Equipo Base de Datos Gestor clínico

Jorge Carlos Parra Alvarado

Luis Gustavo Zepeda González

Asignatura de trabajo: Programación paralela.

Trabajo: Base de Datos + Descripción Servidor Web.

BASE DE DATOS GESTOR CLINICO

La Base de datos fue creado en host local con el uso del gestor XAMPP, en la cual se crearon un total de 14 tablas, cada una de estas con un identificador con autoincremento (el identificador se encuentra como id+ nombre de la tabla)), así mismo a la BD sele asigno una pequeña descripción en la que se especifica y/o menciona que hace cada tabla de la BD.

CREATE DATABASE GestorClinico;

USE GestorClinico;

CREATE TABLE centros\_medicos(

    id\_centromedico INT AUTO\_INCREMENT,

    nombre\_centromedico VARCHAR(100) NOT NULL,

    especialidad\_centromedico VARCHAR(100) DEFAULT 'SE',

    direccion VARCHAR(100) NOT NULL,

    telefono VARCHAR(15) NOT NULL,

    PRIMARY KEY(id\_centromedico)

);

#Todos los centros medicos tienen:

CREATE TABLE personas(

    id\_persona INT AUTO\_INCREMENT,

    nombre VARCHAR(30) NOT NULL,

    apellido\_paterno VARCHAR(30) NOT NULL,

    apellido\_materno VARCHAR(30) NOT NULL,

    f\_nacimiento DATE NOT NULL,

    sexo CHAR(1) NOT NULL DEFAULT 'I', #H=hombre, M=mujer, I=indefinido

    curp VARCHAR(18) NOT NULL UNIQUE,

    tipo\_persona CHAR(1) NOT NULL, #1=admin de sistema, 2=medico, 3=paciente, 4=medico y paciente

    PRIMARY KEY(id\_persona)

);

#todas las personas son usuarios

CREATE TABLE usuarios(

    id\_usuario INT AUTO\_INCREMENT,

    id\_persona INT NOT NULL UNIQUE,

    correo VARCHAR(100) NOT NULL UNIQUE,

    password VARCHAR(300) NOT NULL,

    PRIMARY KEY(id\_usuario),

    FOREIGN KEY (id\_persona) REFERENCES personas (id\_persona)

);

#las personas pueden ser medicos

CREATE TABLE medicos(

    id\_medico INT AUTO\_INCREMENT,

    id\_persona INT NOT NULL UNIQUE,

    rfc VARCHAR(13) NOT NULL UNIQUE,

    cedula\_profesional VARCHAR(50) NOT NULL UNIQUE,

    especialidad VARCHAR(250) NOT NULL DEFAULT 'Medico general',

    PRIMARY KEY(id\_medico),

    FOREIGN KEY(id\_persona) REFERENCES personas (id\_persona)

);

#los medicos pueden dar consultas en varios centrs medicos

CREATE TABLE cm\_medicos(

    id\_cm\_medico INT AUTO\_INCREMENT,

    id\_centromedico INT NOT NULL,

    id\_medico INT NOT NULL,

    horario VARCHAR(200) NOT NULL,

    PRIMARY KEY(id\_cm\_medico),

    FOREIGN KEY(id\_centromedico) REFERENCES centros\_medicos (id\_centromedico),

    FOREIGN KEY(id\_medico) REFERENCES medicos (id\_medico)

);

#las personas pueden ser pacientes

CREATE TABLE pacientes(

    id\_paciente INT AUTO\_INCREMENT,

    id\_persona INT NOT NULL UNIQUE,

    id\_centromedico INT NOT NULL,

    nss VARCHAR(20) NOT NULL UNIQUE,

    grupo\_sanguineo VARCHAR(20) NOT NULL,

    enfermedades VARCHAR(200) DEFAULT NULL, #o padecimientos

    PRIMARY KEY(id\_paciente),

    FOREIGN KEY(id\_persona) REFERENCES personas (id\_persona),

    FOREIGN KEY(id\_centromedico) REFERENCES centros\_medicos (id\_centromedico)

);

#los pacientes pueden agendar citas con medicos

CREATE TABLE citas(

    id\_cita INT AUTO\_INCREMENT,

    id\_paciente INT NOT NULL,

    id\_medico INT NOT NULL,

    asunto VARCHAR(100) NOT NULL,

    fecha\_cita DATE NOT NULL,

    hora\_cita TIME NOT NULL,

    estado\_cita CHAR(1) DEFAULT '0', #0=cita, 1=cita realizada

    cantidad\_pagar FLOAT DEFAULT 0,#Totfal a pagar

    PRIMARY KEY(id\_cita),

    FOREIGN KEY(id\_paciente) REFERENCES pacientes (id\_paciente),

    FOREIGN KEY(id\_medico) REFERENCES medicos (id\_medico)

);

