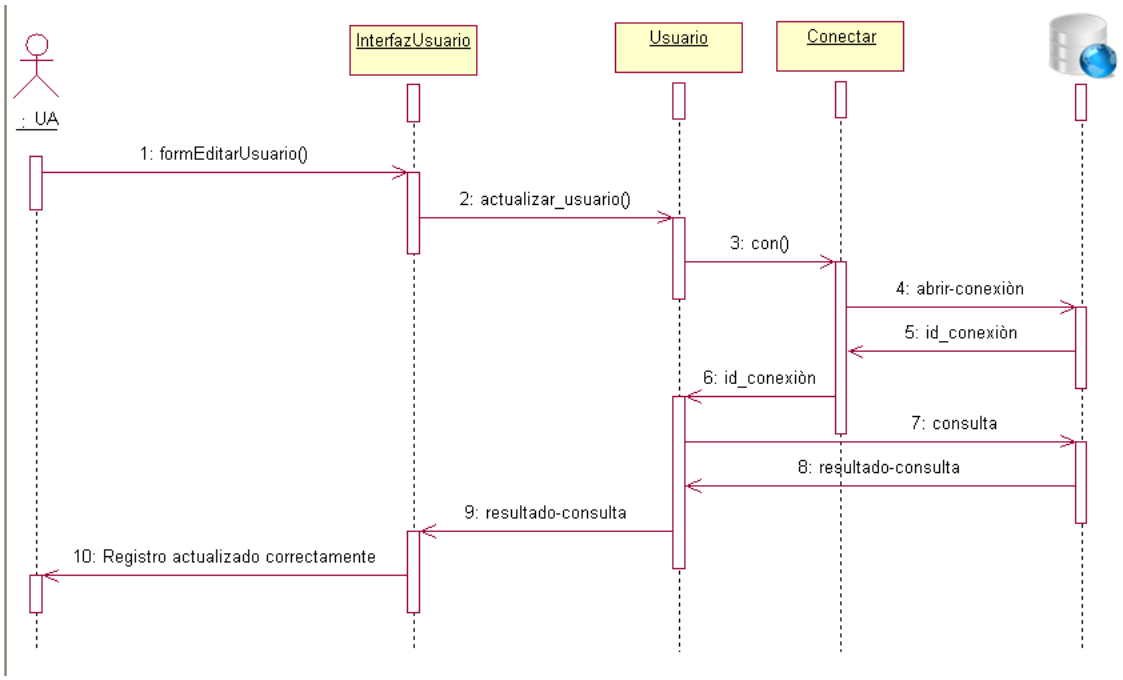


USUARIO

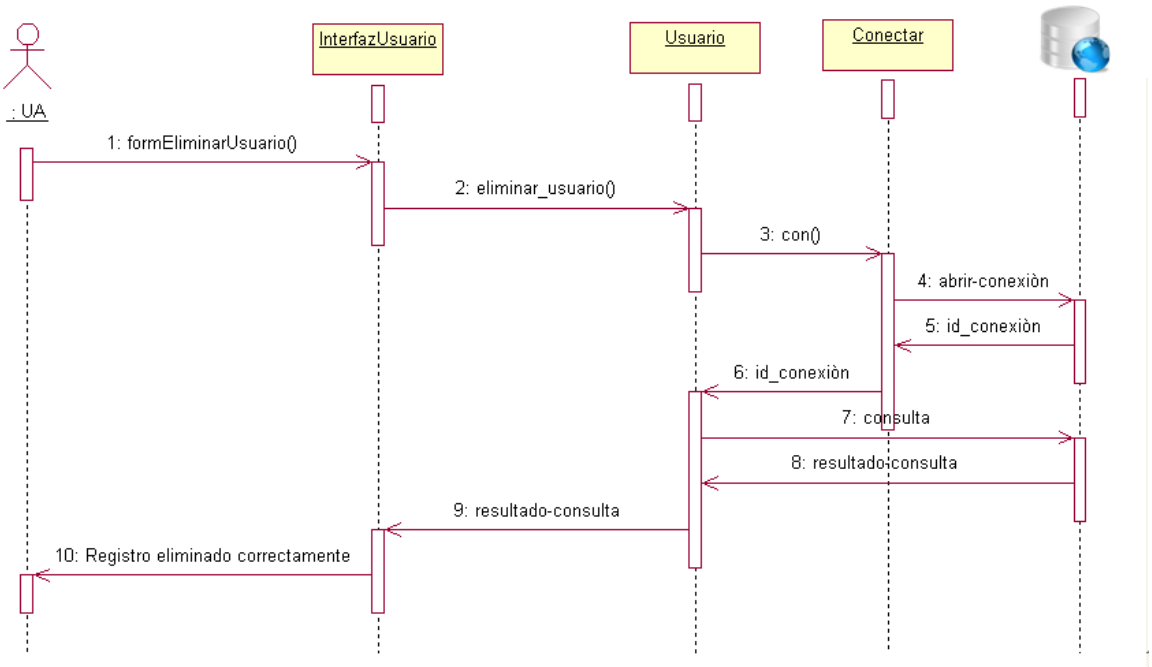
Insertar usuario



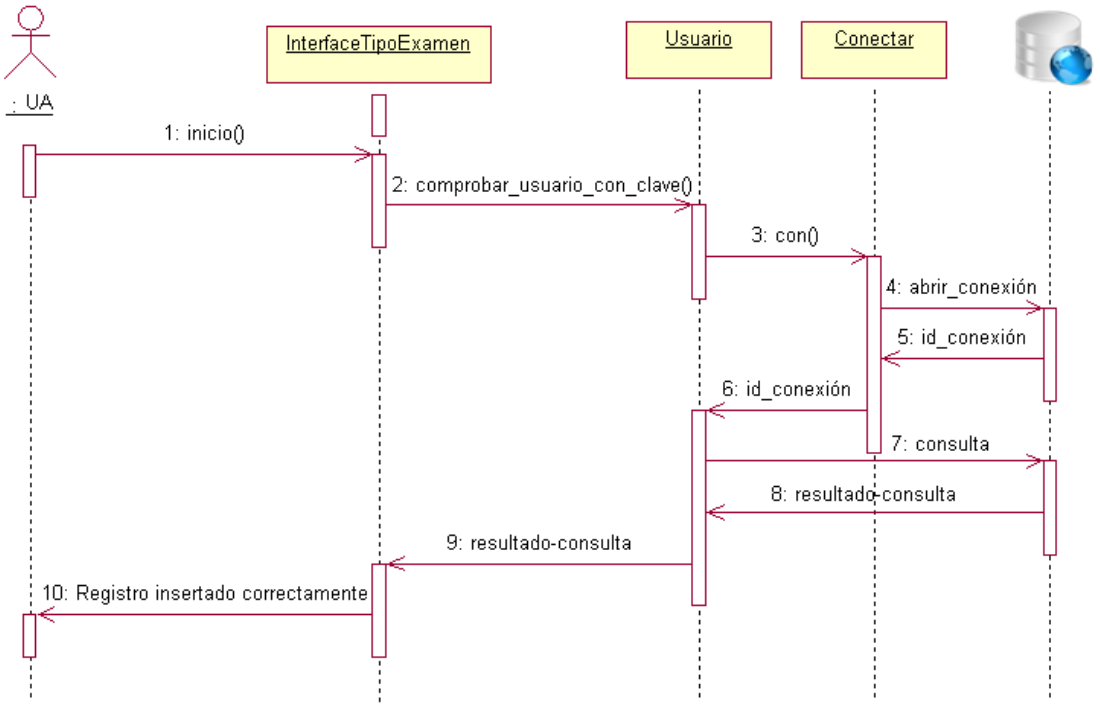
Editar usuario



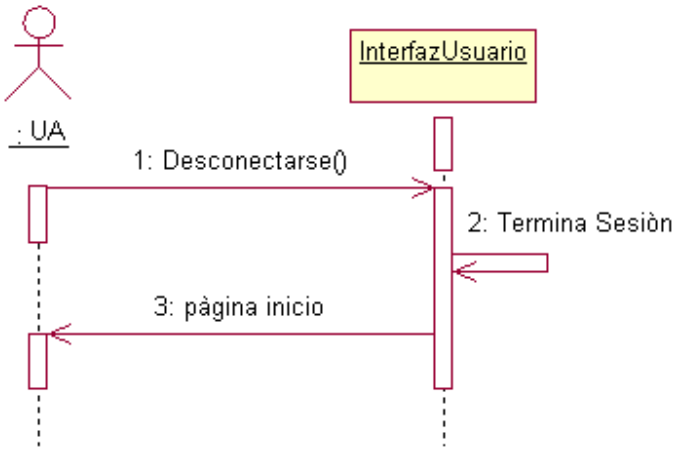
Eliminar usuario



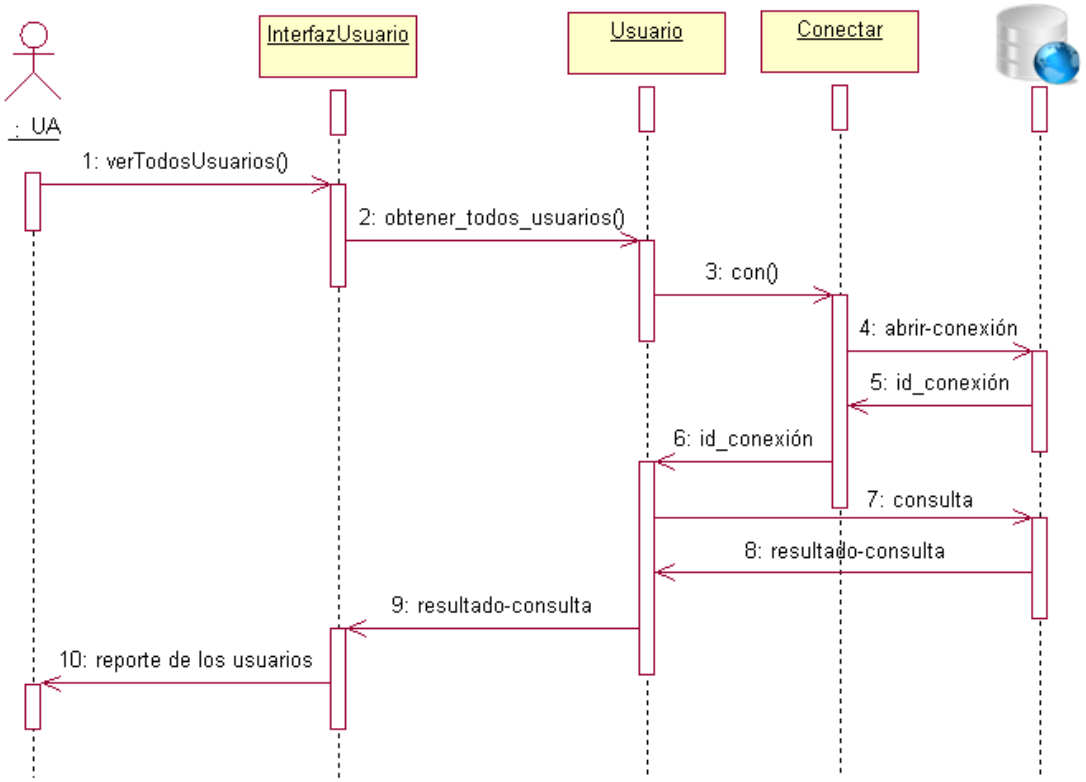
Ingresar a la aplicación



Desconectarse

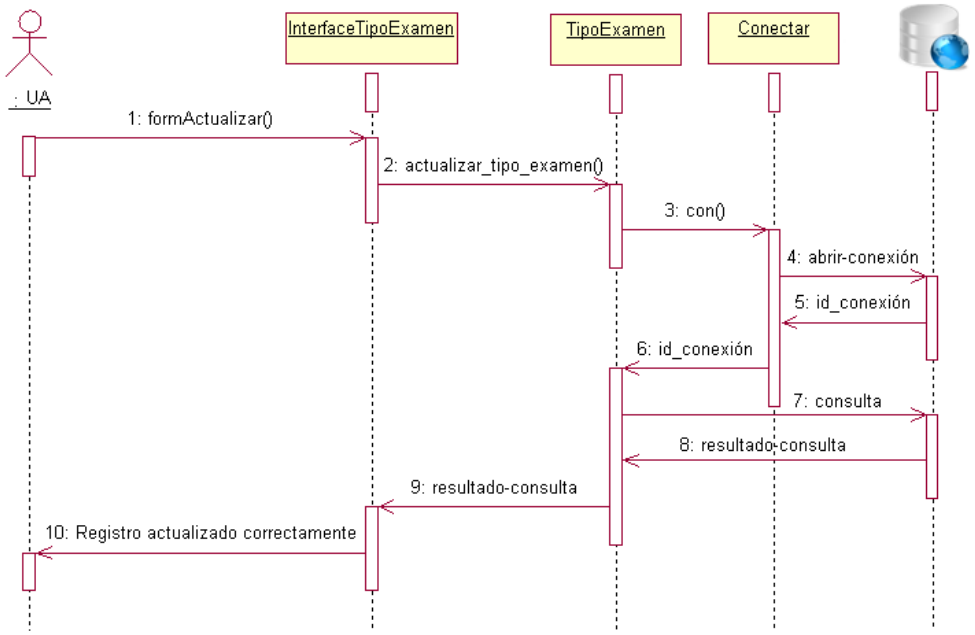


Ver todos los usuarios

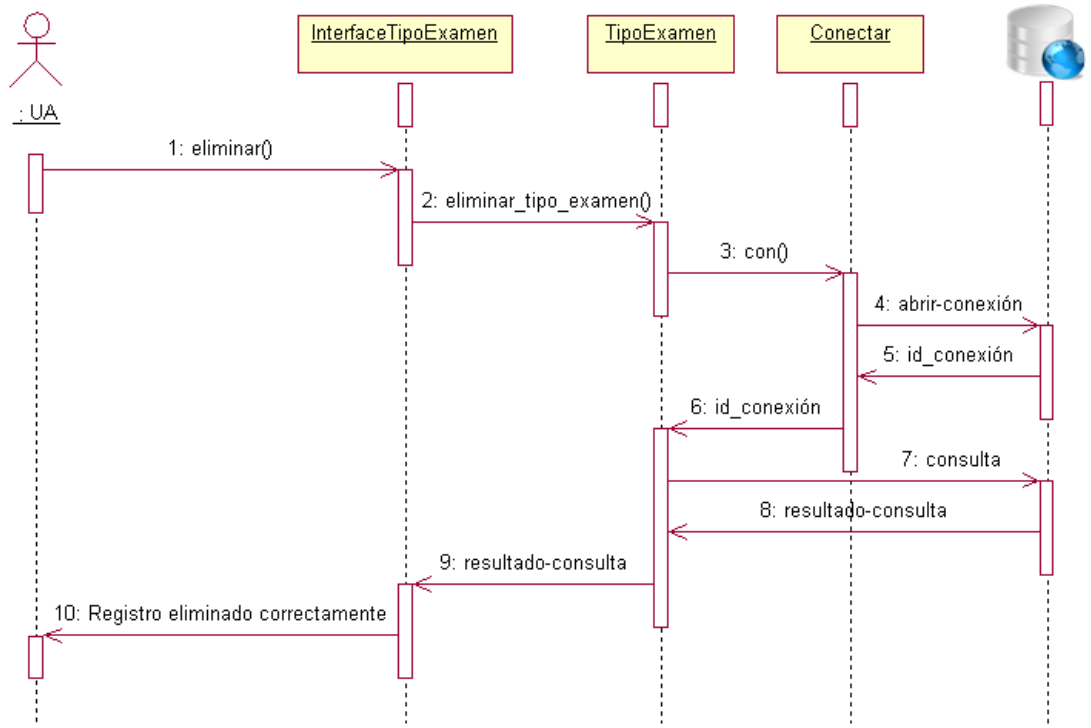


TIPO DE EXAMEN

Editar tipo de examen

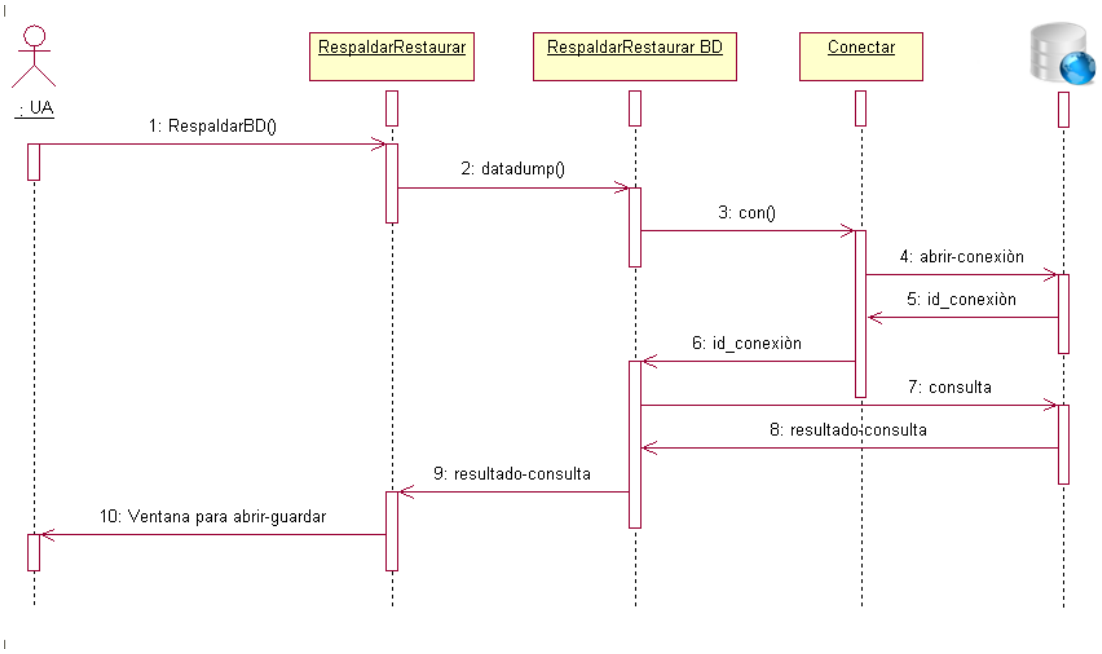


Eliminar tipo de examen

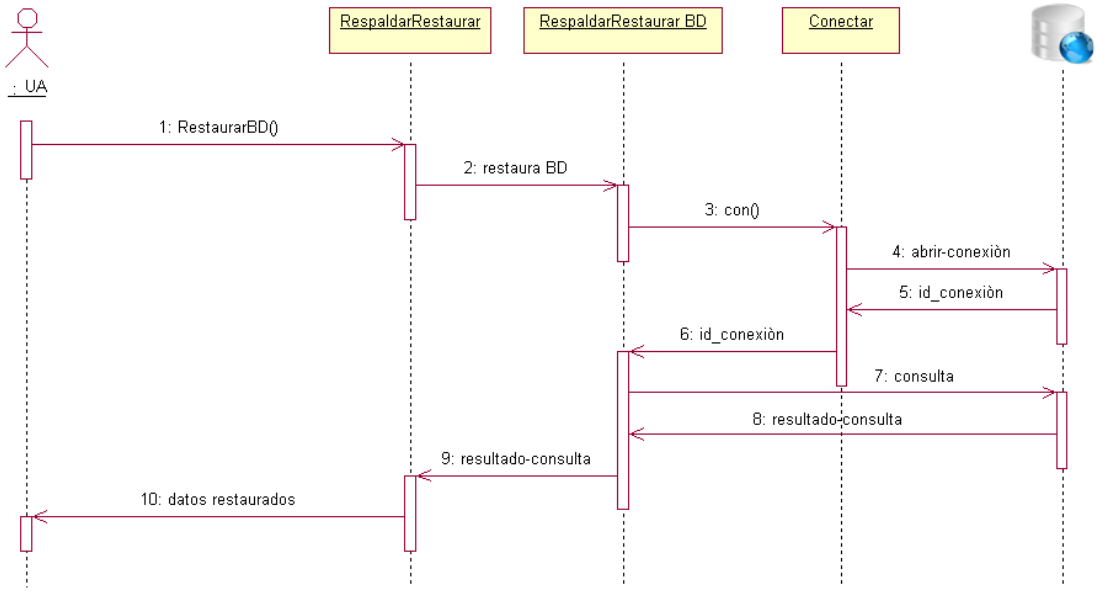


RESPALDAR-RESTAURAR

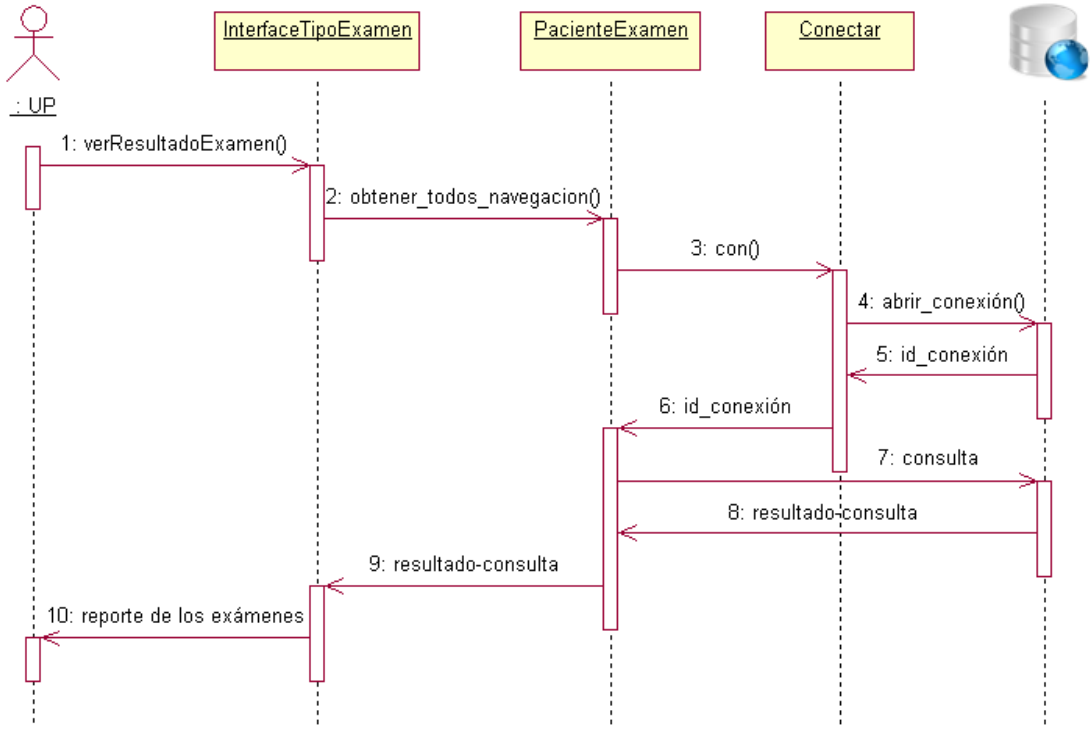
