

## Taller Tolerancia a fallos y Transacciones Distribuidas

Yulieth Cristina Huérfino Pérez  
Laura Alejandra Montañez Estupiñan  
Viviana Alejandra Tirano Parra

Ingeniería de sistemas y computación  
Universidad Pedagógica y Tecnológica de Colombia  
Sogamoso- Boyacá  
2021

## Taller Tolerancia a fallos y Transacciones Distribuidas

Yulieth Cristina Huérfino Pérez  
Laura Alejandra Montañez Estupiñan  
Viviana Alejandra Tirano Parra

Presentado a: Camilo Harvey Bohórquez Dallos  
Ingeniero de sistemas y computación, Especialista en gerencia de proyectos,  
Ms(c) Maestría en Tecnología Informática

Ingeniería de sistemas y computación  
Universidad Pedagógica y Tecnológica de Colombia  
Sogamoso- Boyacá  
2021

## **Introducción**

Los sistemas distribuidos son sistemas cuyos componentes hardware y software que están en ordenadores conectados en red, se comunican y coordinan sus tareas mediante el paso de mensajes, logrando un objetivo. Se logra establecer una comunicación mediante un protocolo: Cliente- servidor.

Las transacciones distribuidas, las operaciones de transacción abarcan múltiples servicios, múltiples servidores y múltiples bases de datos. En pocas palabras, una operación de datos de gran tamaño se divide en muchas operaciones pequeñas, y estas operaciones pequeñas se ejecutan en servidores que pertenecen a diferentes aplicaciones.

El proyecto sobre el que se está trabajando corresponde a un sistema de información que muestra los datos personales de los pacientes del centro de salud multiservicios y también los datos médicos de los pacientes en la misma tabla. También se permite visualizar un archivo con los datos de los pacientes en formato PDF, pero teniendo en cuenta que este archivo se encuentra almacenado en un servidor de archivos en otra máquina.

En el presente trabajo se encuentra la documentación realizada para dos ejercicios prácticos desarrollados con el fin de aplicar los conceptos sobre tolerancia a fallos y transacciones distribuidas. Los mismos se implementaron en el ejercicio que se estaba llevando a cabo sobre conexión remota a través de túneles entre diferentes máquinas que no están conectadas a la misma red.

Durante el desarrollo de la documentación se van a presentar los factores involucrados en la tolerancia a fallos y las soluciones propuestas para que el sistema sea lo más estable posible ante la presencia de fallos durante su ejecución. Por otro lado, se va a diseñar una base de datos distribuida utilizando transacciones que ayudan a garantizar la consistencia de la información minimizando el bloqueo de datos y maximizando la concurrencia de los procesos.

Finalmente se va a mostrar el uso de lenguajes de programación en el proceso de transacciones distribuidas con gestores de bases de datos que permiten relacionar remotamente la información de un sistema a través de paneles de control, para administrar servidores de alojamiento web.

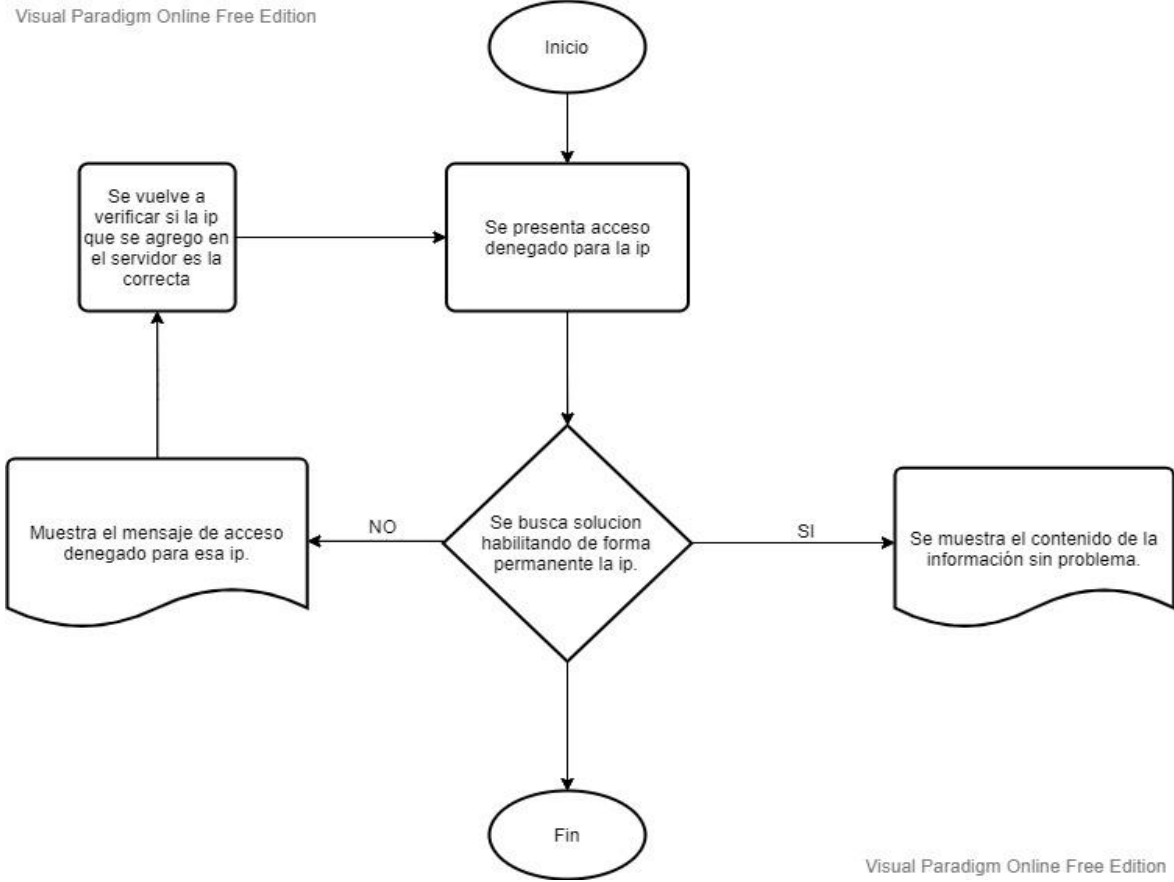
### Requisitos del proyecto

Se realizaron tres estrategias de tolerancia a fallos de las cuales 2 son de tipo interno y una es de tipo externa, a continuación se especificarán cada una de ellas por medio de tablas cada una con sus respectivas especificaciones.

#### Estrategias de tolerancia a fallos

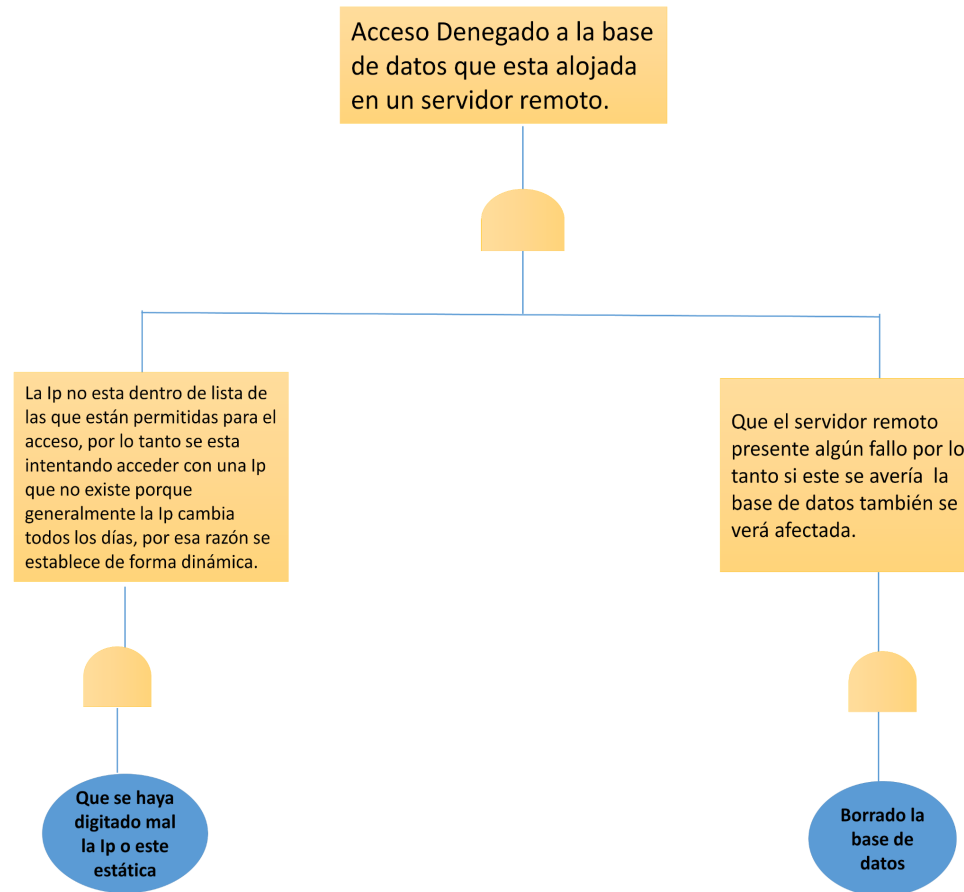
##### ESTRATEGIA INTERNA #1

<b>Nombre a la estrategia: Habilitar ip para que sea dinámica.</b>	
<b>Descripción del fallo</b>	El servidor no permite el acceso a una ip estática.
<b>Descripción de la estrategia</b>	Se debe habilitar la ip de manera permanente, de modo que la ip que caso presentaba un acceso denegado la cual es 190.156.60.191 pero esta cambia todos los días, por lo tanto para habilitar la conexión remota permanente se pone la ip 190.156.%.% y este símbolo % garantiza que así cambien esos números el los va a tomar y se va a mantener el flujo de la información que en este caso es la tabla con los registros de los pacientes.
<b>Descripción de la tolerancia a Fallos</b>	Se buscará habilitar la ip de forma dinámica para que no se presente inconvenientes y se pueda visualizar la información correspondiente a la tabla de pacientes.

<b>Tipo de Estrategia</b>	<p>Es un tipo de fallo interno ya que sin el cargue de la base de datos en este caso de la tabla de pacientes no se podría visualizar la información ni realizar acciones dentro del sistema.</p>
<b>Flujo de información</b>	<p>Visual Paradigm Online Free Edition</p>  <pre> graph TD     Inicio([Inicio]) --&gt; Denegado[Se presenta acceso denegado para la ip]     Denegado --&gt; Decision{Se busca solución habilitando de forma permanente la ip.}     Decision -- SI --&gt; SinProblema[/Se muestra el contenido de la información sin problema./]     Decision -- NO --&gt; Mensaje[Muestra el mensaje de acceso denegado para esa ip.]     Mensaje --&gt; Verificar[Se vuelve a verificar si la ip que se agrego en el servidor es la correcta]     Verificar --&gt; Denegado     Decision --&gt; Fin([Fin])   </pre> <p>Diagrama 1. Flujo de datos en la primer estrategia de tolerancia a fallos</p> <p>Visual Paradigm Online Free Edition</p>

**Escenario de  
ocurrencia del fallo**

El fallo se presenta cuando se intenta acceder al servidor que en este caso es el mismo phpmyadmin pero remoto donde está alojada la base de datos que en este caso es una tabla de pacientes; por lo tanto en el servidor se habilitó 3 ip que pueden acceder al servidor donde está la de base datos y al encontrar una ip que no esté dentro de la lista no permite el acceso.



## Diagrama 2. Escenario de ocurrencia del fallo

### Flujo de corrección del Error

1. Se presenta el error que se observa en la imagen:

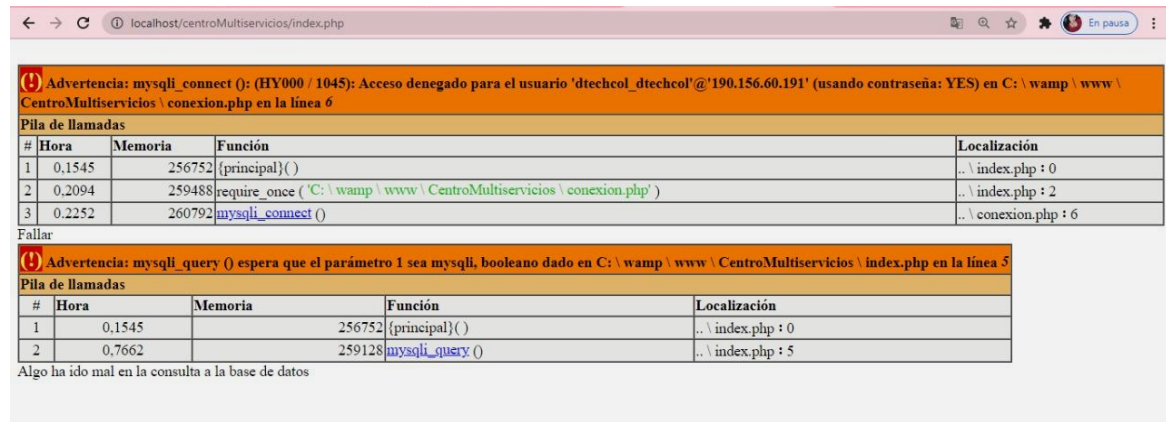


figura 1. Error presentado en el fallo por base de datos

2. Luego de esto lo que se hizo fue verificar si la ip que presentaba problemas estaba dentro de la lista de las que estaban permitidas, al ver que no se encontraba allí se habilitó de manera permanente en remote mysql para que la ip que se ingresara tuviera acceso a la base de datos como se muestra a continuación.

