**PROGRAMACIÓN DE SISTEMAS TELEMÁTICOS**

**MANUAL TÉCNICO**

**21/01/2020, 2019-II**

Facultad de Ingeniería en Mecánica y Ciencias de la Producción (FIMCP)

Sánchez Acosta Juan Sebastián [juasesan@espol.edu.ec](mailto:juasesan@espol.edu.ec)

Bowen Alcívar David Sebastián dbowen@espol.edu.ec

Párraga Mendoza José Gabriel [jgparrag@espol.edu.ec](mailto:jgparrag@espol.edu.ec)

España Ayala Sofía del Rocío sespana@espol.edu.ec

Cedeño Barahona Ronald Adrián ronacede@espol.edu.ec

“Gestión de impresoras 3D en red”

**I. DESCRIPCIÓN DEL PROBLEMA**

Cuando se presenta un error en un proceso de impresión 3D, el usuario no tiene la capacidad de detener la operación si no se encuentra presente en el sitio donde se halla alojada la máquina, haciendo que no sea posible detener la operación de manera rápida para evitar pérdidas de material, tiempo y energía consumida en vano. Esto ocurre para la mayoría de impresoras de gama básica y media, dado que no poseen una tarjeta de red integrada en sus placas madres que les permita conectarse a una red donde se puedan operar de manera remota.

**II. SOLUCIÓN PROPUESTA**

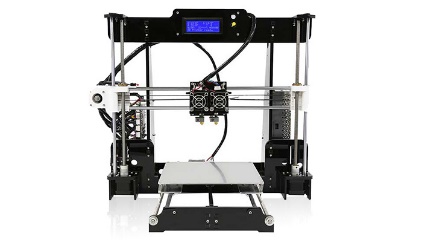
Se plantea el desarrollo de una aplicación móvil capaz de acceder a las redes donde se encuentren alojadas las impresoras 3D que se desean manipular, a través de una dirección ip. Para ello el usuario debe iniciar sesión en la app con su id y contraseña, previamente aprobados por el administrador de las impresoras. Hecho esto, podrá indicar la dirección ip de la red a la que desea acceder para luego visualizar las máquinas que se encuentran en la misma, así como su estado actual (disponible u ocupada). Finalmente, selecciona aquella que prefiera y envía el archivo con el modelo digital que desea imprimir para así comenzar la operación.

**III. FUNCIONAMIENTO DEL SISTEMA**

La solución propuesta funciona gracias al uso de una placa Raspberry Pi 3 que se encuentra conectada tanto a la impresora de destino, a través de un cable USB, como a la red donde esta se encuentra alojada, utilizando un cable de red o conexión wifi.

La placa Raspberry actúa como un ordenador, en el cual se encuentra instalado un servidor que accede a una base de datos SQL, extrae información y se la suministra a la impresora para que pueda realizar sus operaciones. Remotamente, desde un ordenador o la aplicación móvil creada, un usuario puede acceder a los controles de la impresora, ejecutar operaciones básicas (movimiento de los ejes, calibración, control de temperatura, etc.), y cargar archivos de modelos a imprimir almacenados en el dispositivo o en la DB. En este último caso, el servidor realiza un query en segundo plano hacia phpmyadmin para extraer los datos necesarios.

**IV. DIAGRAMAS DE LA SOLUCIÓN**



Red pública universitaria

Router Local

Router Local

Dispositivo móvil

Raspberry Pi 3

Impresora 3D

Acceso a la base de datos datos

Ilustración 1: diagrama de conexión física

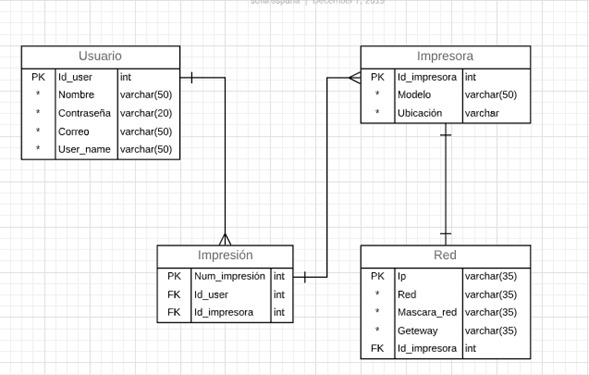


Ilustración 2: diagrama entidad-relación de la data base

**V. RECURSOS DE HARDWARE**

* Impresora 3D
* Cable USB
* Raspberry Pi 3
* Cable ethernet
* Enrutador
* Smartphone