Respaldar base de datos



Restaurar base de datos



Ver resultados de paciente logueado



XII. DIAGRAMA RELACIONAL

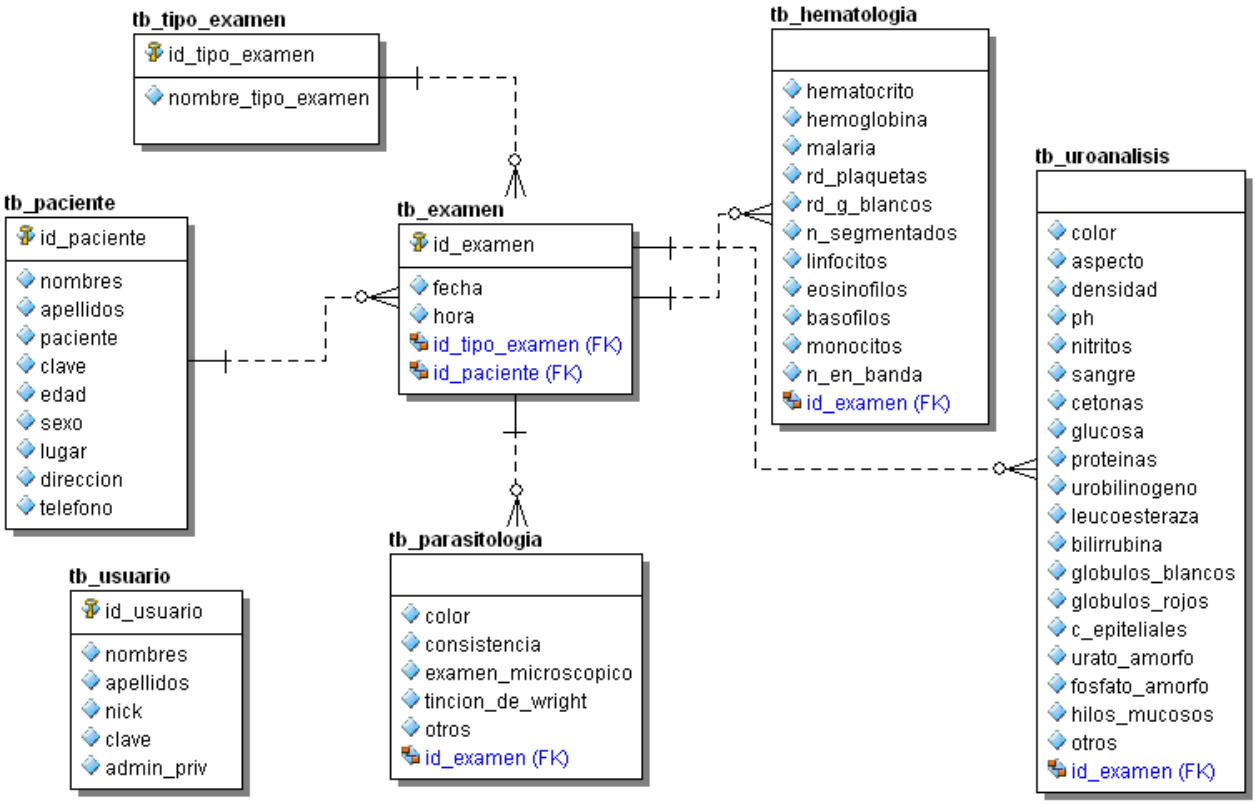


Figura 14: Diagrama relacional

XIII. DISEÑO DE DATOS

TABLA PACIENTE

Almacena todos los datos de un paciente

Nombre	Descripción	Tipo
id_paciente	**Llave primaria**	int(10)
nombres	Nombre completo del paciente	varchar(50)
apellidos	Apellido completo del paciente	varchar(50)
paciente	Nombre de usuario del paciente	varchar(30)
clave	Clave del paciente	varchar(40)
edad	Edad del paciente	date
sexo	Sexo del paciente	enum('m','f')
lugar	Lugar de origen del paciente	varchar(40)
dirección	Dirección donde habita el paciente	varchar(400)
teléfono	Teléfono del paciente	varchar(20)

TABLA EXAMEN

Almacena todos los datos de un examen

Nombre	Descripción	Tipo
id_examen	**Llave primaria**	int(10)
fecha	Fecha de realización del examen	date
hora	Hora de realización del examen	time
id_tipo_examen	Clave foránea del tipo de examen	int(10)
id_paciente	Clave foránea de identificación del paciente	int(10)

TABLA EXAMEN

Almacena todos los tipos de exámenes que puede realizarse un paciente

Nombre	Descripción	Tipo
id_tipo_examen	**Llave primaria**	int(10)
nombre_tipo_examen	Nombre del tipo de examen	varchar(40)