**cPanel Remote MySQL®**

Add a specific domain name to allow visitors to connect to your MySQL databases. Applications like bulletin boards, online shopping carts, and content management systems can use this feature. For more information, read the [documentation](#).

### Add Access Host

Host (% wildcard is allowed)

Comment (optional)

[Add Host](#)

*Figura 2. Tolerancia de fallos implementada*

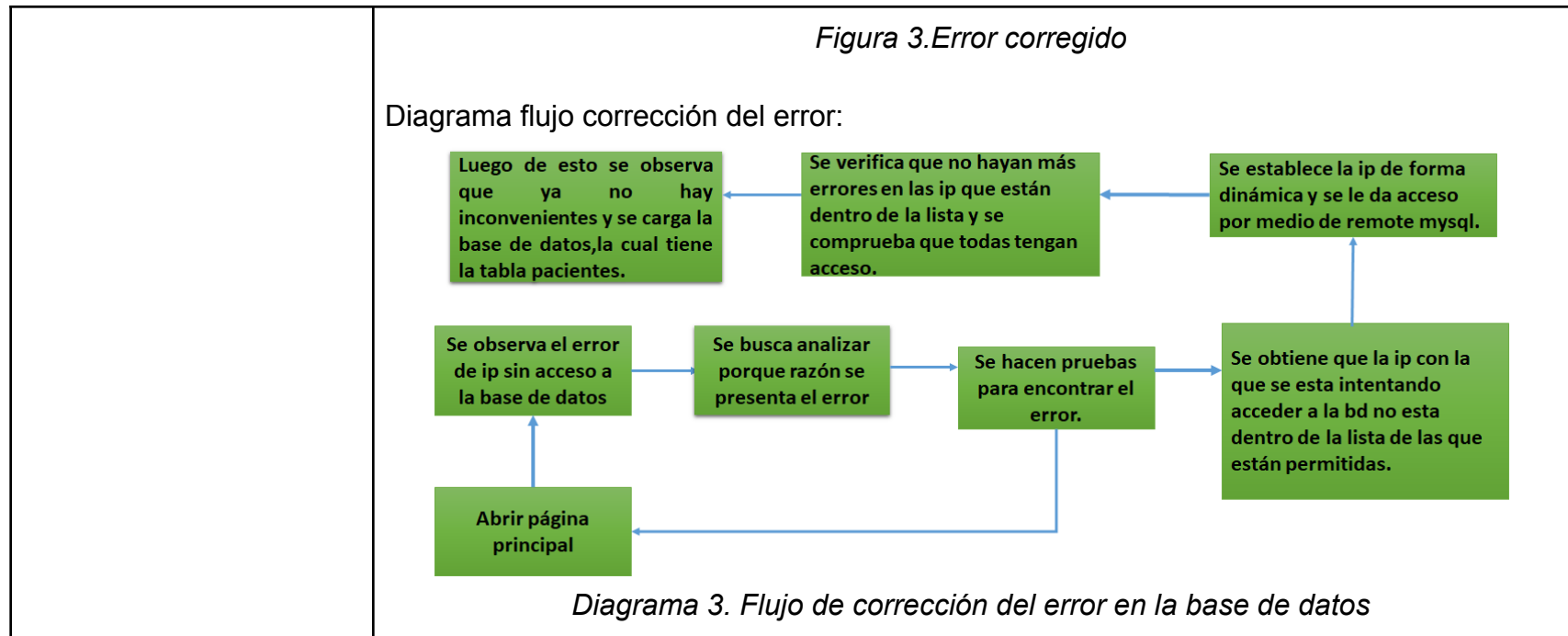
3. Luego de esto se puede ver que cargó la base de datos actual sin ningún error.

Nombre	Apellido	Identificación	Ciudad	Dirección	Número de contacto	Ocupación	Correo Electrónico	Eps	Diagnóstico	Tratamiento
Carmen	Alvarez	1234567890	Sogamoso	Centro	3104456789	Ama de casa	carmen@gmail.com	Nueva eps	Paciente con dolor abdominal	Se le asigna una ecografía
Lina	Castro	1236784321	Duitama	Postobon	3216543789	Estudiante	lina@hotmail.com	Sisben	Paciente para citología	Se le asigna una colposcopia
Laura	Para	1057643897	Noboa	Centro	3165537815	Estudiante	laura@gmail.com	Nueva eps	Paciente con fractura	Se le asigna una radiografía
Dario	Gonzalez	1346892345	Sogamoso	Uptc	3003672637	Enfermero	dario@gmail.com	Nueva eps	Paciente para revision	ninguno
Helena	Diaz	22874567	Tibasosa	Cll 10 # 8-12	3217748215	Independiente	helena@gmail.com	Sisben	Paciente con fuerte dolor de cabeza	Se le asigna un tac
Alberto	Perez	1057456324	Sogamoso	Magdalena	3216578934	Soldador	alberto@gmail.com	Nueva eps	Paciente con síntomas de covid	Se le asigna hospitalización
Martha	Rincon	1067543456	Sogamoso	Pradera	3227436789	Ama de casa	martha@gmail.com	Nueva eps	Hemorragia de vías digestivas	Se le asigna pastilla omeprazol 40
Nicolas	Zambrano	106845821	Sogamoso	Cll 10 # 24-15	3218885643	Estudiante	nicolas@gmail.com	Nueva eps	Exámenes de rutina	Se le asigna cita con especialista

[Ver políticas](#)  
[Ver pacientes](#)

Copyright © Todos los derechos reservados | El presente software fue diseñado en Colombia





### ESTRATEGIA EXTERNA #1

<b>Nombre a la estrategia: Corregir fallo por concurrencia</b>	
<b>Descripción del fallo</b>	<p>El software puede fallar en el momento de mostrar el archivo de pacientes, ya que el servicio ngrok no permite que en la conexión al servidor de archivos hayan más de 20 conexiones por minuto. Entonces, en caso de más de 20 conexiones concurrentes el sistema puede fallar.</p>

<b>Descripción de la estrategia</b>	Vamos a utilizar una estrategia de tolerancia a fallos para ampliar la capacidad de conexiones.
<b>Descripción de la tolerancia a Fallos</b>	Para ampliar la capacidad de conexiones por minuto se va a utilizar una versión premium de ngrok con el fin de ampliar la capacidad de conexiones por minuto y tolerar el fallo teniendo en cuenta el número de pacientes esperados para conexión.
<b>Tipo de Estrategia</b>	Externa

## Flujo de información

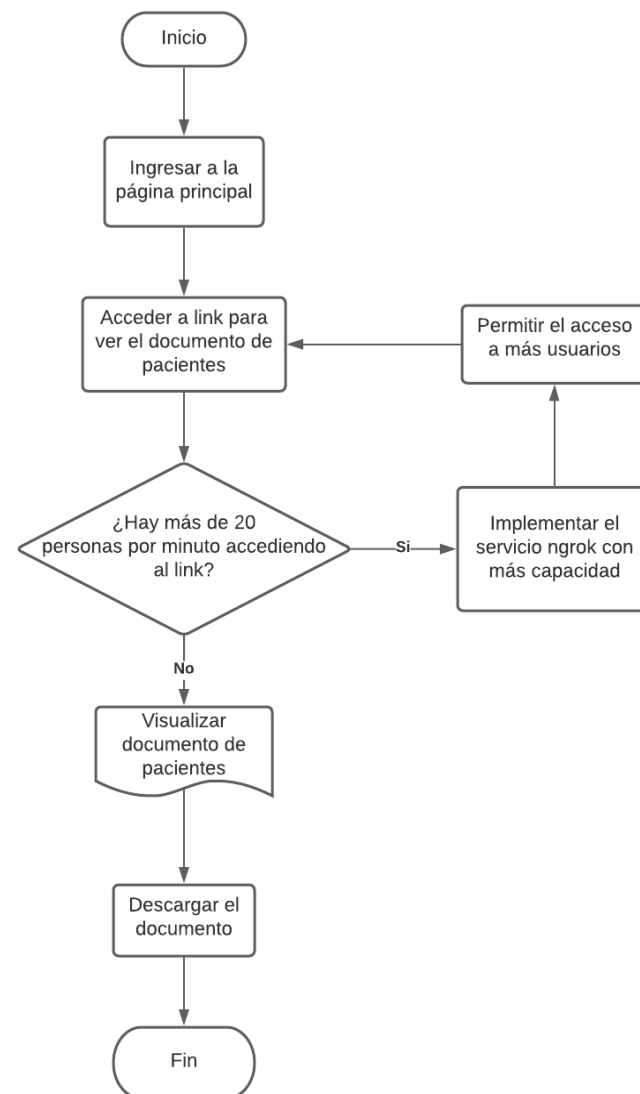
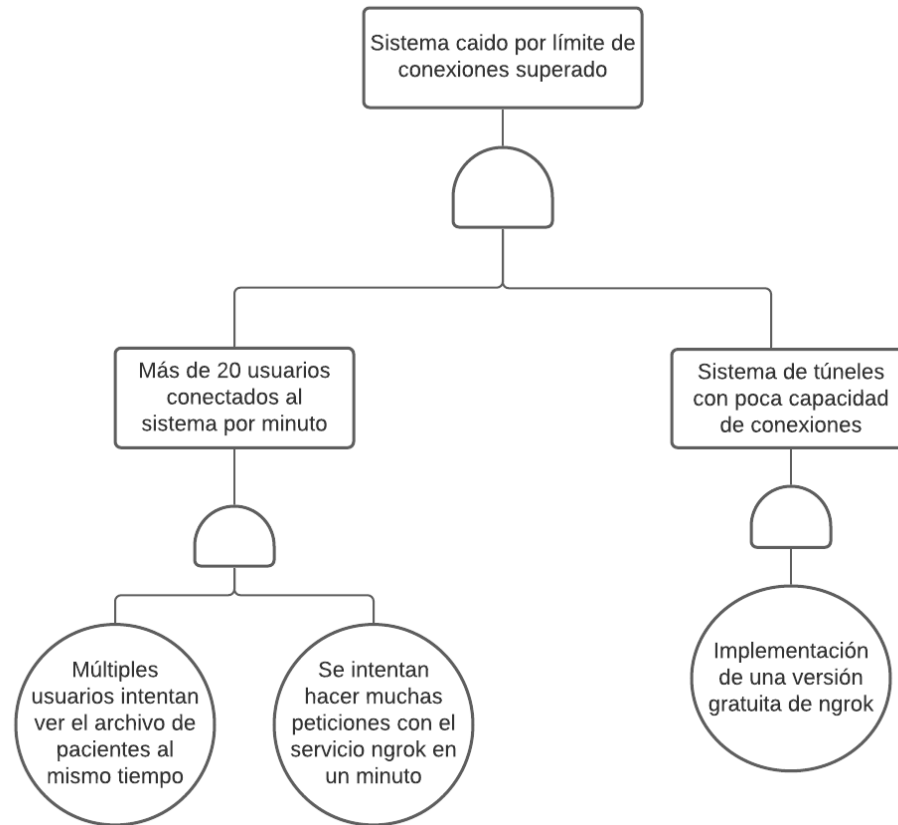


Diagrama 4. Flujo de la información en la estrategia interna 1

**Escenario de  
ocurrencia del fallo**

El fallo ocurre cuando más de 20 usuarios quieren ver concurrentemente el archivo que muestra los pacientes que se encuentra en el servidor de archivos, ya que es ahí donde se establece el túnel de conexión con la otra máquina.



*Diagrama 5. Escenario de ocurrencia del fallo por concurrencia*

## Flujo de corrección del Error

Inicialmente más de 20 usuarios pulsan en el hipervínculo de ver pacientes en el lapso de un minuto de la siguiente forma:

Gestión de pacientes

← → ↻ 🏠

localhost/Paciente/

🔍 📄 🌟

🔌 📶 📶 📶

Laura	Parra	1057643897	Nobsa	Centro	3165537815	Estudiante	laura@gmail.com	Nueva eps	Paciente con fractura	Se le asigna una radiografía
Dario	Gonzales	1346892345	Sogamoso	Uptc	3003672637	Enfermero	dario@gmail.com	Nueva eps	Paciente para revision	ninguno
Helena	Diaz	22874567	Tibasosa	Cll 10 # 8-12	3217748215	Independiente	helen@gmail.com	Sisben	Paciente con fuerte dolor de cabeza	Se le asigna un tac
Alberto	Perez	1057456324	Sogamoso	Magdalena	3216578934	Soldador	alberto@gmail.com	Nueva eps	Paciente con sintomas de covid	Se le asigna hospitalización
Martha	Rincon	1067543456	Sogamoso	Pradera	3227436789	Ama de casa	martha@gmail.com	Nueva eps	Hemorragia de vias digestivas	Se le asigna pastilla omeprazol IB
Nicolas	Zambrano	1068345821	Sogamoso	Cll 10 # 24-15	3218005643	Estudiante	nicolas@gmail.com	Nueva eps	Exámenes de rutina	Se le asigna cita con especialista

Ver políticas

Ver pacientes

Copyright © Todos los derechos reservados | El presente software fue diseñado en Colombia

Activar Windows

Vea la Configuración para activar Windows.