**VI. RECURSOS DE SOFTWARE**

* Conexión a internet en la red local
* Aplicación móvil instalada en el smartphone
* Conexión wifi para el usuario remoto

**VII. IMPLEMENTACIÓN DEL PROYECTO**

1. Realizar la respectiva conexión de la Raspberry Pi 3 hacia la impresora de destino mediante un cable USB.
2. Revisar la conexión entre el ordenador y la red local.
3. Elaboración del servidor en la RP3, instalando el software en su unidad de almacenamiento.
4. Conectar la placa RP3 a la red en la que se va a trabajar mediante un cable de red o WiFi.
5. Anotar la dirección IP local de la RP3.
6. Ingresar la dirección IP en la barra de búsqueda del navegador e iniciar sesión.
7. Configurar la impresora 3D con sus respectivas características y establecer la conexión con esta.
8. Probar los controles de la impresora desde el navegador, así como los medidores de temperatura, nivelación de la cama caliente y velocidad de los ventiladores.
9. Realizar el paso 6 desde el dispositivo móvil mediante la aplicación previamente descargada e instalada.
10. Cargar el archivo a imprimir desde el dispositivo móvil, el cual puede encontrarse guardado en una base de datos o puede ser añadido por el usuario.
11. Revisar el funcionamiento de la impresora, la conexión de la webcam y, en caso de ser necesario, cancelar la impresión.

**VIII. DESCRIPCIÓN Y CÓDIGO DE BASE DE DATOS**

Se realizó un método para registrar los datos del usuario en la base de datos y por otra parte, se consultaba a la base de datos cada vez que un usuario desea ingresar para validar si existe o no.

En la siguiente parte del código para la base datos para las impresoras 3D, solamente se recoge la información depositada en la base de datos para mostrársela al usuario.

Finalmente, cuando se le da al botón imprimir se guarda el dato de la impresión dentro de una tabla de impresiones, dentro de la cual, cada fila representa a una impresión que tiene como parámetros al usuario que imprimió, a la impresora en la cual se imprimió, la fecha de impresión y el nombre del archivo.

**VIX. EXPLICACIÓN DEL CÓDIGO FUENTE**

**Función Ingresar Usuario:** Esta función tiene como finalidad generar una cuenta para aquellos usuarios que deseen registrarse en la plataforma, de manera que obtiene los datos ingresados de la persona (Nombre, Correo, Nombre de usuario, Contraseña), y los ubica dentro de la base de datos de registros.

**public void** insertarUsuario(View view) {  
 **try** {  
 String[] resultadoSQL = **null**;  
 datosConexion = **new** String[]{  
 serverIP,  
 port,  
 database,  
 userMySQL,  
 pwdMySQL,  
 **"SELECT \* FROM Usuario;"** };  
 String driver2 = **"com.mysql.jdbc.Driver"**;  
 Class.forName(driver2).newInstance();  
 resultadoSQL = **new** AsyncQuery().execute(datosConexion).get();  
 String resultadoConsulta2 = resultadoSQL[0];  
 String[] lista=resultadoConsulta2.split(**"\n"**);  
 lista[0]=**"Nombre,Contraseña,Correo,User\_name"**;  
 **int** contador=1;  
 **boolean** veridico=**true**;  
 **for**(String dato: lista){  
 **if**(contador!=1){  
 String[] listaCompleta= dato.split(**","**);  
 **if** ((listaCompleta[2].equals(corre.getText().toString()))||((corre.getText().toString()).equals(**""**))){  
 veridico=**false**;  
 }  
 }**else**{  
 contador++;  
 }  
 }  
  
 **if**(veridico){  
 String driver = **"com.mysql.jdbc.Driver"**;  
 Class.forName(driver).newInstance();  
  
 String[] nuevaCita = **new** String[]{  
 nomb.getText().toString(),  
 cont.getText().toString(),  
 corre.getText().toString(),  
 usu.getText().toString(),  
 serverIP,  
 port,  
 database,  
 userMySQL,  
 pwdMySQL,  
 };  
  
  
 **new** TareaAsincronaInsertar().execute(nuevaCita);  
  
 Toast.makeText(**this**,**"Cuenta creada con exito"**, Toast.LENGTH\_LONG).show();  
 nomb.setText(**""**);  
 cont.setText(**""**);  
 corre.setText(**""**);  
 usu.setText(**""**);  
 finish();  
  