TABLA USUARIO

Almacena todos los datos de un usuario

Nombre	Descripción	Tipo
id_usuario	**Llave primaria**	int(10)
nombres	Nombre completo del usuario	date
apellidos	Apellido completo del usuario	time
nick	Nombre que identifica al usuario en el sistema	int(10)
clave	Contraseña del usuario registrado	int(10)
admin_priv	Tipo de usuario registrado	enum('s','n')

TABLA PARASITOLOGÍA

Almacena todos los datos de parasitología

Nombre	Descripción	Tipo
color	Color del resultado de un examen	varchar(20)
consistencia	Consistencia del resultado de un examen	varchar(25)
examen_microscopico	Examen micróscopico del resultado de un examen	text
tincion_de_wright	Tinción de wright del resultado de un examen	set
otros	Otros valores del resultado de un examen	text
id_examen	Clave foránea del examen realizado	int(10)

TABLA HEMATOLOGÍA

Almacena todos los datos de hematología

Nombre	Descripción	Tipo
hematocrito	Hematócrito del resultado de un examen	float(4,2)
hemoglobina	Hemoglobina del resultado de un examen	float(4,2)
malaria	Malaria del resultado de un examen	enum
rd_plaquetas	Recto de plaquetas del resultado de un examen	int(11)
rd_g_blancos	Recto de glóbulos blancos del resultado de un examen	int(11)
n_segmentados	Neutrófilos segmentados del resultado de un examen	int(11)
linfocitos	Linfocitos del resultado de un examen	int(11)
eosinofilos	Eosinófilos del resultado de un examen	int(11)
basofilos	Basófilos del resultado de un examen	int(11)
monocitos	Monócitos del resultado de un examen	int(11)
n_en_banda	Neutrófilos en banda del resultado de un examen	int(11)
id_examen	Clave foránea del examen realizado	int(10)

TABLA UROANÁLISIS

Almacena todos los datos de uroanálisis

Nombre	Descripción	Tipo
color	Color del resultado de un examen	varchar(25)
aspecto	Aspecto del resultado de un examen	varchar(255)
densidad	Densidad del resultado de un examen	int(11)
ph	Ph del resultado de un examen	int(11)
nitritos	Nitritos del resultado de un examen	enum
sangre	Sangre segmentados del resultado de un examen	int(11)
cetonas	Cetonas del resultado de un examen	int(11)
glucosa	Glucosa del resultado de un examen	int(11)
proteinas	Proteínas del resultado de un examen	int(11)
urobilinogeno	Urobilinogeno del resultado de un examen	int(11)
leucoesteraza	Leucoesteraza del resultado de un examen	int(11)
bilirrubina	Bilirrubina del resultado de un examen	int(11)
globulos_blancos	Glóbulos blancos del resultado de un examen	varchar(30)
globulos_rojos	Glóbulos rojos del resultado de un examen	varchar(30)
c_epiteliales	Células epiteliales del resultado de un examen	varchar(30)
urato_amorfo	Urato amorfo del resultado de un examen	varchar(30)
fosfato_amorfo	Fosfato amorfo del resultado de un examen	varchar(30)
hilos_mucosos	Hilos mucosos del resultado de un examen	varchar(30)
otros	Otros del resultado de un examen	set
id_examen	Clave foránea del examen realizado	int(10)