46d2284c555f.ngrok.io/Diagnostico Pacientes.pdf

🏠 🔍

Escribe aquí para buscar

🕒 📶 📶 📶 📶

8:29 p. m. 3/05/2021

Figura 4. Interfaz de aplicación del hipervínculo

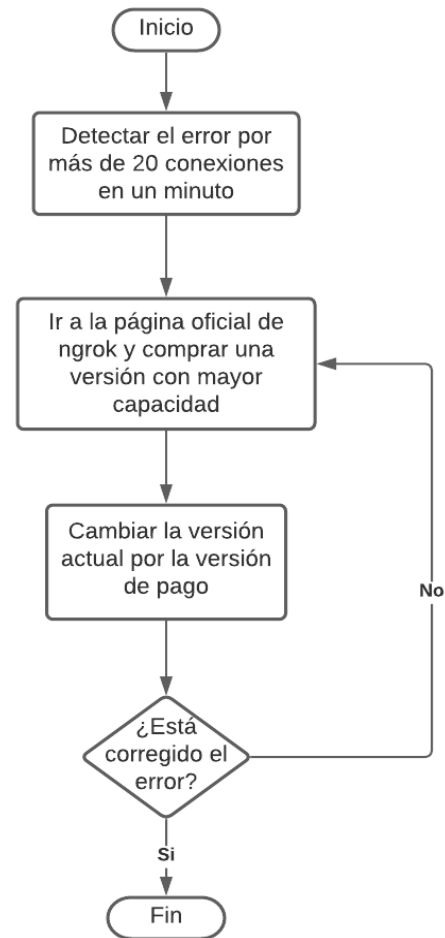
El error que aparece antes de tolerar el fallo es el siguiente:

### Too Many Connections

Too many connections! The tunnel session '1PQc5FV3ILroBYMOgARFBHY0R6b' has violated the rate-limit policy of 20 connections per minute by initiating 22 connections in the last 60 seconds. Please decrease your inbound connection volume or upgrade to a paid plan for additional capacity.

The error encountered was: **ERR\_NGROK\_702**

*Figura 5. Error por demasiadas conexiones en un minuto de ngrok*

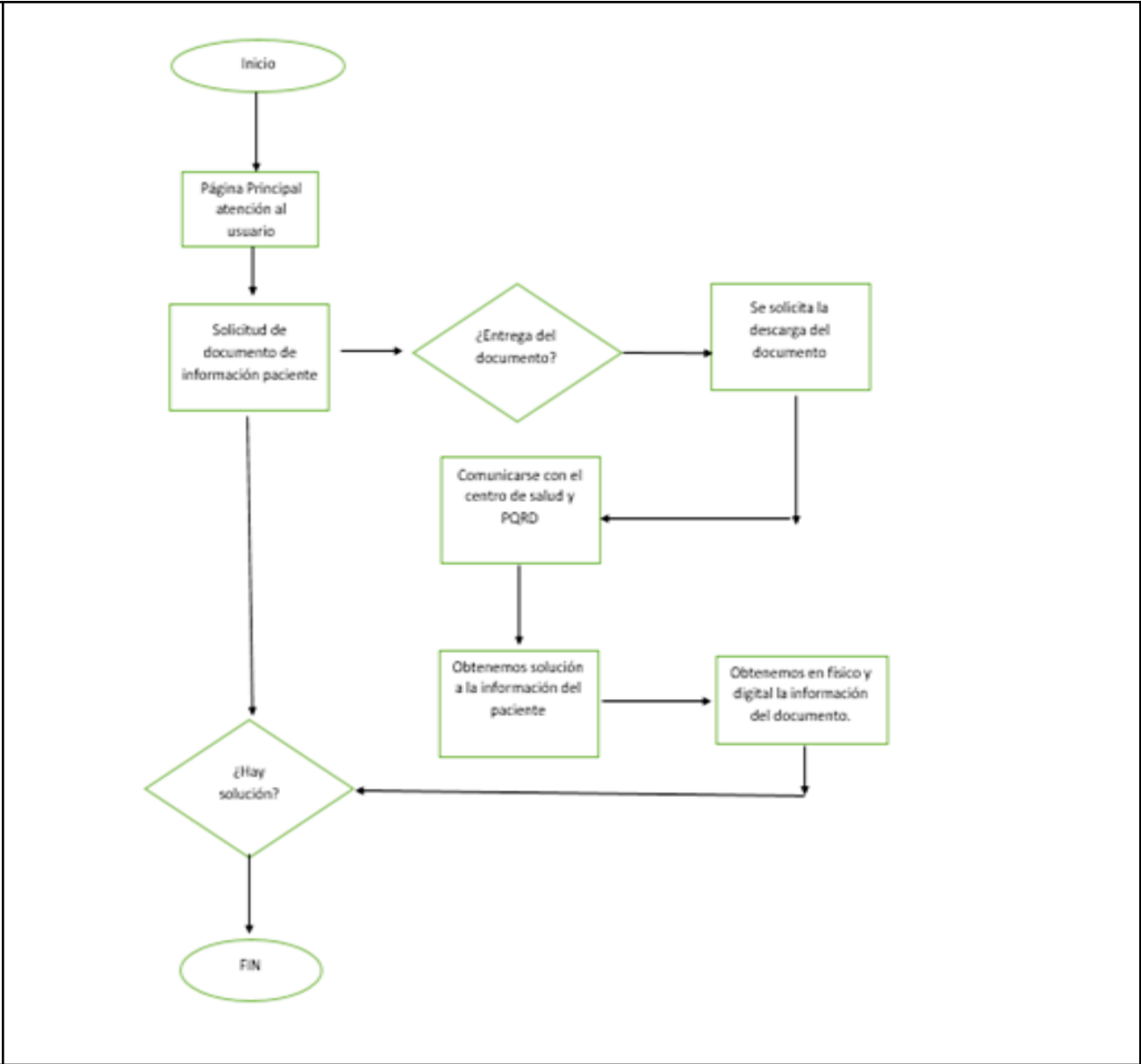


*Diagrama 6. Flujo de corrección del error por concurrencia*


### ESTRATEGIA INTERNA #3

<b>Nombre a la estrategia:</b> Fallo causado por errores en el diseño.	
<b>Descripción del fallo</b>	El sistema puede presentar fallos causados por errores en el diseño, tanto del hardware como del software debido a no seguir las especificaciones correctamente, presentándose errores de codificación o de implementación, como seleccionar un componente incorrecto. Etc. El cual puede ser permanente si no se toma alguna acción correctora.
<b>Descripción de la estrategia</b>	Se utilizará un sistema tolerante a fallos mejorando el hardware e interconexión de los componentes y especificación del software con métodos de programación orientada a objetos
<b>Descripción de la tolerancia a Fallos</b>	Podemos introducir diferentes componentes de software que sean capaces de producir otras alternativas proporcionando resultados a comparar para asegurar de que sea libre de errores los cuales serán programados por personas, lenguajes y compiladores, los cuales son ejecutados concurrentemente y comparados.
<b>Tipo de Estrategia</b>	Interna

Flujo de información
----------------------





	<p>Diagrama 7. Flujo de la información en el fallo por diseño</p>
<p><b>Escenario de ocurrencia del fallo</b></p>	<p>El fallo se da, cuando el usuario final desea ingresar a la plataforma y descargar la información del paciente, la cual se encuentra alojada allí, si al ingresar se presenta un fallo de diseño respecto a la especificación del lenguaje de programación presentara error.</p> <div><div>fallo</div><div>componente, diseño</div><div>error</div><div>avería, defecto</div></div> <p>Diagrama 8. Escenario de ocurrencia del fallo por diseño</p>
<p><b>Flujo de corrección del Error</b></p>	<div><p>The screenshot displays the 'CENTRO DE SALUD MULTISERVICIOS' interface. It features a table with patient information. The table has columns for Nombre, Apellido, Identificación, Ciudad, Dirección, Número de contacto, Ocupación, Correo Electrónico, Tipo, Diagnóstico, and Tratamiento. Below the table, there is a footer with copyright information: 'Copyright © Todos los derechos reservados. El presente software fue desarrollado en Colombia'.</p></div> <p>Figura 6. Interfaz inicial del sistema</p>

Aquí se presenta la información del paciente al usuario final el cual podrá visualizar los datos ingresados, su diagnóstico y el tratamiento a seguir .

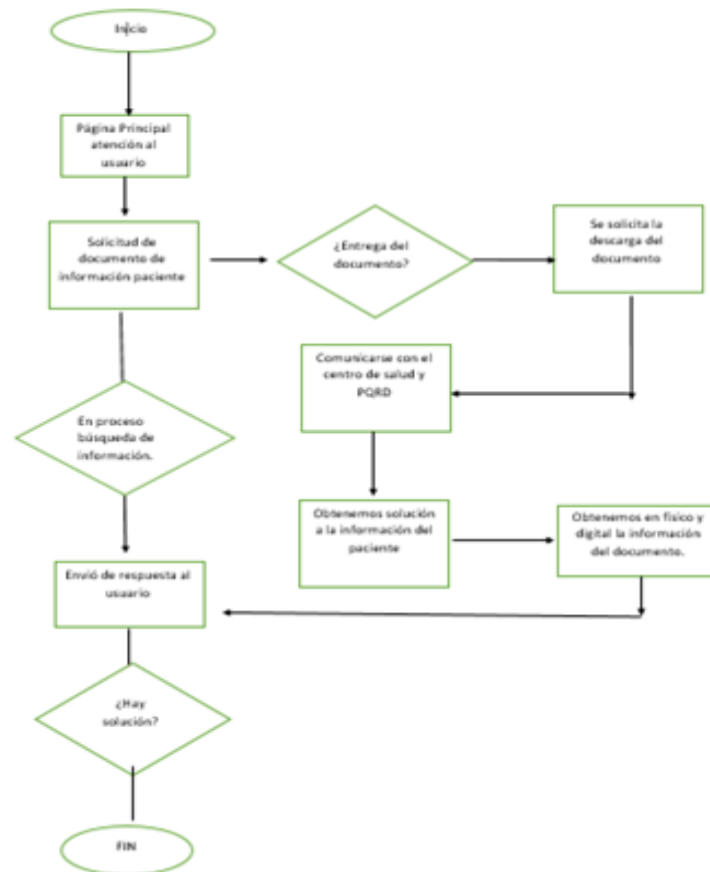


Diagrama 9. Flujo de corrección del error en el diseño

## TRANSACCIONES DISTRIBUIDAS

### Herramientas utilizadas

Se utilizaron diferentes herramientas de software para el desarrollo del aplicativo sobre transacciones distribuidas. Estas herramientas son:

- Lenguaje de programación: Hypertext Preprocessor (PHP), se utilizó para hacer la conexión a la base de datos remota, la base de datos local, la interfaz principal y la transacción distribuida con las tablas de las bases de datos a las que se accedió. También en este lenguaje se dió acceso y se envió la información del log de error cuando alguna de las bases de datos reporta fallo o inconsistencia en los datos ingresados.
- IDE:
  - Visual Studio Code: se utilizó como editor de texto para escribir el código php y html de la interfaz del programa.
  - Notepad: Se utilizó para crear el archivo de texto plano donde se guarda el log de error cuando se reporta un fallo en alguna de las bases de datos y también cuando no se presentan fallos.
- Herramientas:
  - Postman: Se utilizó como extensión del navegador Google Chrome para el envío de peticiones HTTP REST.
  - Wampp server: Se utilizó como servidor de aplicaciones para almacenar el código de la aplicación de la interfaz. También para implementar PHP y Mysql.
  - LiteSpeed: Se utilizó como servidor de aplicaciones web en la nube para almacenar los archivos de conexión y código de la aplicación. Se accedió a través de una URL proporcionada por el mismo para concatenar las variables locales de la información con la dirección de la base de datos.
  - Cpanel: Se utilizó como panel de control para administrar el servidor de alojamiento web de la base de datos que se encontraba en la nube. También se utilizó para almacenar el localhost de la base de datos remota.