 }**else**{  
 Toast.makeText(**this**,**"Correo o Usuario ya en uso o no ingreso ningun correo"**, Toast.LENGTH\_LONG).show();  
 }  
  
 } **catch** (IllegalAccessException e) {  
 e.printStackTrace();  
 } **catch** (InstantiationException e) {  
 e.printStackTrace();  
 } **catch** (ClassNotFoundException e) {  
 e.printStackTrace();  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 } **catch** (ExecutionException e) {  
 e.printStackTrace();  
 }  
}

**Función Verificar Usuario:** Se encarga de buscar y validar dentro de la base de datos de usuarios, si la persona que quiere ingresar se encuentra previamente registrada, mediante su nombre de usuario y contraseña, y por ende aparece en la base de datos. Caso contrario emitirá un mensaje señalando que el usuario ingresado no existe o no es válido.

**public void** verificarUsuario(View view)  
{  
 String[] resultadoSQL = **null**;  
 **try**{  
 datosConexion = **new** String[]{  
 serverIP,  
 port,  
 database,  
 userMySQL,  
 pwdMySQL,  
 **"SELECT \* FROM Usuario;"** };  
 String driver = **"com.mysql.jdbc.Driver"**;  
 Class.forName(driver).newInstance();  
 resultadoSQL = **new** AsyncQuery().execute(datosConexion).get();  
 Toast.makeText(Ingresar.**this**,**"Conexión Establecida"**, Toast.LENGTH\_LONG).show();  
 String resultadoConsulta = resultadoSQL[0];  
 String[] lista=resultadoConsulta.split(**"\n"**);  
 lista[0]=**"Nombre,Contraseña,Correo,User\_name"**;  
 **int** contador=1;  
 **boolean** veridico=**false**;  
 **for**(String dato: lista){  
 **if**(contador!=1){  
 String[] listaCompleta= dato.split(**","**);  
 **if** ((listaCompleta[2].equals(corr.getText().toString()))&&(listaCompleta[1].equals(contra.getText().toString()))&&(!(corr.getText().toString()).equals(**""**))&&(!(contra.getText().toString()).equals(**""**))){  
 veridico=**true**;  
 }  
 }**else**{  
 contador++;  
 }  
 }  
 **if**(veridico){  
 Toast.makeText(**this**,**"Si existe el usuario"**, Toast.LENGTH\_LONG).show();  
 Intent i4 = **new** Intent(**this**, Printers.**class**);  
 i4.putExtra(**"Correo"**,corr.getText().toString());  
 corr.setText(**""**);  
 contra.setText(**""**);  
 startActivity(i4);  
 }**else**{  
 Toast.makeText(**this**,**"Usuario no valido"**, Toast.LENGTH\_LONG).show();  
 }  
 }**catch**(Exception ex)  
 {  
 Toast.makeText(**this**, **"Error al obtener resultados de la consulta Transact-SQL: "** + ex.getMessage(), Toast.LENGTH\_LONG).show();  
 }  
}

**Func**

**X. ANÁLISIS DE PRESUPUESTO**

|  |  |
| --- | --- |
| **MATERIALES Y EQUIPOS** | **PRECIO** |
| RASPBERRY PI 3 | 70.00 $ |
| CABLE USB (IMPRESORA) | 3.00 $ |
| IMPRESORA 3D | 250.00-400.00 $ |
| DISPOSITIVO MÓVIL | 300.00 – 500.00 $ |
| CABLE DE RED UTP | 3.00 $ |
| ROUTER | 20.00 $ |
| WEBCAM | 5.00 $ |
| ORDENADOR | 500.00 – 750.00 $ |

**XI. CONCLUSIONES**

Se desarrolló una aplicación móvil con la capacidad de acceder a red local mediante una IP, a través de un usuario creado y aprobado por el administrador para controlar una impresora 3D mediante un dispositivo móvil (celular o Tablet), con la finalidad de eliminar la dependencia de la presencia física al momento de realizar una impresión. Esto ya que al existir errores dentro del proceso de impresión debemos de parar manualmente la impresora lo que podría ocasionar pérdidas de material y tiempo cuando no nos encontramos en el lugar de la impresión.

Se realizaron las correspondientes conexiones físicas utilizando principalmente una placa Raspberry Pi 3 vinculada directamente a la impresora de destino y a la red en la se desea trabajar, de manera que placa pueda suministrar la información necesaria a la impresora para empezar a realizar la función que queramos. Esto se logró conectando la placa a un servicio de almacenamiento en nube desde el cual se carga el archivo del modelo 3D que se desea trabajar, desde una red externa.