XIV. MAPA DEL SITIO

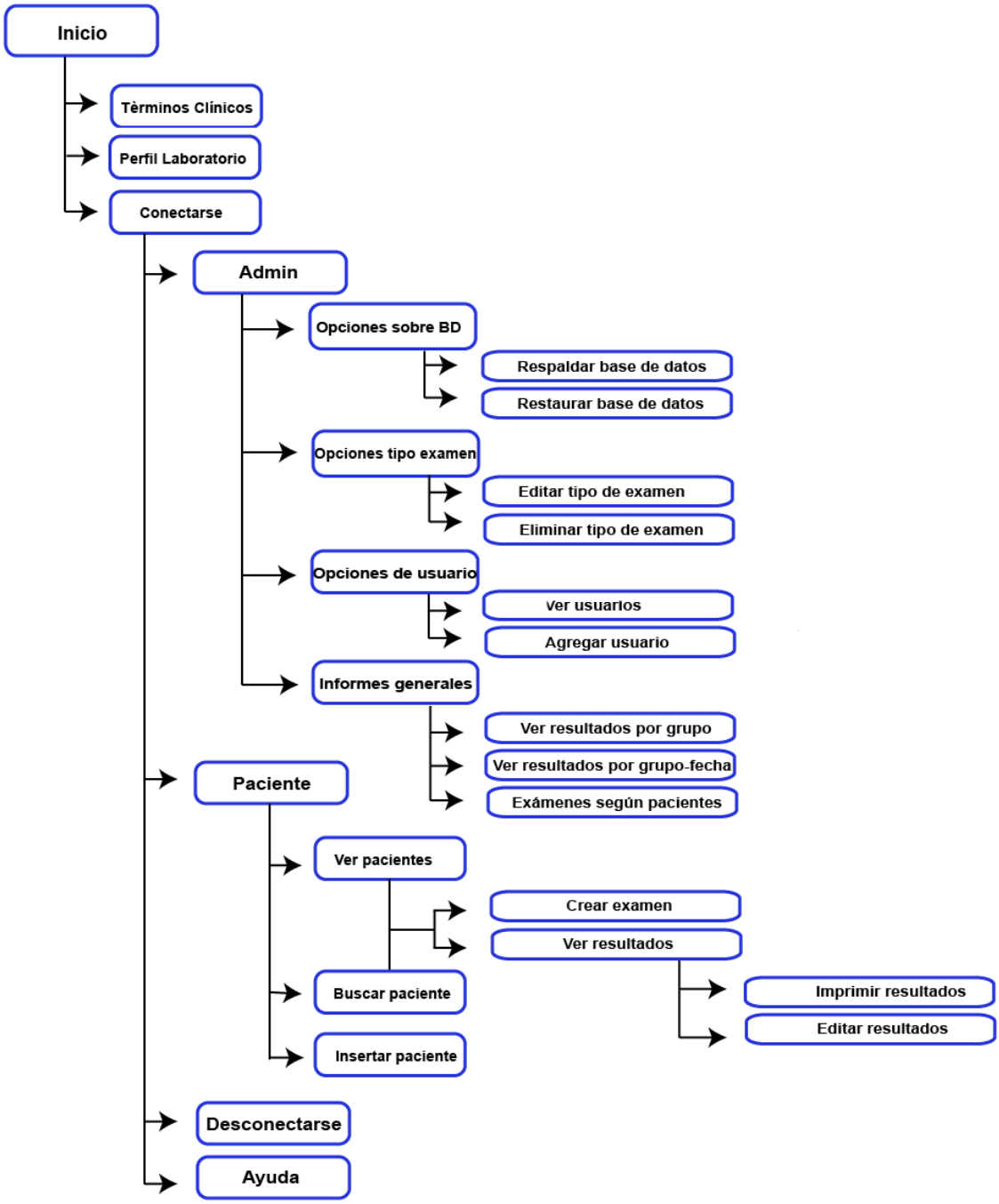


Figura 15: Mapa del sitio

XV. DISEÑO DE INTERFAZ

Figura 16: Interfaz de Ingresar a la aplicación

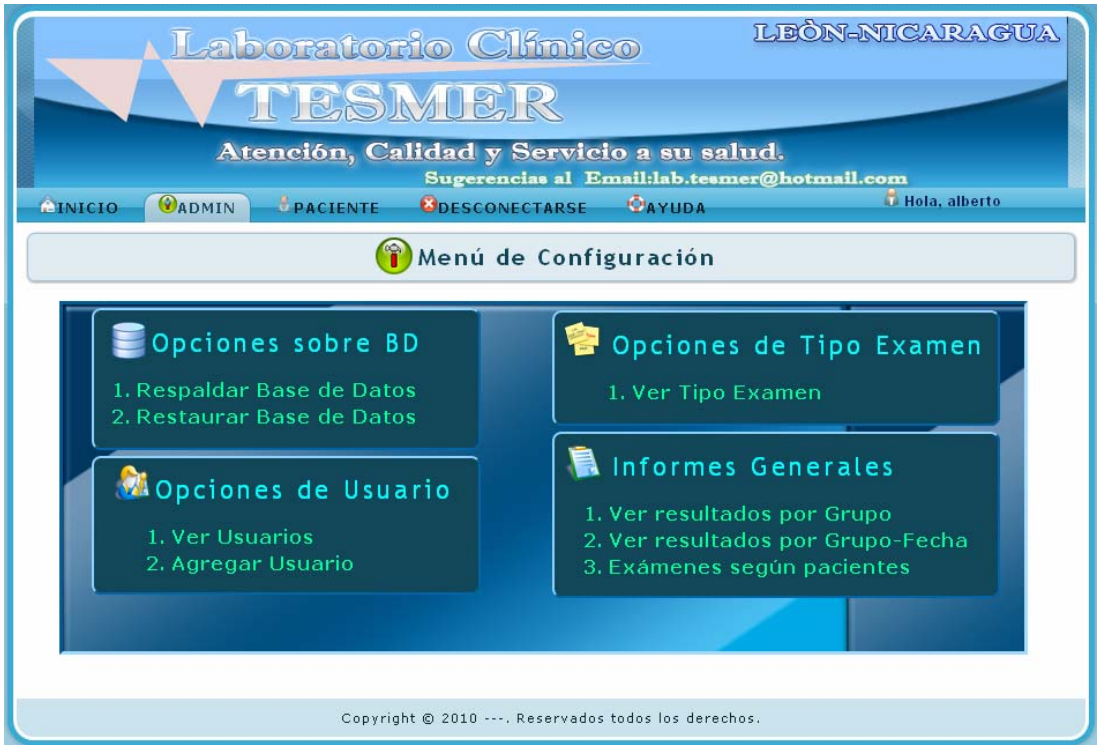


Figura 17: Interfaz de administración

Figura 18: Interfaz de insertar paciente

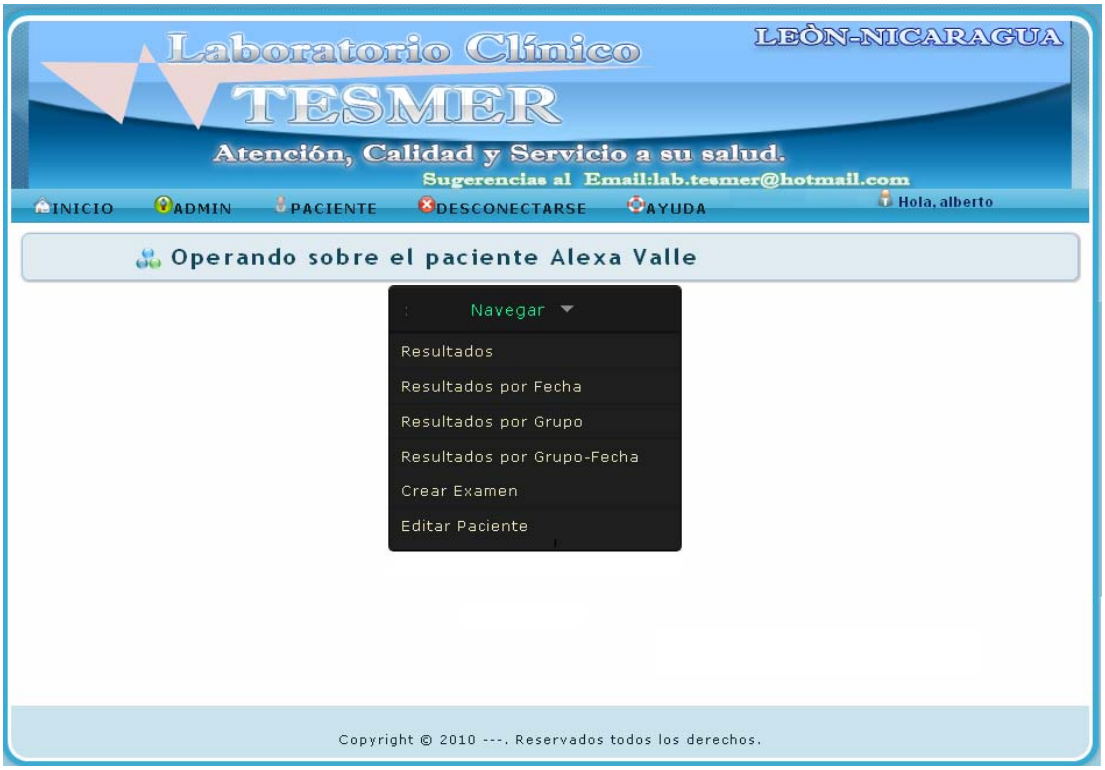


Figura 19: Interfaz de menú de navegación

Figura 20: Interfaz de buscar paciente

Figura 21: Interfaz de resultado de exámenes

Figura 22: Interfaz de resultado de exámenes por grupo

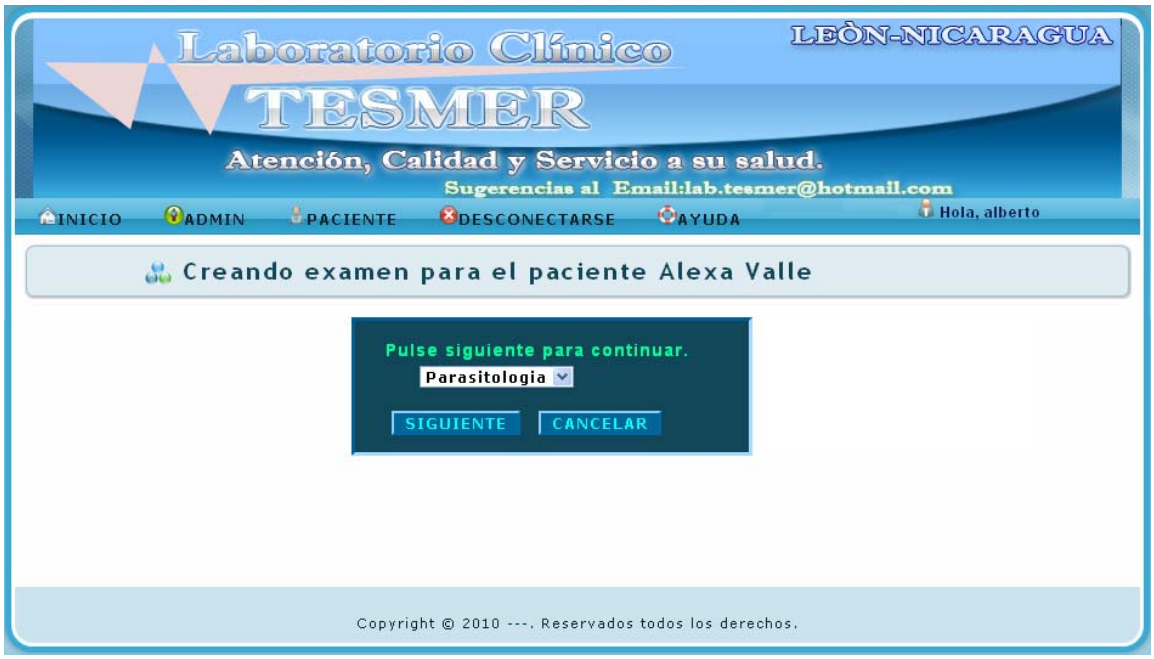


Figura 23: Interfaz de crear examen

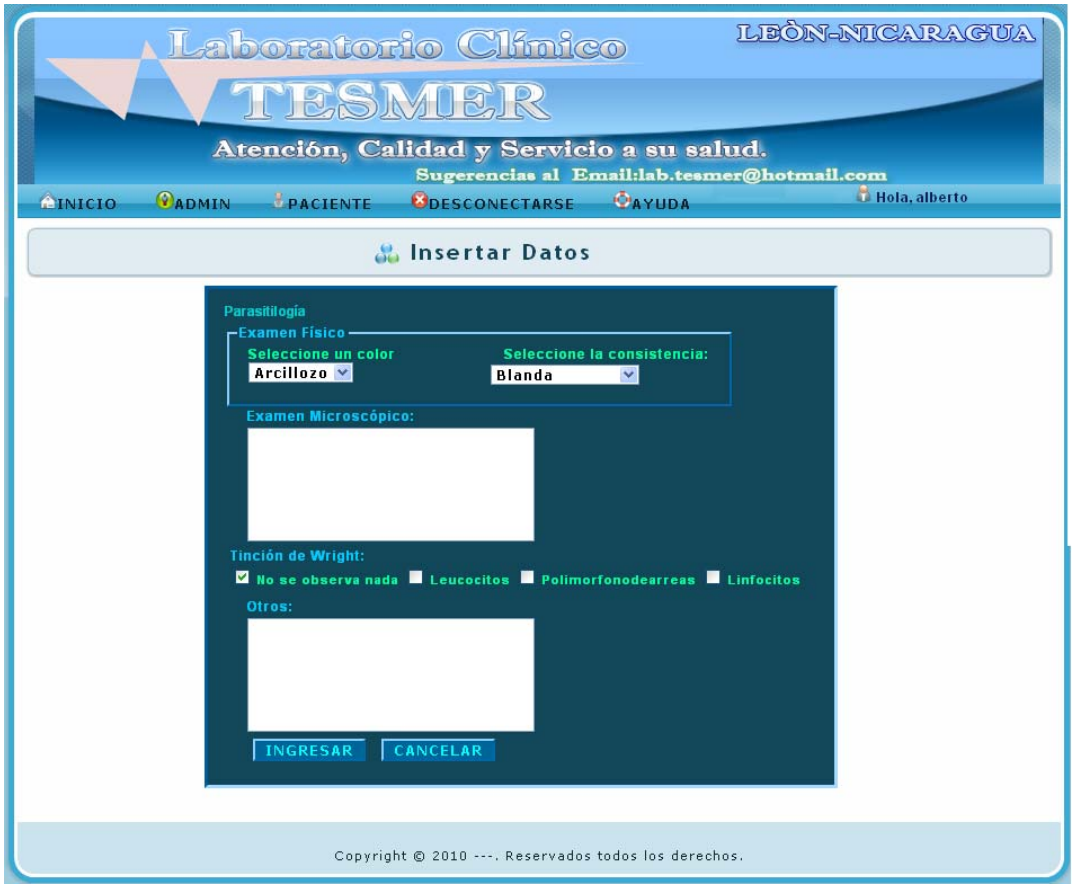


Figura 24: Interfaz de insertar datos de examen

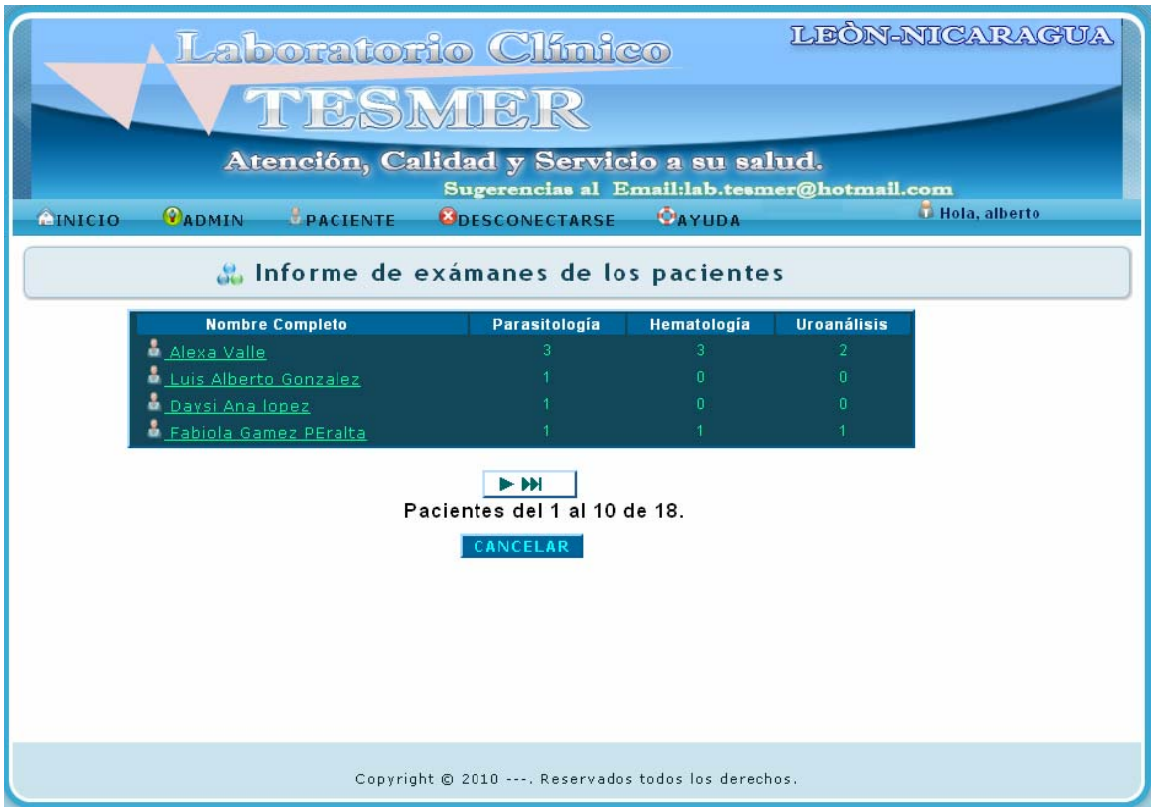


Figura 25: Interfaz de exámenes por paciente

Figura 26: Interfaz de ver usuarios

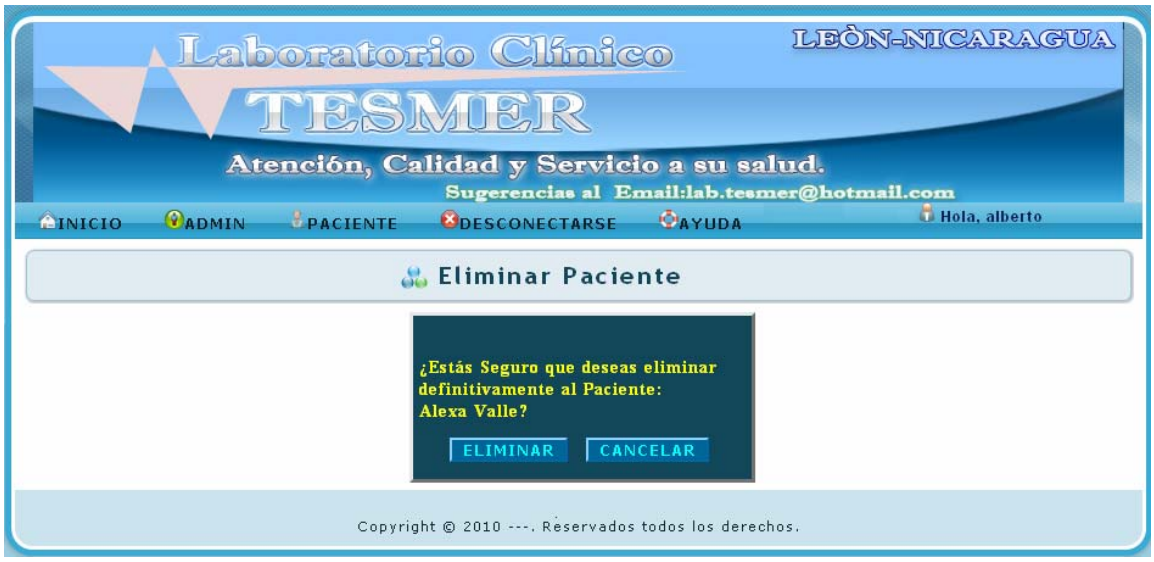


Figura 27: Interfaz de eliminar paciente

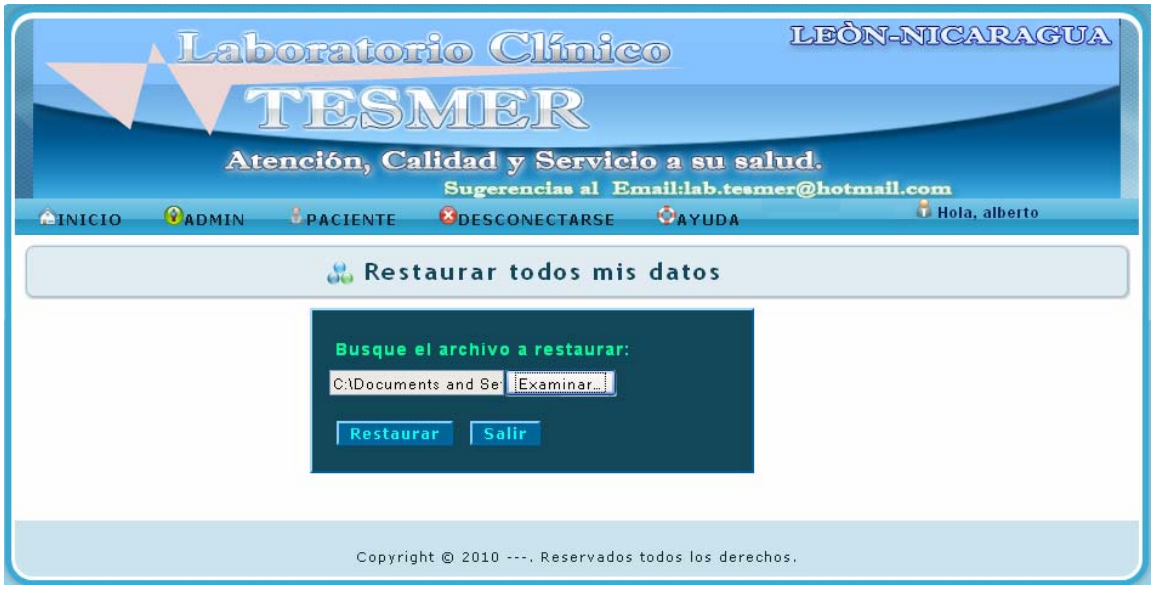


Figura 28: Interfaz de restaurar base de datos

XVI. CONCLUSIONES

Del presente trabajo puedo concluir lo siguiente:

- Se permitió el registro de pacientes que se realizaron exámenes en el Laboratorio Clínico TESMER de manera más fácil.
- Se logró gestionar los diferentes exámenes que se realizan los pacientes de forma segura y muy rigurosa de cara a la base de datos y la aplicación.
- Se crearon interfaces para respaldar y restaurar la base de datos siempre que un usuario con Rol de Administrador lo desee.
- Se generaron reportes de los pacientes con sus respectivos exámenes.
- Se logró la definición de los roles de que usuario que permite la manipulación segura de la aplicación.

Con lo anterior puedo decir que hemos diseñado y desarrollado una aplicación segura y confiable que resuelve la problemática del Laboratorio Clínico TESMER.

XVII.
RECOMENDACIONES

Terminado el trabajo consideró los siguientes aspectos interesantes para la continuación y mejora de esta aplicación:

- Mejorar la apariencia de la página de entrada a la aplicación para que sea más atractiva y así dar una mejor publicidad al laboratorio.

- Analizar la posibilidad de agregar más consultas a la aplicación para llevar un mayor detalle de los resultados de los exámenes.

- Hacer que esta aplicación Web sea capaz de ejecutarse en cualquiera de los navegadores de internet actuales.

- Extender el número interfaces que permitan gestionar los diferentes exámenes que pueda realizarse un paciente.