- Mysql: Se utilizó como sistema gestor de bases de datos para almacenar la información con sql y también hacer la inserción de datos en las tablas creadas. Se crearon bases de datos con sus tablas en la máquina local y luego se subieron a la nube para acceder a ellas remotamente.
- phpMyAdmin: Se utilizó como herramienta para manejar la administración de MySQL en la aplicación, esta herramienta también está escrita en lenguaje PHP.
- Sistemas operativos:  
Las dos máquinas utilizadas tienen sistema operativo Windows 10 pro.
- Componentes físicos utilizados:
  - Computador portátil DELL
  - Computador portátil Asus

## Configuración del entorno del proyecto

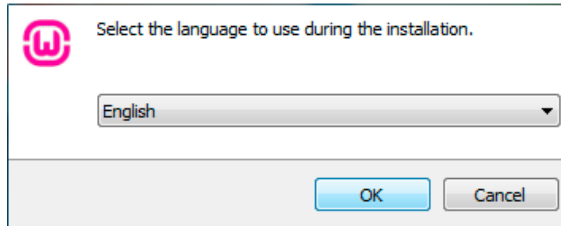
A continuación, se muestra la instalación del servidor Wampp. Este servidor contiene el servidor de aplicaciones Apache y el gestor de bases de datos Mysql que fueron los utilizados en el desarrollo del taller:

- Descarga del instalador en la página oficial:



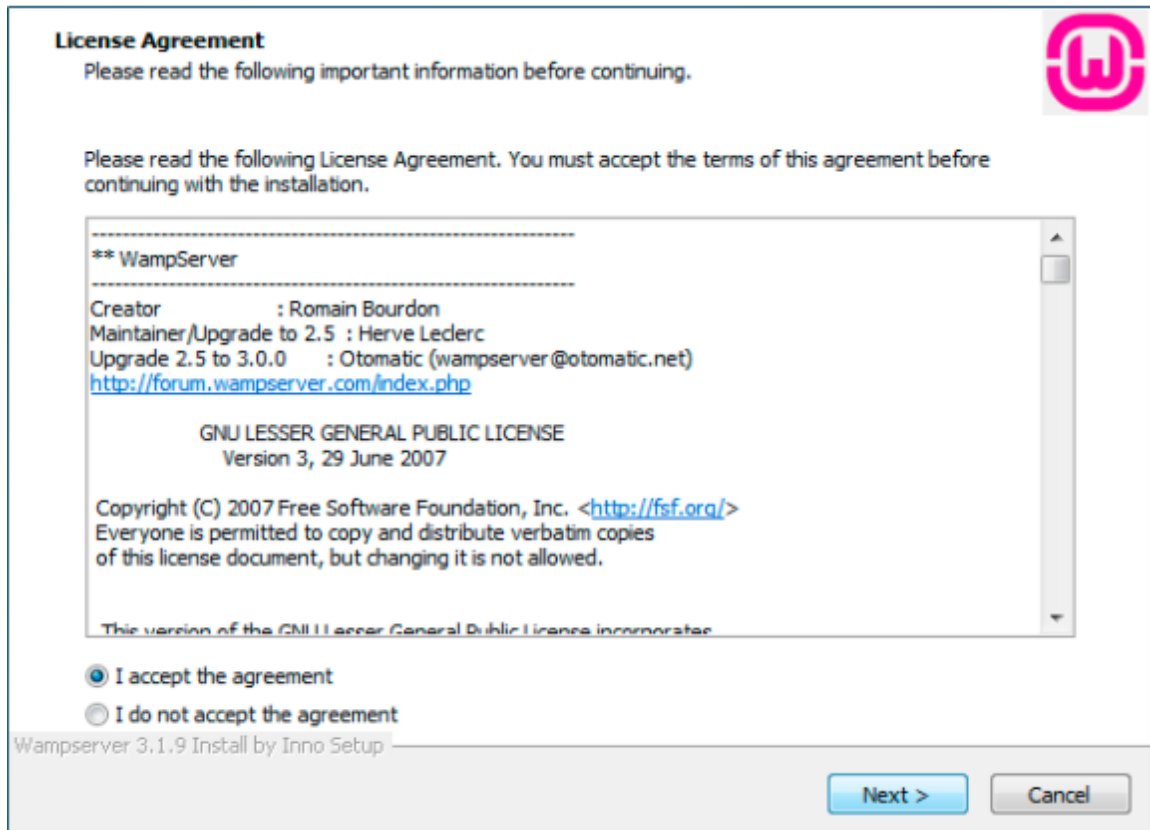
*Figura 7. Entorno de descarga del instalador de WampServer*

- Elección del idioma:



*Figura 8. Elección del idioma durante la instalación de WamppServer*

- Aceptar el acuerdo de licencia:



*Figura 9. Paso de aceptar las políticas de WamppServer*

- Continuar con la instalación:

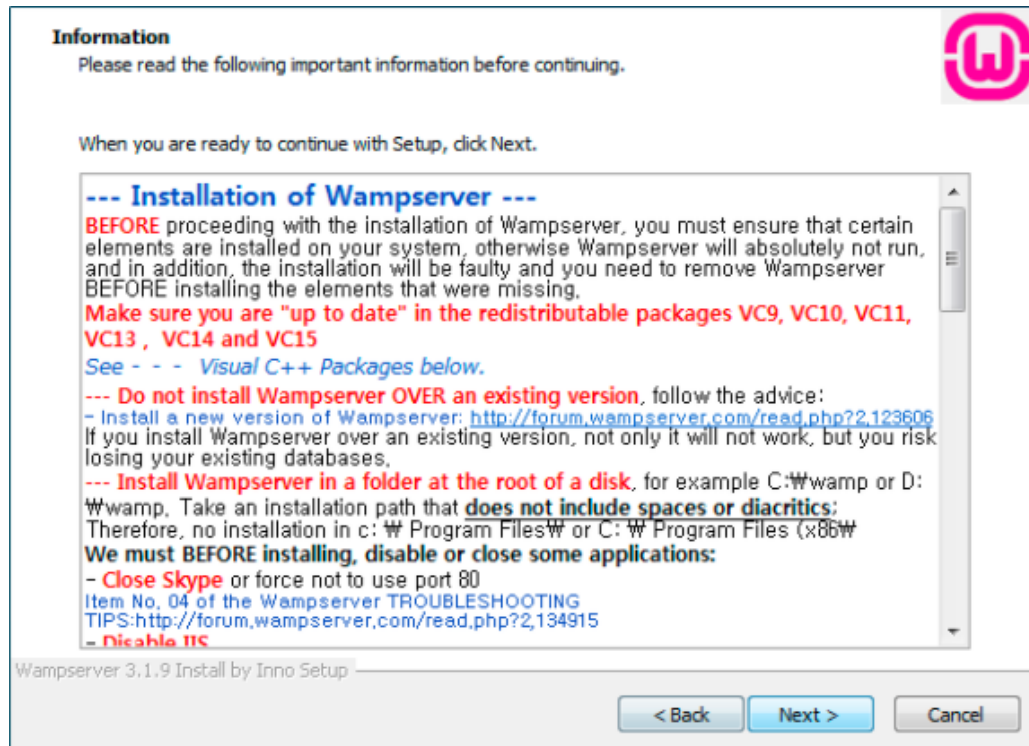


Figura 10. Condiciones de instalación de WampServer

- Elección de la ubicación:

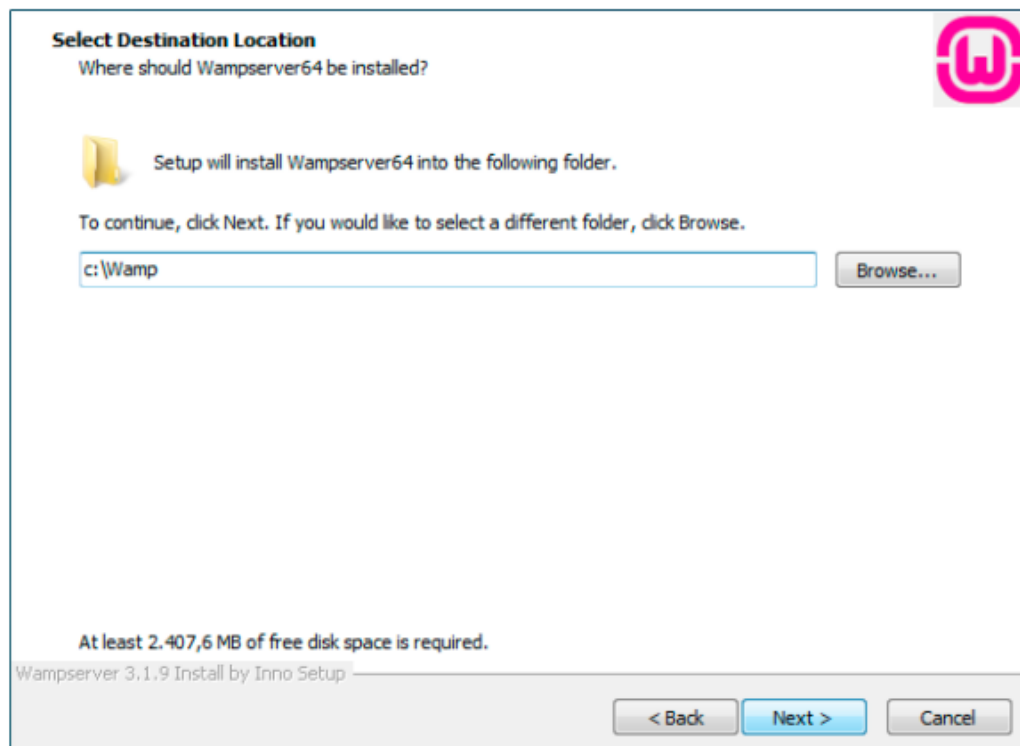
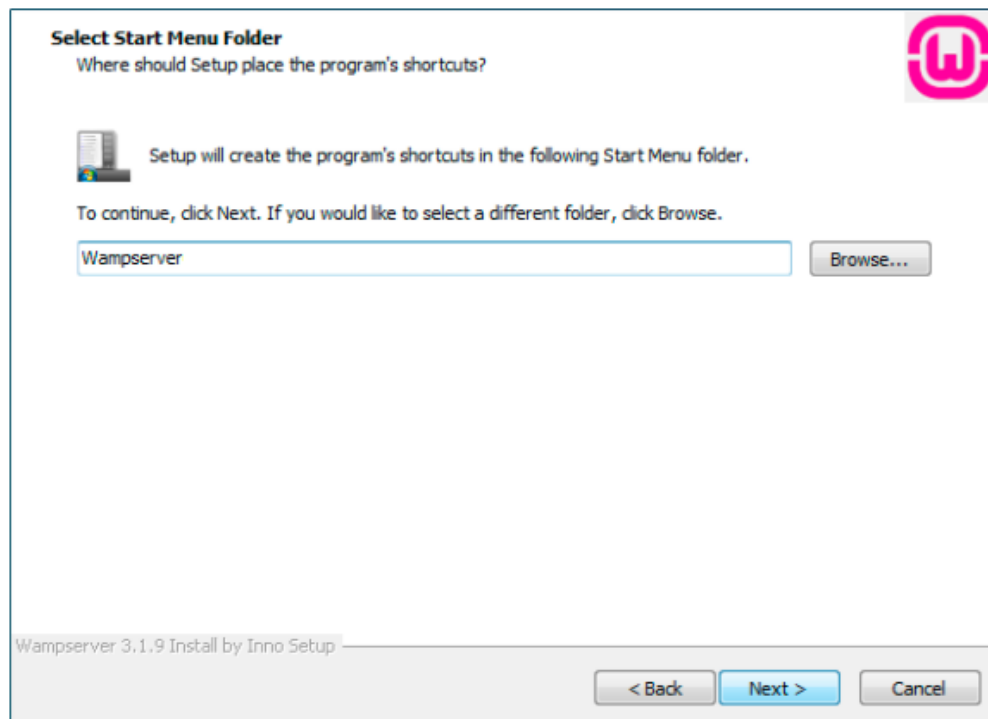


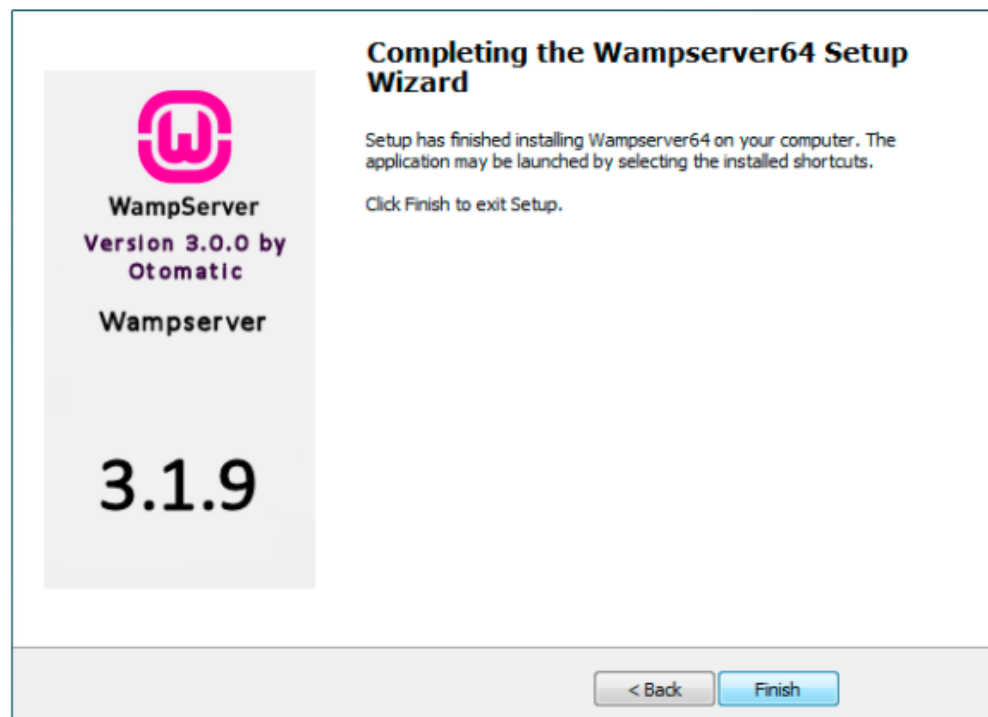
Figura 11. Elección de la ubicación de WampServer