XVIII.BIBLIOGRAFÍA

Cancino, C. (1 de Agosto de 2010). *Web Master Cesar Cancino Zapata.* Recuperado el 20 de Agosto de 2010, de Web Master Cesar Cancino Zapata: http://www.cesarcancino.com/?cat=2

Conde, J. (09 de Septiembre de 2004). *Videotutoriales - Downloads.* Recuperado el 2 de Diciembre de 2008, de Videotutoriales - Downloads: http://www.illasaron.com/html/modules.php?name=Downloads&d_op=viewdownload&cid=70

Conde, J. (03 de Agosto de 2006). *Videotutoriales - Downloads.* Recuperado el 30 de Octubre de 2010, de Videotutoriales - Downloads: http://www.illasaron.com/html/modules.php?name=Downloads&d_op=viewdownload&cid=104

Conde, J. (10 de Noviembre de 2008). *Videotutoriales - Downloads.* Recuperado el 5 de Marzo de 2010, de Videotutoriales - Downloads: http://www.illasaron.com/html/modules.php?name=Downloads&d_op=viewdownload&cid=142

Project, T. j. (5 de Julio de 2010). *UI/API/1.7.2/Datepicker.* Recuperado el 6 de Septiembre de 2010, de jQuery: The Write Less, Do More, JavaScript Library:

http://docs.jquery.com/UI/API/1.7.2/Datepicker
Rasmus Lerdorf, H. M. (22 de Julio de 2010). *PHP: mysql_result- Manual.* Recuperado el 5 de Noviembre de 2010, de PHP: Manual: http://mx2.php.net/manual/es/function.mysql-result.php

Team, t. j. (5 de Julio de 2010). *jQuery UI - Datepicker Demos and Documentation.* Recuperado el 6 de Septiembre de 2010, de jQuery UI - Demos and Documentation: http://jqueryui.com/download

Yanes, F. (14 de Abril de 2010). *Blog de Fernando Yanes: Videotutorial de Jquery.* Recuperado el 25 de Agosto de 2010, de Blog de Fernando Yanes: http://fdoyanes.blogspot.com/2010/04/05-videotutorial-de-jquery.html

XIX. ANEXOS

Los componentes necesarios para la elaboración de la aplicación fueron los siguientes:

WAMP: (Windows-Apache-MySQL- PHP/Python/PERL).

El término hace referencia al sistema creado por la conjunción de esas aplicaciones libres (de código abierto) y el sistema operativo Windows. Este grupo de aplicaciones generalmente son usados para crear servidores web.

WAMP provee a los desarrolladores con los cuatro elementos necesarios para un servidor web: un sistema operativo (Windows), un manejador de base de datos (MySQL), un software para servidor web (Apache) y un software de programación script web (PHP, Python o PERL).

Instalación de WAMP para Windows:
Abra el instalador, verá una ventana como la siguiente:

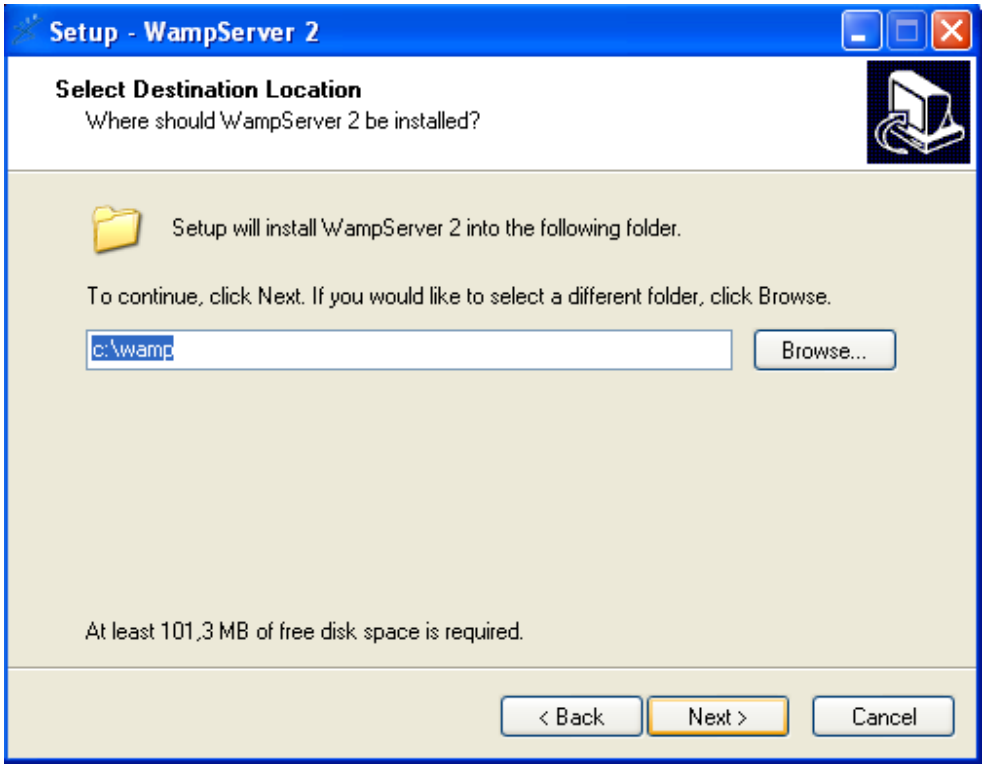


Figura 29: Instalador de WAMP

Seleccione el directorio en el cual desea instalar el programa y de clic en el botón siguiente para continuar la instalación.

Después de instalar WAMP podrá encontrarlo en el **área de notificaciones**. Allí encontrará el Panel de Control de WAMP a través del cual podrá iniciar y parar el servidor e instalar y desinstalar servicios.

Ahora comprobaremos que WAMP fue instalado correctamente:

1. Abra una ventana del navegador.
2. Escriba http://localhost/ o 127.0.0.1
3. Verá una página como la siguiente:

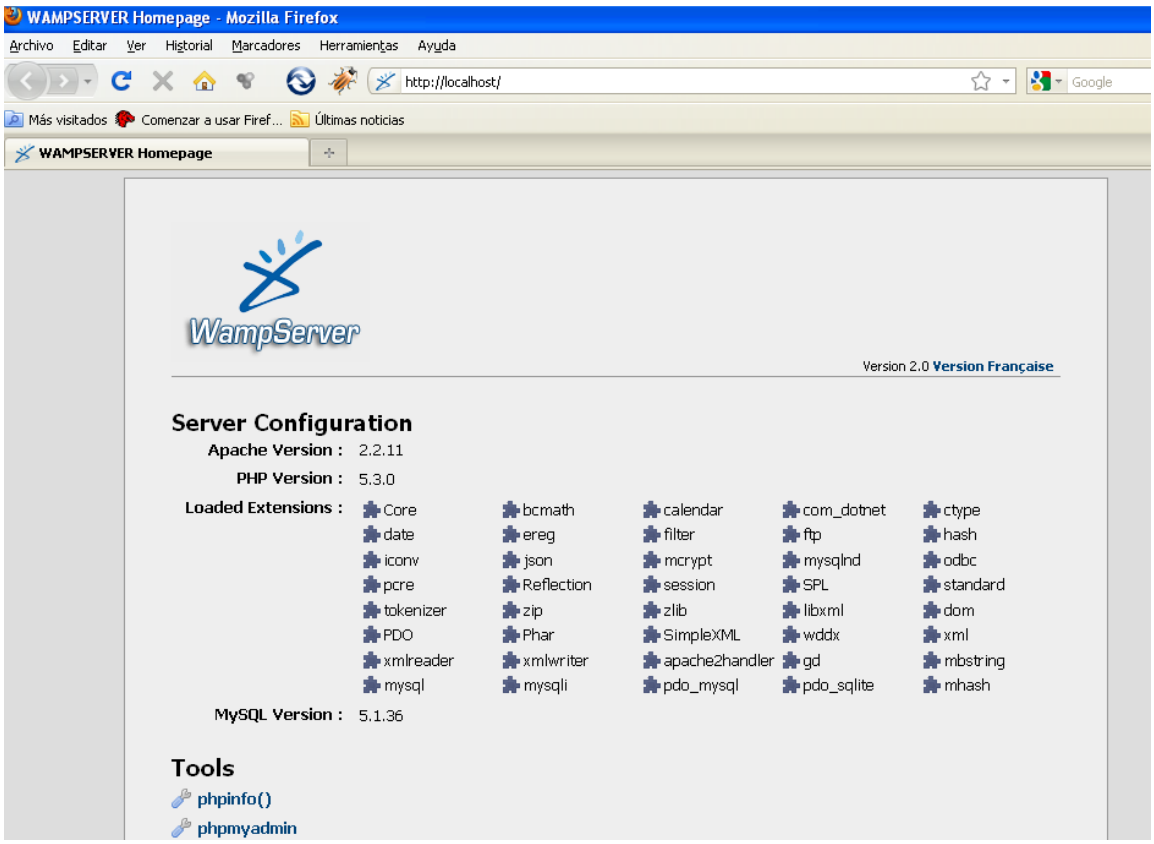


Figura 30: Probando WAMP

Para la elaboración del código se utilizó Adobe DreamWeaver CS3 pero también puede utilizarse phpExpertEditor, NotePad++ o cualquier otro editor de texto. Las páginas creadas se guardarán dentro de la carpeta www para ser accedidas desde el servidor, la carpeta está dentro de C:\ wamp.