- Elección del menú de inicio:



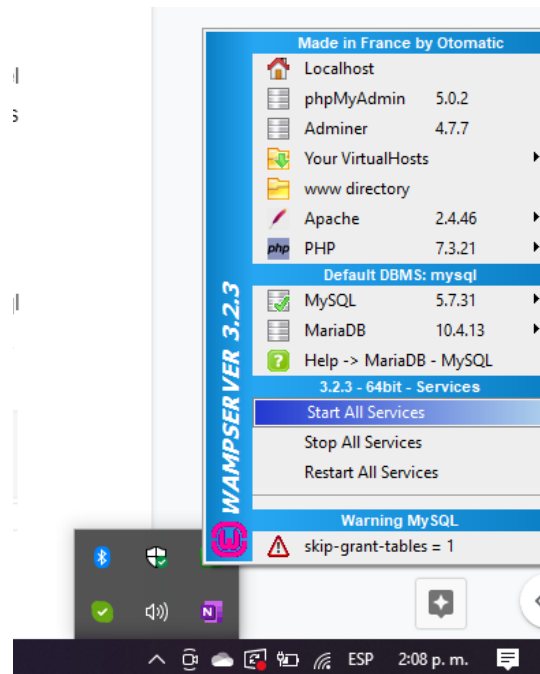
*Figura 12. Selección de la ubicación del menú de inicio*

- Finalización de la instalación:



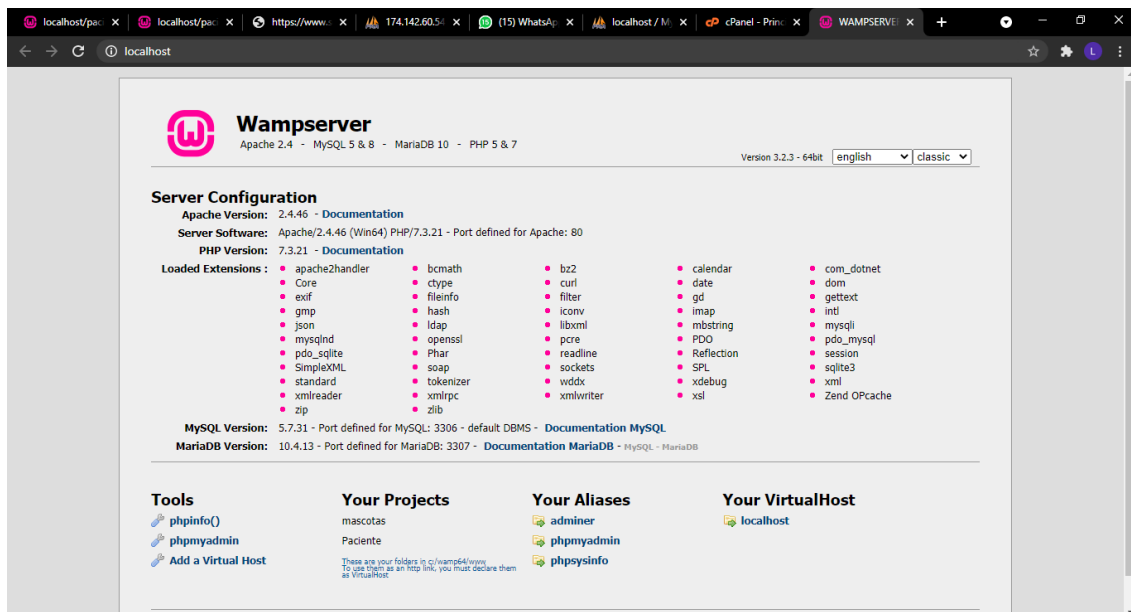
*Figura 13. Finalización de la instalación de WamppServer*

- Inicialización del servidor, en esta parte se inicia el servidor de aplicaciones que es apache y el gestor de base de datos que es MySQL con la herramienta localhost:



*Figura 14. Inicialización de WampServer en el menú de inicio*


- Página inicial de WampServer con el localhost del servidor (puerto 8080):



*Figura 15. Entorno inicial de WampServer*

- Login para ingresar a la herramienta phpMyAdmin:





Bienvenido a phpMyAdmin

Idioma - Language

Español - Spanish

Iniciar sesión

Usuario: root

Contraseña: .....

Elección del servidor: MySQL

Continuar

Figura 16. Ingreso con credenciales al phpMyAdmin

- Sitio de phpMyAdmin donde se va a manejar la administración de las bases de datos que hay en MySQL:

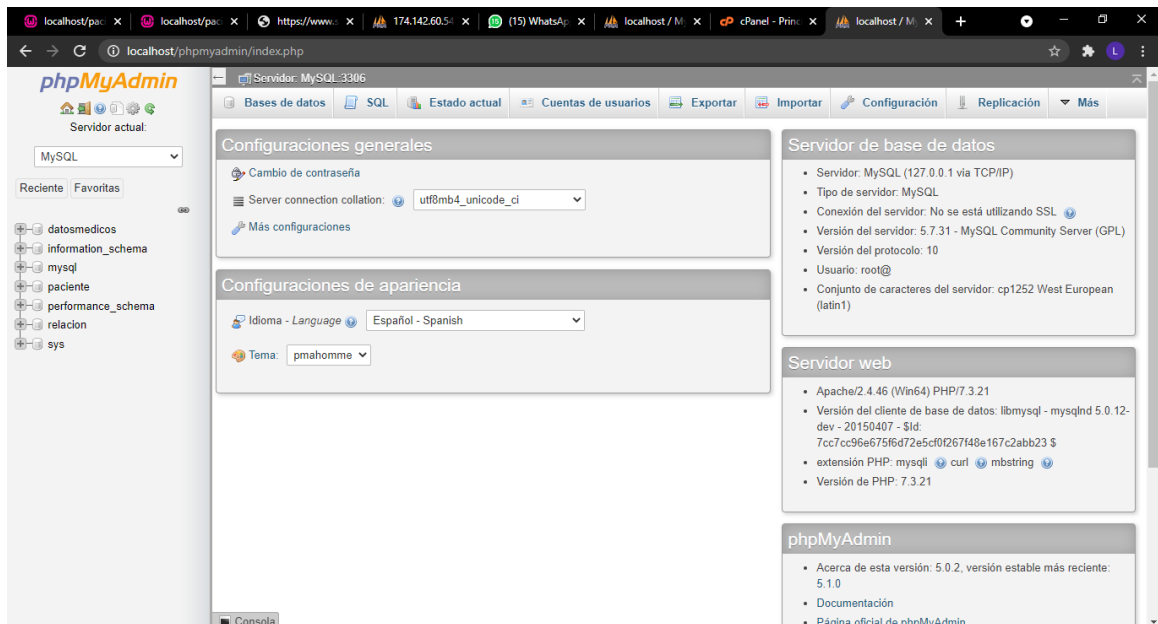


Figura 17. Interfaz de gestión de las bases de datos

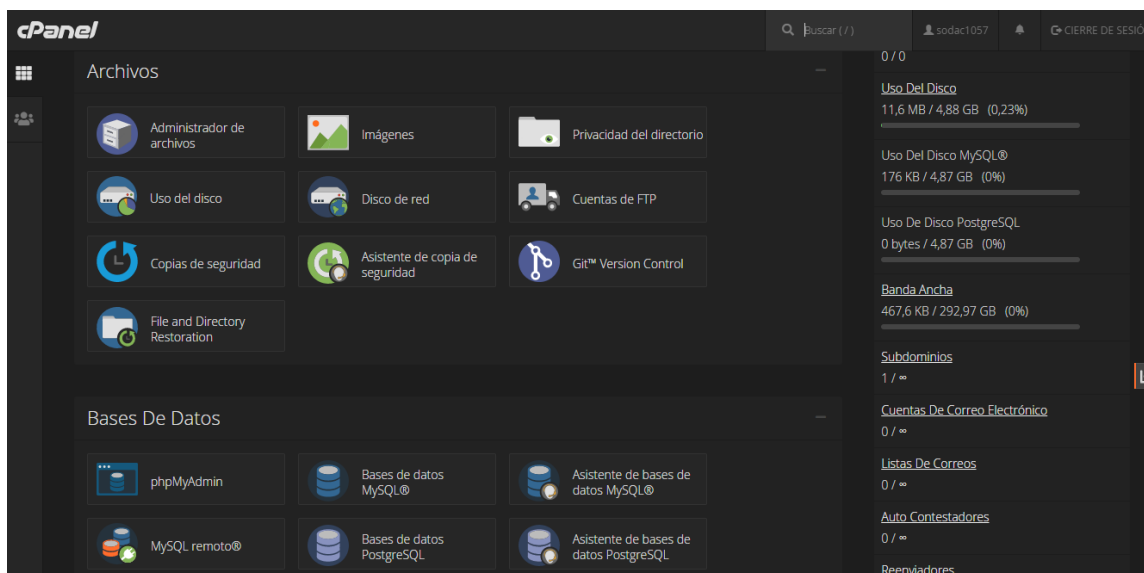
En el servidor LiteSpeed de alojamiento web se creó un usuario para subir los archivos de la aplicación. A su vez en la herramienta cPanel sirvió para gestionar el servidor y el localhost de la base de datos a la que se accede remotamente. Para poder utilizar el servidor es necesario comprar el dominio. A continuación se muestra el acceso al proyecto remoto:

- Ingreso al usuario de cPanel:



*Figura 18. Ingreso a la herramienta cPanel con las credenciales otorgadas*

- Entorno de almacenamiento de cPanel:



*Figura 19. Gestor de archivos de cPanel*

- Acceso al phpMyAdmin de manera remota por medio del servidor:

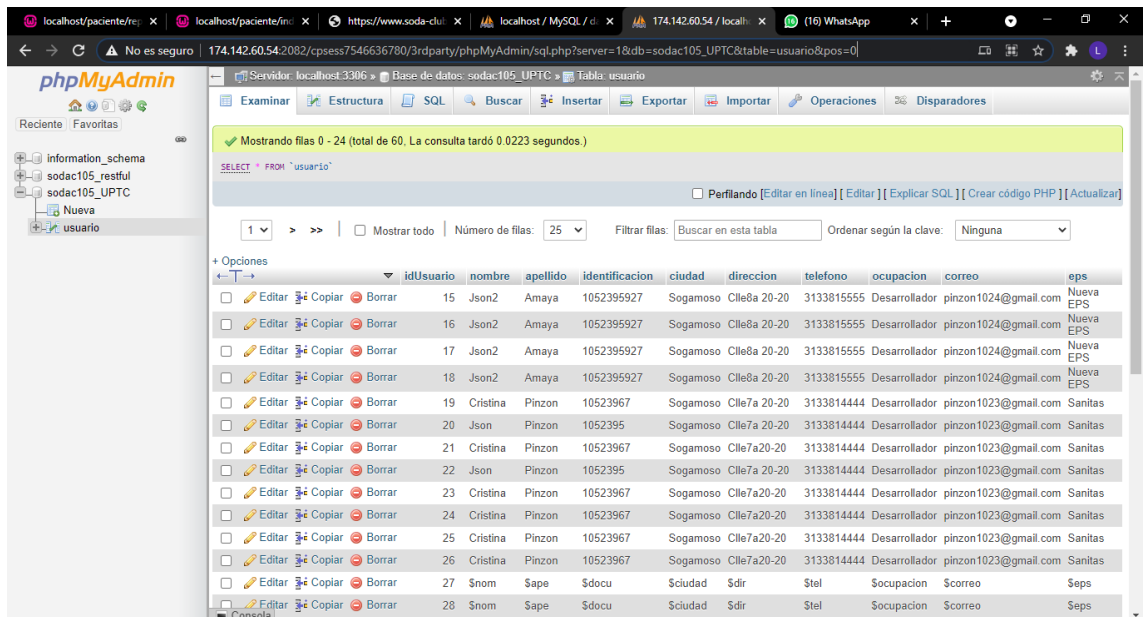


Figura 20. Gestor de bases de datos con la tabla creada para pacientes

- Acceso a los archivos en cPanel:

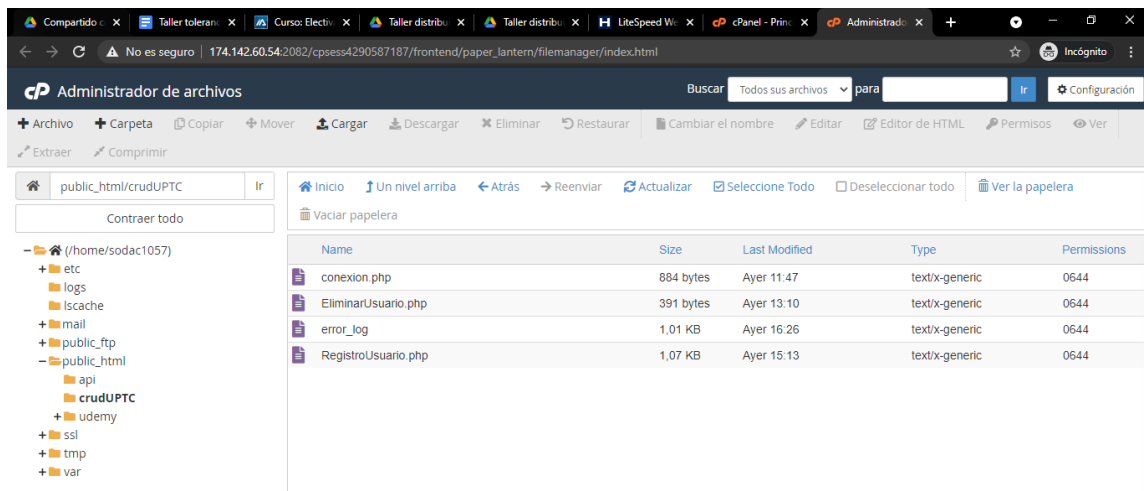


Figura 21. Almacenamiento de archivos en cPanel

## Políticas de tolerancia a fallos

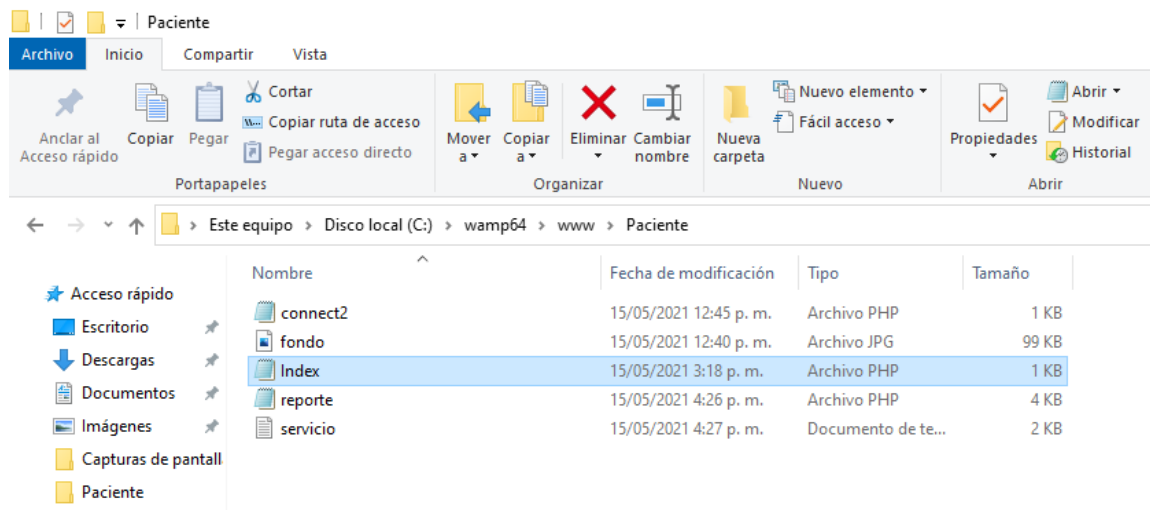
Para la tolerancia de fallos en la transacción distribuida que se desarrolló, se implementó la estrategia de enmascaramiento. Por medio de código php se verificó que los datos se enviaran consistentemente primero en la base de datos local y luego en la base de datos a la que se accede remotamente. Con la confirmación de envío exitoso o erróneo de la información a las dos bases de datos se diseñó un archivo de error o log

de error con los datos de la transacción exitosa o fallida indicando la fecha, hora, descripción de la operación, si es fallida o no y la descripción de la falla en caso de que sucediera. De esta forma, en caso de que el registro de los datos no haya sido exitoso se anula la operación y se tolera el fallo de información inconsistente o incompleta.

Por otra parte, se eliminó el fallo de caída de la base de datos por pérdida de la ip, ya que en este caso la ip es general y el funcionamiento no depende de la misma.

## Pasos del desarrollo del proyecto

1. Para el entorno general del proyecto se instala WamppServer para utilizar Mysql y se utilizó como servidor de aplicaciones para subir la carpeta raíz del proyecto.



*Figura 22. Almacenamiento del proyecto en la máquina local*

2. Se hicieron dos bases de datos en diferentes máquinas con mysql y la herramienta de administración phpMyAdmin. Una se alojó localmente en el disco c de la máquina, a la que se accedió con el localhost y la otra se alojó en el servidor de alojamiento LiteSpeed con ayuda del panel de control cPanel y se accedió a ella remotamente.

- Base de datos local:

The screenshot shows the phpMyAdmin interface for a local MySQL database named 'datosmedicos'. The selected table is 'historiaclinica'. The table structure is as follows:

Id	Diagnostico	Tratamiento
1	Paciente con dolor abdominal	Se le asigna una ecografia
2	Paciente con dolor pélvico y perineal	Se solicita ecografia pélvica
3	Paciente con cefálea crónica	Se remite a medicina especializada
4	Paciente que refiere colon irritable	Se remite a medicina especializada con gastroenter...
5	Paciente con irritación en rostro y manos	Se remite a medicina especializada con dermatologi...
6	Paciente con dolor en garganta	Se da tratamiento con hidrocodeina
7	Paciente con presión ocular anormal	Se remite a oftalmología con exámenes oculares
8	Paciente con artritis	Se envía tratamiento con alendronato, 1 cada mes
9	dolor	acetaminofen
10	dolor	acetaminofen
11	dolor	acetaminofen
12	dolor	acetaminofen
13	dolor	acetaminofen
14	dolor	acetaminofen

Figura 23. Tabla creada en la base de datos local

- Base de datos en alojamiento web:

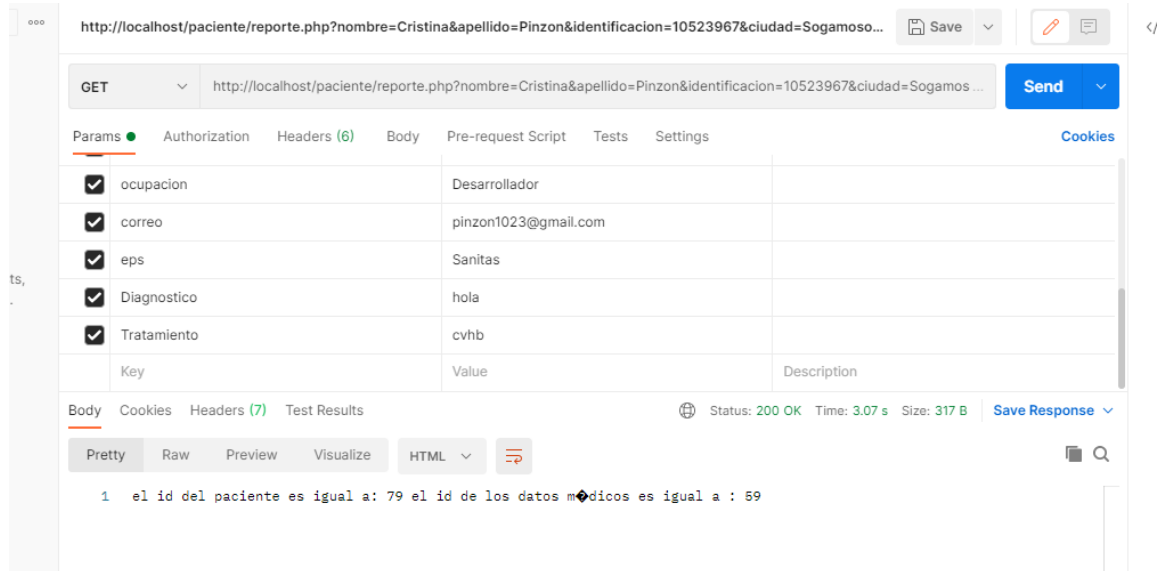
The screenshot shows the phpMyAdmin interface for a remote MySQL database named 'sodac105\_UPTC'. The selected table is 'usuario'. The table structure is as follows:

idUsuario	nombre	apellido	identificacion	ciudad	direccion	telefono	ocupacion	correo	eps
15	Json2	Amaya	1052395927	Sogamoso	Clle8a 20-20	3133815555	Desarrollador	pinzon1024@gmail.com	Nueva EPS
16	Json2	Amaya	1052395927	Sogamoso	Clle8a 20-20	3133815555	Desarrollador	pinzon1024@gmail.com	Nueva EPS
17	Json2	Amaya	1052395927	Sogamoso	Clle8a 20-20	3133815555	Desarrollador	pinzon1024@gmail.com	Nueva EPS
18	Json2	Amaya	1052395927	Sogamoso	Clle8a 20-20	3133815555	Desarrollador	pinzon1024@gmail.com	Nueva EPS
19	Cristina	Pinzon	10523967	Sogamoso	Clle7a 20-20	3133814444	Desarrollador	pinzon1023@gmail.com	Sanitas
20	Json	Pinzon	1052395	Sogamoso	Clle7a 20-20	3133814444	Desarrollador	pinzon1023@gmail.com	Sanitas
21	Cristina	Pinzon	10523967	Sogamoso	Clle7a20-20	3133814444	Desarrollador	pinzon1023@gmail.com	Sanitas
22	Json	Pinzon	1052395	Sogamoso	Clle7a 20-20	3133814444	Desarrollador	pinzon1023@gmail.com	Sanitas
23	Cristina	Pinzon	10523967	Sogamoso	Clle7a20-20	3133814444	Desarrollador	pinzon1023@gmail.com	Sanitas
24	Cristina	Pinzon	10523967	Sogamoso	Clle7a20-20	3133814444	Desarrollador	pinzon1023@gmail.com	Sanitas
25	Cristina	Pinzon	10523967	Sogamoso	Clle7a20-20	3133814444	Desarrollador	pinzon1023@gmail.com	Sanitas
26	Cristina	Pinzon	10523967	Sogamoso	Clle7a20-20	3133814444	Desarrollador	pinzon1023@gmail.com	Sanitas
27	\$nom	\$ape	\$docu	\$ciudad	\$dir	\$tel	\$ocupacion	\$correo	\$eps
28	\$nom	\$ape	\$docu	\$ciudad	\$dir	\$tel	\$ocupacion	\$correo	\$eps

Figura 24. Tabla creada en la base de datos remota

3. Para hacer el testeo del proyecto se instaló la herramienta postman. En esta herramienta se hicieron las pruebas de envío de peticiones a las bases de datos

enviando los parámetros por URL para probar los datos enviados y generar fallos en las bases de datos.



*Figura 25. Pruebas de URL en postman*

- Con la herramienta notepad se creó un archivo de texto plano (extensión.txt) en el que se almacenó el log de descripción de eventos sucedidos con la fecha, operacion, descripción de la operación, si falla o si no y la descripción de la falla si sucede.