DETALLES DE LAS CLASES DEL DIAGRAMA

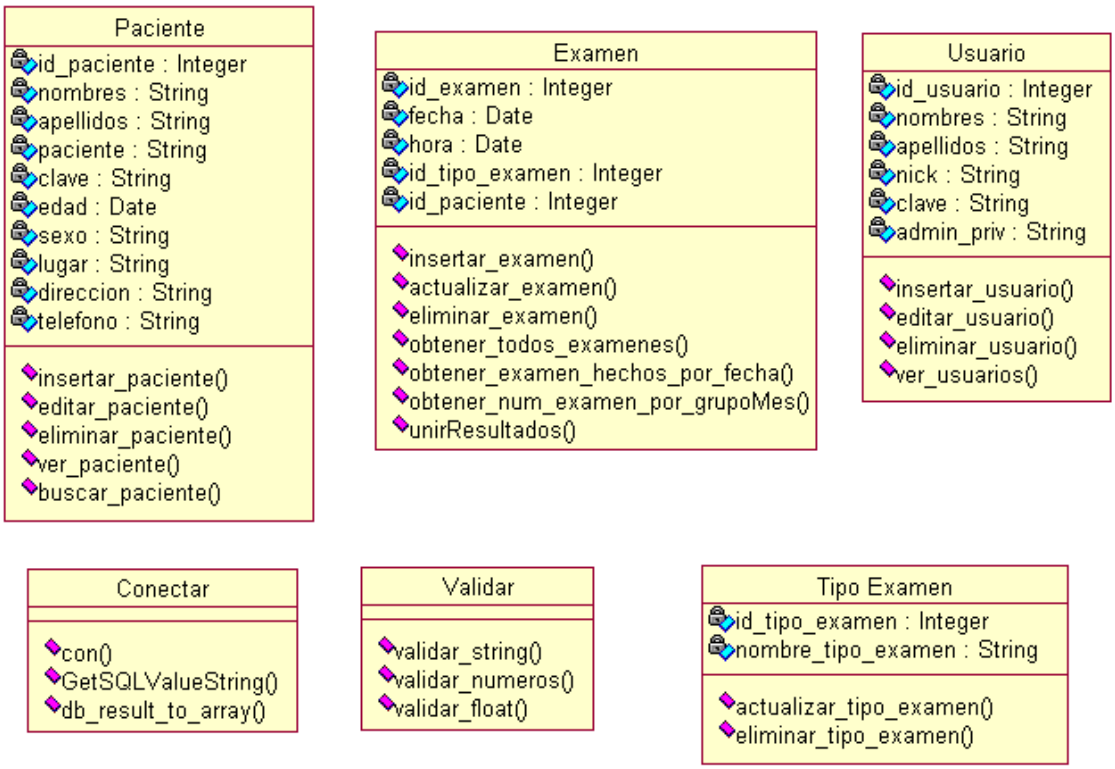


Figura 31: Clases del diagrama-parte1

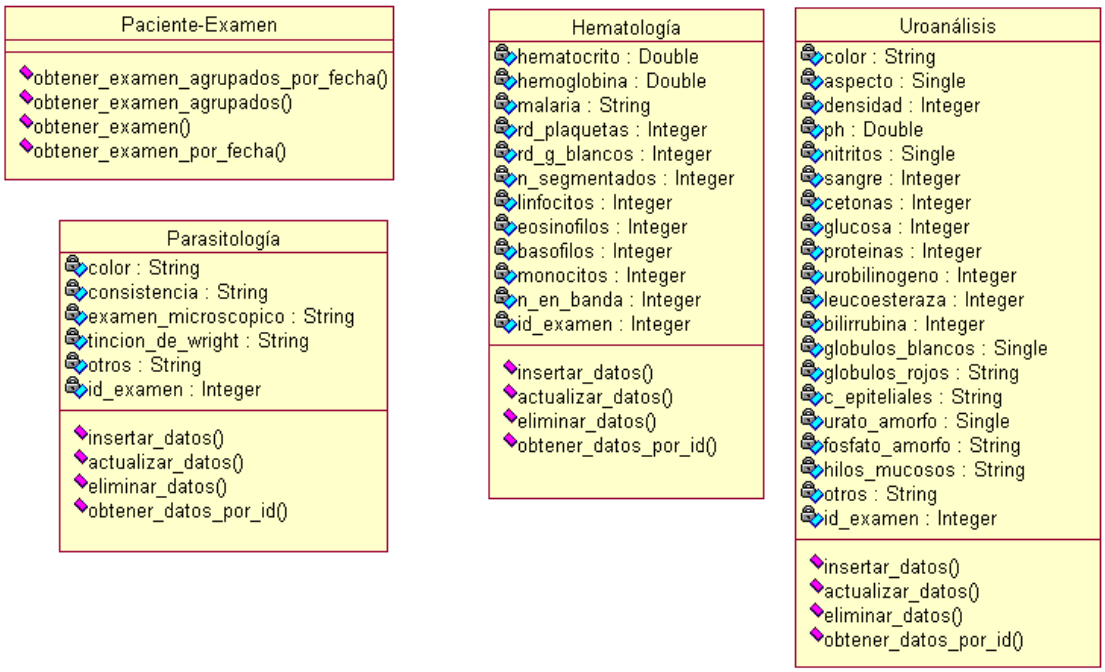


Figura 32: Clases del diagrama-parte2

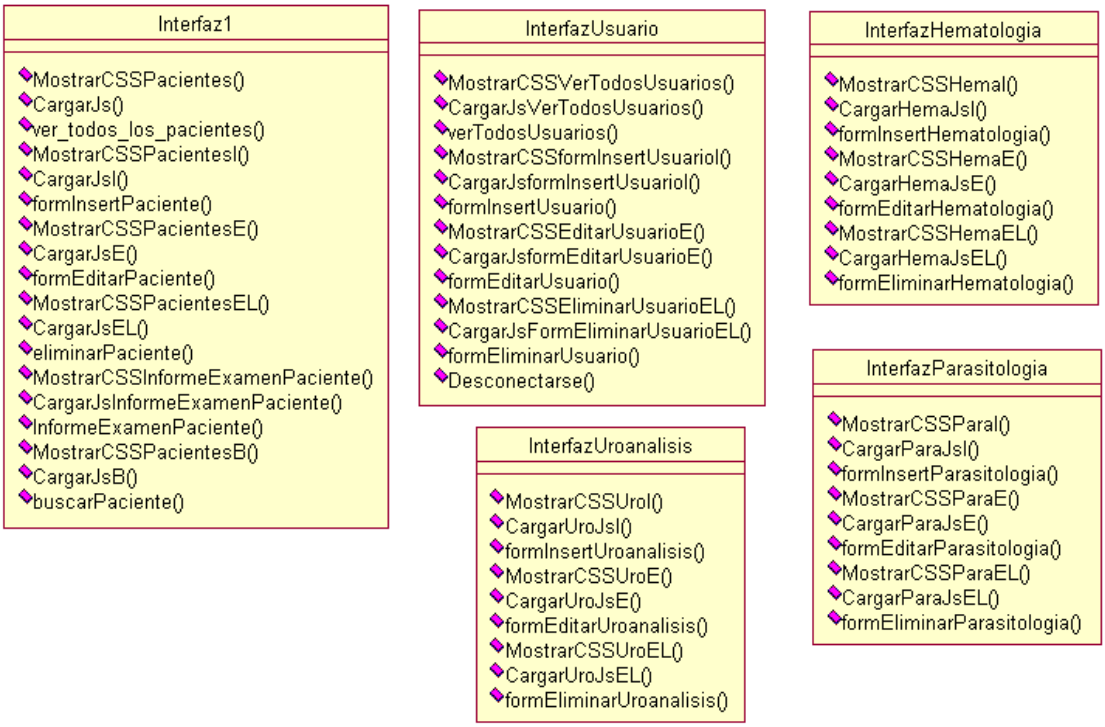


Figura 33: Clases del diagrama-parte3

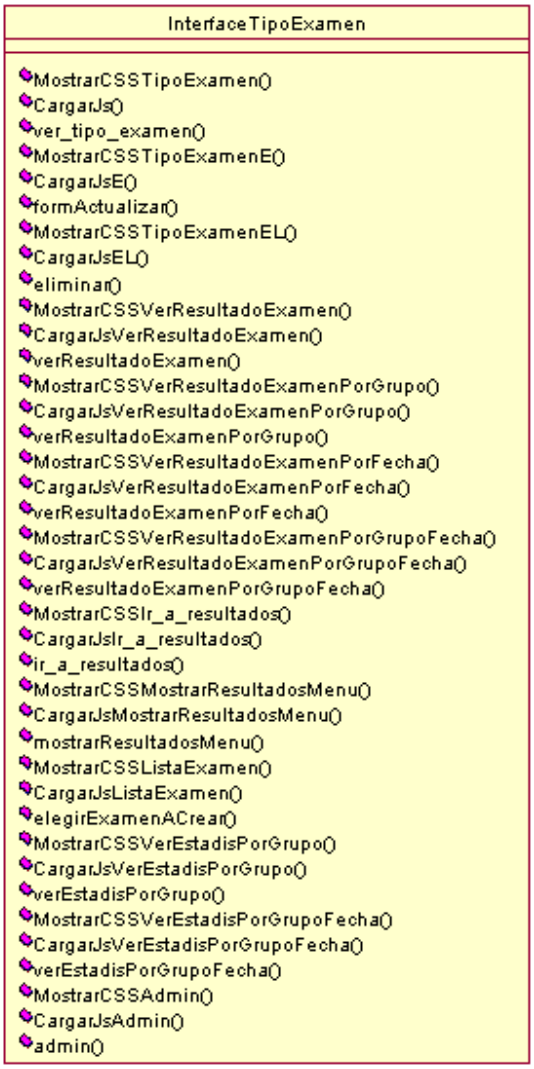


Figura 34: Clases del diagrama-parte4