*Figura 26. Archivo de log*

## Diagrama de arquitectura de solución

### Diagrama de distribución

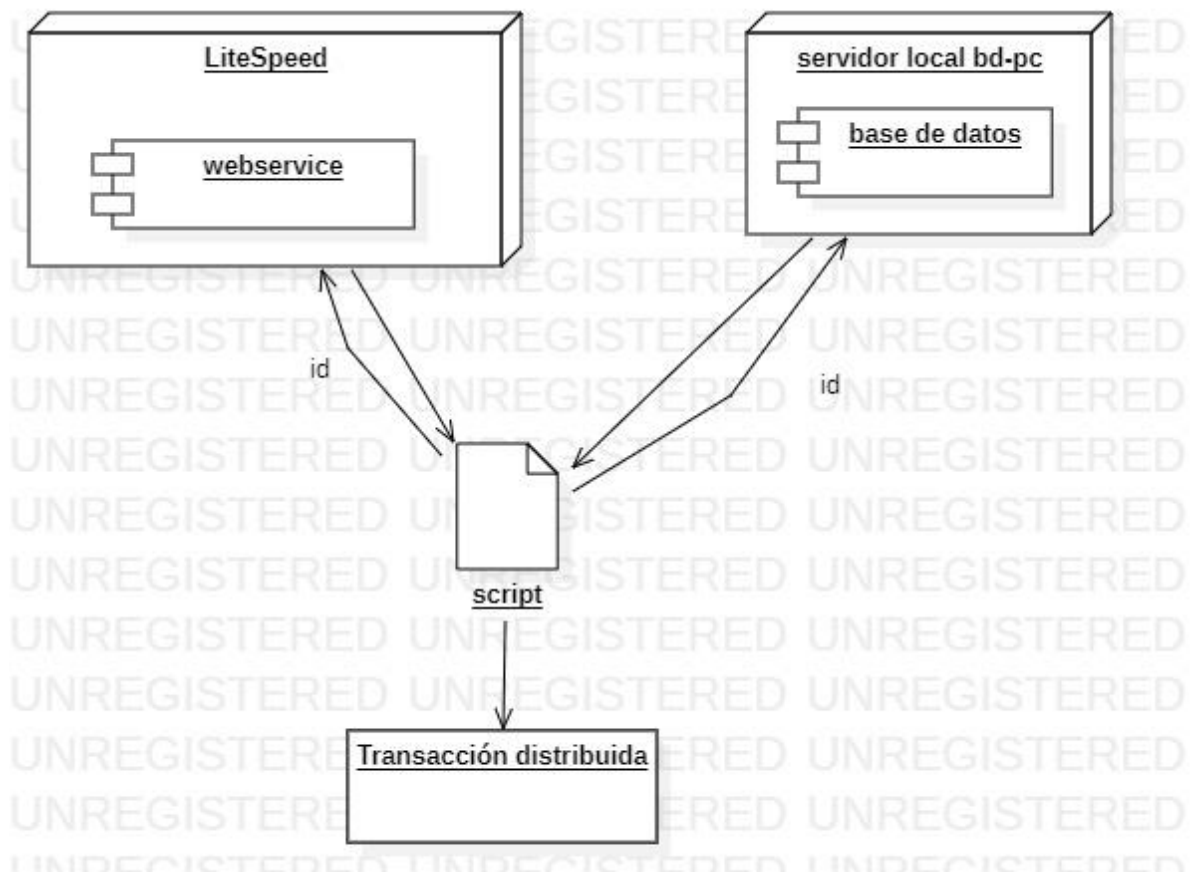
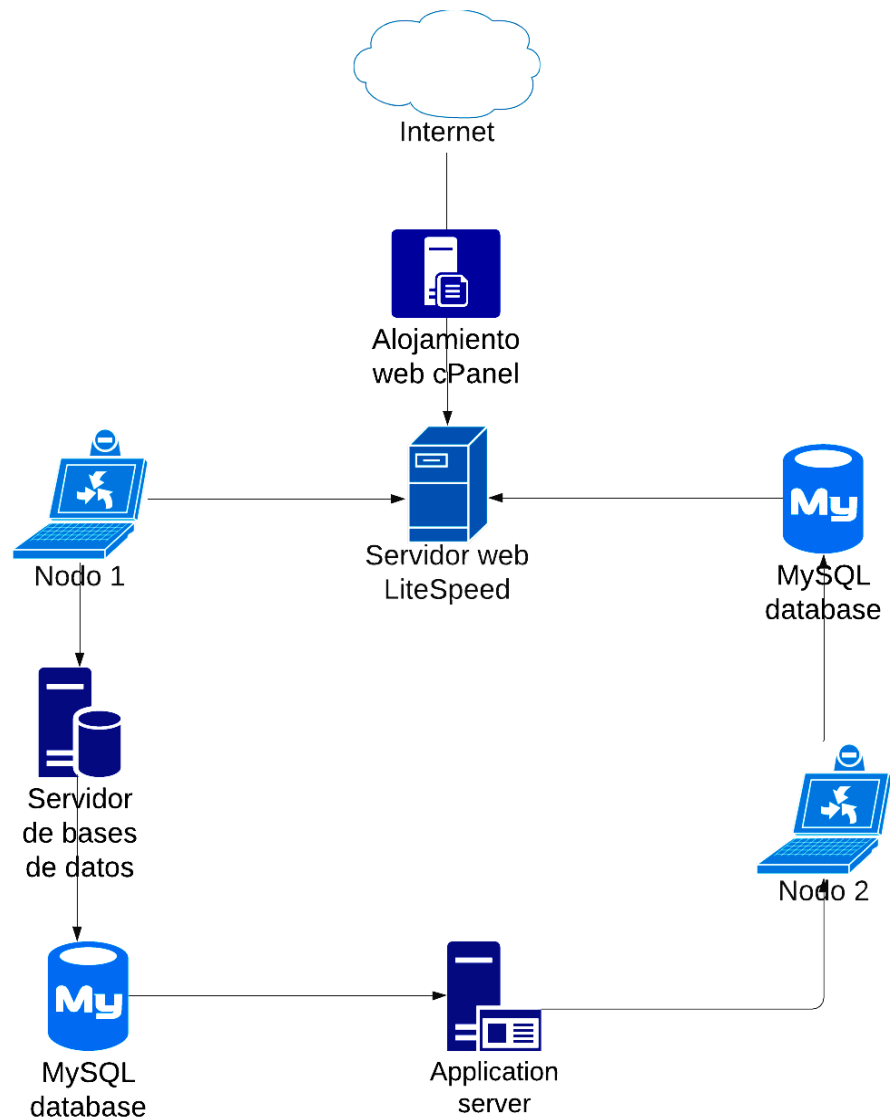


Diagrama 10. Diagrama de distribución en la transacción distribuida

## Diagrama de red



*Diagrama 11. Diagrama de la arquitectura en la transacción distribuida*

### Explicación del diagrama

Para la conexión en la nube:

- Nodo 1:
  - Datos físicos: Pc Dell
  - 4Gb de Ram
  - Procesador: Intel(R) Core(™)i3-1005G1 CPU 1.20GHz 1.19 GHz
  - Sistema operativo: Windows 10 pro



Tipo de sistema: Sistema operativo de 64 bits, procesador x64  
Middleware: servidor de aplicaciones Apache  
gestor de bases de datos MySql

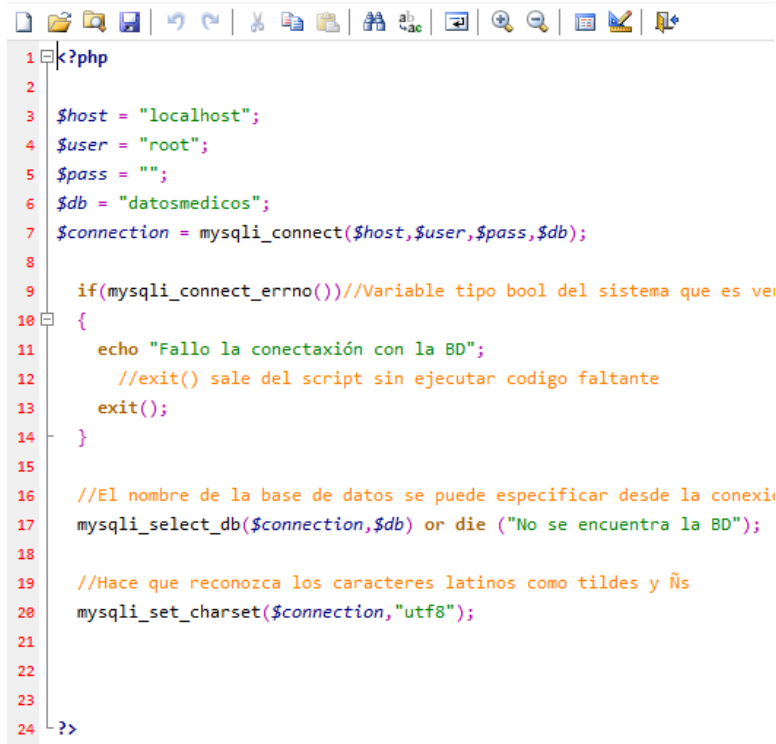
- Nodo 2:

Datos físicos: Pc Asus  
4Gb de Ram  
Procesador: Intel(R) Core(™)i5-7200 CPU 2.50GHz-2.71GHz  
Sistema operativo: Windows 10 pro  
Tipo de sistema: Sistema operativo de 64 bits, procesador x64  
Middleware: servidor de aplicaciones LiteSpeed  
gestor de bases de datos MySql

En la arquitectura del programa se tienen dos máquinas. Una tiene alojada la base de datos en el disco local c en el sistema gestor de bases de datos MySQL con ayuda de WamppServer. También utilizando WamppServer están almacenados los archivos de la aplicación en el servidor Apache. La base de datos que se encuentra en esta máquina tiene datos personales de pacientes en una tabla que luego van a ser relacionados con los datos médicos que se encuentran en la base de datos que está en la nube.

La otra máquina implementa un servidor en la nube. En este servidor se encuentra el phpMyAdmin donde está alojada la segunda base de datos. Con la url que redirecciona a la tabla de la base de datos correspondiente se permite ingresar la información respectiva a la misma.

En cada máquina se hizo un archivo de conexión php que envía los datos del servidor correspondiente: el host, usuario, contraseña, base de datos y se crea la variable de conexión. En este archivo se valida si se accedió exitosamente a la base de datos o no. A continuación se muestra el archivo de conexión con la base de datos que se encuentra en la máquina local.



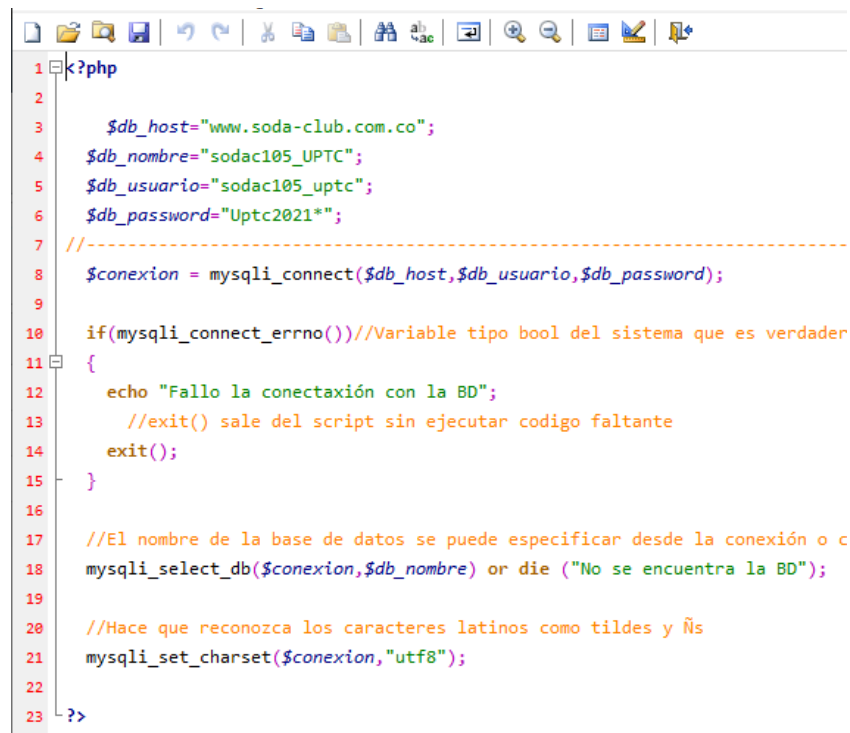
```

1 <?php
2
3 $host = "localhost";
4 $user = "root";
5 $pass = "";
6 $db = "datosmedicos";
7 $connection = mysqli_connect($host,$user,$pass,$db);
8
9 if(mysqli_connect_errno())//Variable tipo bool del sistema que es ver
10 {
11     echo "Fallo la conectación con la BD";
12     //exit() sale del script sin ejecutar codigo faltante
13     exit();
14 }
15
16 //El nombre de la base de datos se puede especificar desde la conexi
17 mysqli_select_db($connection,$db) or die ("No se encuentra la BD");
18
19 //Hace que reconozca los caracteres latinos como tildes y Ñs
20 mysqli_set_charset($connection,"utf8");
21
22
23
24 ?>

```

*Figura 27. Archivo de conexión de la información con la base de datos local*

El siguiente es el archivo de conexión con la base de datos que está en la nube.



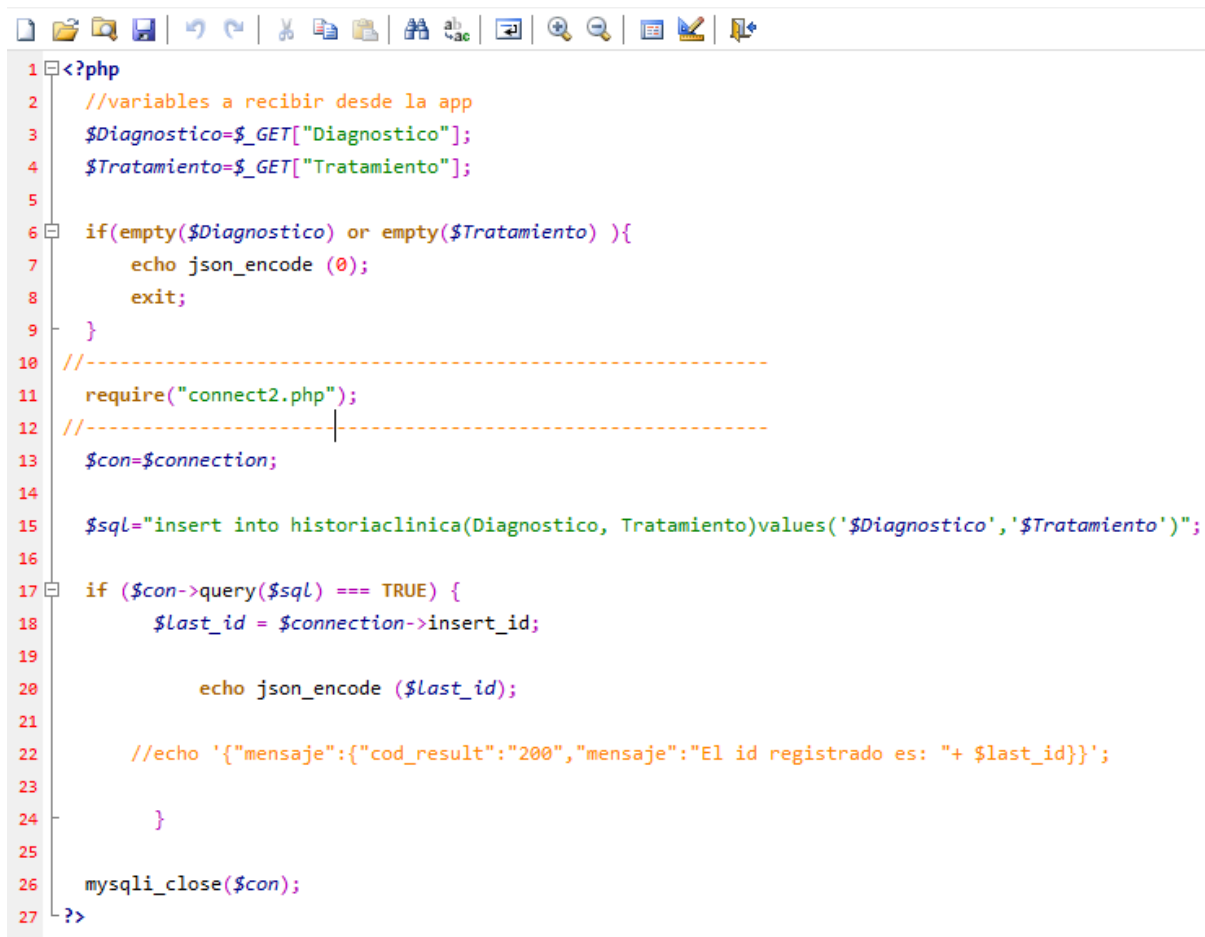
```

1 <?php
2
3     $db_host="www.soda-club.com.co";
4     $db_nombre="sodac105_UPTC";
5     $db_usuario="sodac105_uptc";
6     $db_password="Uptc2021*";
7     //-----
8     $conexion = mysqli_connect($db_host,$db_usuario,$db_password);
9
10 if(mysqli_connect_errno())//Variable tipo bool del sistema que es verdadero
11 {
12     echo "Fallo la conectación con la BD";
13     //exit() sale del script sin ejecutar codigo faltante
14     exit();
15 }
16
17 //El nombre de la base de datos se puede especificar desde la conexión o c
18 mysqli_select_db($conexion,$db_nombre) or die ("No se encuentra la BD");
19
20 //Hace que reconozca los caracteres latinos como tildes y Ñs
21 mysqli_set_charset($conexion,"utf8");
22
23
24 ?>

```

*Figura 28. Código de conexión con la base de datos remota*

También en las dos máquinas se creó un archivo donde se pasan las variables que va a recibir la tabla de cada respectiva base de datos. En la máquina local se pasaron los datos médicos del paciente que son el diagnóstico y el tratamiento. En la otra máquina se pasaron los datos personales del paciente que son el nombre, identificación, apellido, ciudad, dirección telefono, ocupación, correo y eps. Se validó que los datos enviados fueran válidos y se llamó al archivo de conexión. Se realizó la consulta a la base de datos con el respectivo comando y se realizó la inserción. El siguiente es el contenido del documento de registro creado en la base de datos local.

A screenshot of a code editor window with a standard toolbar at the top. The editor displays PHP code for connecting to a database and inserting data. The code is numbered from 1 to 27 on the left margin. The code includes comments in Spanish, variable assignments for diagnostic and treatment data, a validation check for empty values, a connection to a database using 'connect2.php', an SQL insert statement, and a conditional execution of the query with JSON output and a success message.

```
1 <?php
2 //variables a recibir desde la app
3 $Diagnostico=$_GET["Diagnostico"];
4 $Tratamiento=$_GET["Tratamiento"];
5
6 if(empty($Diagnostico) or empty($Tratamiento) ){
7     echo json_encode (0);
8     exit;
9 }
10 //-----
11 require("connect2.php");
12 //-----
13 $con=$connection;
14
15 $sql="insert into historiaclinica(Diagnostico, Tratamiento)values('$Diagnostico','$Tratamiento')";
16
17 if ($con->query($sql) === TRUE) {
18     $last_id = $connection->insert_id;
19
20     echo json_encode ($last_id);
21
22     //echo '{"mensaje":{"cod_result":"200","mensaje":"El id registrado es: "+ $last_id}}';
23
24 }
25
26 mysqli_close($con);
27 ?>
```

*Figura 29. Código de inserción de la información en la base de datos local*

El siguiente es el contenido del archivo de registro de datos en la máquina a la que se accedió remotamente.

```

1 <?php
2 //variables a recibir desde la app
3 $nom=$_GET["nombre"];
4 $ape=$_GET["apellido"];
5 $docu=$_GET["identificacion"];
6 $ciudad=$_GET["ciudad"];
7 $dir=$_GET["direccion"];
8 $tel=$_GET["telefono"];
9 $ocupacion=$_GET["ocupacion"];
10 $correo=$_GET["correo"];
11 $eps=$_GET["eps"];
12
13 if(empty($nom) or empty($ape) or empty($docu) or empty($ciudad) or empty($dir) or empty($tel) or empty($ocupacion) or empty($correo) or empty($eps)){
14     echo json_encode(0);
15     exit;
16 }
17
18
19 //-----
20 require("conexion.php");
21 //-----
22 $con=$conexion;
23
24 $sql="insert into usuario(nombre,apellido,identificacion,ciudad,direccion,telefono,ocupacion,correo,eps)values('$nom','$ape','$docu','$ciudad','$dir','$tel','$ocupacion','$correo','$eps')";
25
26 if ($con->query($sql) == TRUE) {
27     $last_id = $con->insert_id;
28
29     echo json_encode ($last_id);
30
31     //echo '{"mensaje":{"cod_result":"200","mensaje":"El id registrado es: "+ $last_id}}';
32
33 }
34
35 mysqli_close($con);
36 >>

```

*Figura 30. Código de inserción de la información en la base de datos remota*

Finalmente se creó el archivo de reporte donde se validó que el registro de los datos que se envían por la URL sea exitoso en las dos bases de datos. Para esto se confirmó que los datos estuvieran completos. Primero en la base de datos remota y luego en la base de datos local. En este documento se creó una función que envía el reporte del estado de la petición enviada donde se registra la fecha, la hora, la operación realizada, la descripción de la operación, si hubo falla o si no y la descripción de la falla en caso de que esta ocurra. Estos datos son almacenados en un archivo de texto plano con extensión .txt en el que se muestra el estado de cada petición realizada. El siguiente es el archivo creado de envío y validación de los datos a las dos bases de datos.

```

1 <?php
2 $nom=$_GET["nombre"];
3 $ape=$_GET["apellido"];
4 $docu=$_GET["identificacion"];
5 $ciudad=$_GET["ciudad"];
6 $dir=$_GET["direccion"];
7 $tel=$_GET["telefono"];
8 $ocupacion=$_GET["ocupacion"];
9 $correo=$_GET["correo"];
10 $eps=$_GET["eps"];
11 $Diagnostico=$_GET["Diagnostico"];
12 $Tratamiento=$_GET["Tratamiento"];
13 //Insertar info del usuario
14 $idFase1 = file_get_contents("https://www.soda-club.com.co/crudUPTC/RegistroUsuario.php?nombre=$nom&apellido=$ape&identificacion=$docu&ciudad=$ciudad&di
15 //Si no viene vacío el idFase1, continuamos con el registro de los datos médicos del paciente
16 if($idFase1!=0){
17     //echo "Registro fase 1 exitoso";
18     $idFase2 = file_get_contents("http://localhost/paciente/index.php?Diagnostico=$Diagnostico&Tratamiento=$Tratamiento");
19
20     //Si falla el registro de datos médicos, eliminamos el registro creado en la fase 1 --> rollback
21     if($idFase2==0){
22         registrarLog("Registro fase 2 erróneo, por lo que se elimina el registro de la fase 1 ", true, " No se completó la transacción exitosamente porque r
23         echo "Registro fase 2 erróneo, por lo que se elimina el registro de la fase 1";
24         //Eliminar lo agregado en la fase 1 si viene vacía la fase 2
25
26         $msjResultado = file_get_contents("https://www.soda-club.com.co/crudUPTC/EliminarUsuario.php?idUsuario=$idFase1");
27         exit;
28     }
29     registrarLog("Ambos registros realizados exitosamente con los id's $idFase1:$idFase2 ", false, " N/A ", " Registro de usuario e historial médico\n");
30 }
31 echo " el id del paciente es igual a: $idFase1 el id de los datos médicos es igual a : $idFase2" ;

```

Figura 31. Código de validación de la información en el envío a la base de datos

Esta es la función creada para enviar el reporte al log (archivo de texto):

```

33 function registrarLog($descripcion, $esFalla, $descripcionFalla, $operacion){
34
35     $fecha = date("Y/m/d");
36     $hora = date("h:i:sa");
37     $registro = $fecha.$hora.$descripcion.$esFalla.$descripcionFalla.$operacion;
38
39     $myfile = fopen("servicio.txt", "r") or die("Unable to open file!");
40     $textoActual = fread($myfile,filesize("servicio.txt"));
41     fclose($myfile);
42     $myfile = fopen("servicio.txt", "w") or die("Unable to open file!");
43     $textoNuevo = $textoActual.$registro;
44     fwrite($myfile, $textoNuevo);
45     fclose($myfile);
46 }

```

Figura 32. Función que escribe y elimina en el archivo de texto sobre el log de estado de la transacción

Y así queda el log:

servicio.txt - Notepad2-mod

Archivo Editar Ver Configuración ?

1 2021/05/1509:14:23pmAmbos registros realizados exitosamente con los id's 68:51N/ARegistro de usuario e historial médico2021/05/  
2 2021/05/1509:17:47pmAmbos registros realizados exitosamente con los id's 72:55 N/A Registro de usuario e historial médico  
3 2021/05/1509:17:48pmAmbos registros realizados exitosamente con los id's 73:56 N/A Registro de usuario e historial médico  
4 2021/05/1509:17:51pmAmbos registros realizados exitosamente con los id's 74:57 N/A Registro de usuario e historial médico  
5 2021/05/1509:21:38pmAmbos registros realizados exitosamente con los id's 75:0 N/A Registro de usuario e historial médico  
6 2021/05/1509:22:17pmAmbos registros realizados exitosamente con los id's 76:0 N/A Registro de usuario e historial médico  
7 2021/05/1509:26:20pmRegistro fase 2 erróneo, por lo que se elimina el registro de la fase 1 1 No se completó la transacción exi  
8 2021/05/1509:27:06pmAmbos registros realizados exitosamente con los id's 78:58 N/A Registro de usuario e historial médico  
9 2021/05/1606:08:45pmAmbos registros realizados exitosamente con los id's 79:59 N/A Registro de usuario e historial médico  
10

*Figura 34. Log de estado de la transacción*

## **Conclusiones**

1. Las aplicaciones web requieren de estrategias de tolerancia a fallos para cumplir correctamente con las especificaciones de funcionalidad del cliente.
2. Implementando el concepto de transacciones distribuidas se logró conectar dos máquinas con bases de datos almacenadas en cada una que luego se unieron, por lo que se facilitó la tarea de trabajo desde diferentes lugares.
3. Cuando se tiene una base de datos distribuida se debe implementar la tolerancia a fallos para asegurar que la información se almacene íntegramente.
4. Para tolerar fallos en la transacción distribuida de bases de datos se implementó el enmascaramiento por medio de un archivo de log de error.
5. Mediante estrategias de tolerancia a fallos el sistema se torna más seguro y confiable ya que si se tienen en cuenta los posibles fallos y hay estrategias para cada uno de ellos implementando una solución se previene al usuario con respecto a la funcionalidad del sistema y aumenta la confiabilidad para que el usuario manipule el sistema de una forma correcta y eficiente.