#Las citas pueden generar recetas

CREATE TABLE recetas(

    id\_receta INT AUTO\_INCREMENT,

    id\_cita INT NOT NULL,

    fecha\_receta DATE,

    observaciones VARCHAR(500) NOT NULL,

    PRIMARY KEY(id\_receta),

    FOREIGN KEY(id\_cita) REFERENCES citas (id\_cita)

);

#Las recetas pueden requerir de medicamentos, por lo tanto esto quiere decir

# que los centros medicos tienen almacenes

CREATE TABLE almacenes(

    id\_almcen INT AUTO\_INCREMENT,

    id\_centromedico INT NOT NULL,

    nombre\_almacen VARCHAR(100) NOT NULL,

    PRIMARY KEY(id\_almcen),

    FOREIGN KEY(id\_centromedico) REFERENCES centros\_medicos (id\_centromedico)

);

#los almacenes a su ves tienen productos

CREATE TABLE medicamentos(

    id\_medicamento INT AUTO\_INCREMENT,

    id\_almcen INT NOT NULL,

    nombre\_producto VARCHAR(100) NOT NULL,

    descripcion VARCHAR(250) DEFAULT 'SD',

    stok INT NOT NULL DEFAULT 0,

    precio\_unitario FLOAT NOT NULL,

    PRIMARY KEY(id\_medicamento),

    FOREIGN KEY(id\_almcen) REFERENCES almacenes (id\_almcen)

);

#retomando las recetas, estas pueden requerir de medicamentos

CREATE TABLE receta\_medicamentos(

    id\_receta\_medicamento INT AUTO\_INCREMENT,

    id\_receta INT NOT NULL,

    id\_medicamento INT NOT NULL,

    cantidad\_medicamento INT NOT NULL,

    PRIMARY KEY(id\_receta\_medicamento),

    FOREIGN KEY(id\_receta) REFERENCES recetas(id\_receta),

    FOREIGN KEY(id\_medicamento) REFERENCES medicamentos (id\_medicamento)

);

#los almacenes pueden realizar pedidos

CREATE TABLE pedidos(

    id\_pedido INT AUTO\_INCREMENT,

    id\_almcen INT NOT NULL,

    id\_usuario INT NOT NULL,

    fecha\_pedido DATE NOT NULL,

    estado\_pedido CHAR(1) NOT NULL DEFAULT '0',#0=pedido,#2=recibido

    fecha\_recibido DATE DEFAULT NULL,

    PRIMARY KEY(id\_pedido),

    FOREIGN KEY(id\_almcen) REFERENCES almacenes (id\_almcen),

    FOREIGN KEY(id\_usuario) REFERENCES usuarios (id\_usuario)

);

#los pedidos pueden ser de diferentes proveedores

CREATE TABLE proveedores(

    id\_proveedor INT AUTO\_INCREMENT,

    nombre\_proveedor VARCHAR(100) NOT NULL,

    direccion VARCHAR(250),

    correo VARCHAR(100) NOT NULL UNIQUE,

    PRIMARY KEY(id\_proveedor)

);

#Para registrar los medicamentos del pedido

CREATE TABLE pedido\_medicamento\_proveedor(

    id\_ppp INT AUTO\_INCREMENT,

    id\_pedido INT NOT NULL,

    id\_proveedor INT NOT NULL,

    id\_medicamento INT NOT NULL,

    cantidad\_solicitada INT NOT NULL,

    precio\_unitario FLOAT NOT NULL,

    PRIMARY KEY(id\_ppp),

    FOREIGN KEY(id\_pedido) REFERENCES pedidos (id\_pedido),

    FOREIGN KEY(id\_proveedor) REFERENCES proveedores (id\_proveedor),

    FOREIGN KEY(id\_medicamento) REFERENCES medicamentos (id\_medicamento)

);

Para insertar la BD existen diversas formas para agregarla al gestor xampp, una es por medio de la consola y la segunda es por medio de la importación desde el navegador Chrome.

1.- Desde consola

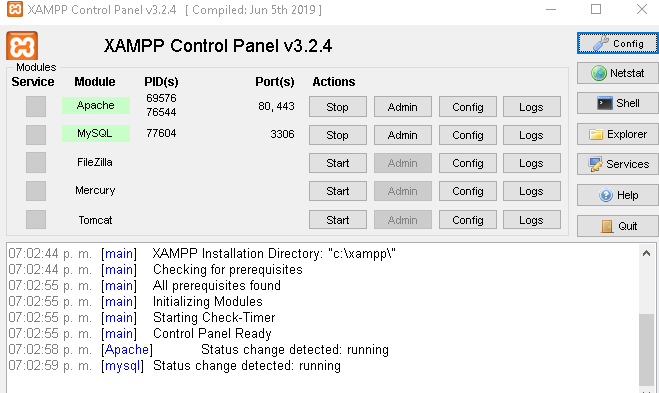


Ilustración 1: Se arrancan los servicios Apache y MySQL

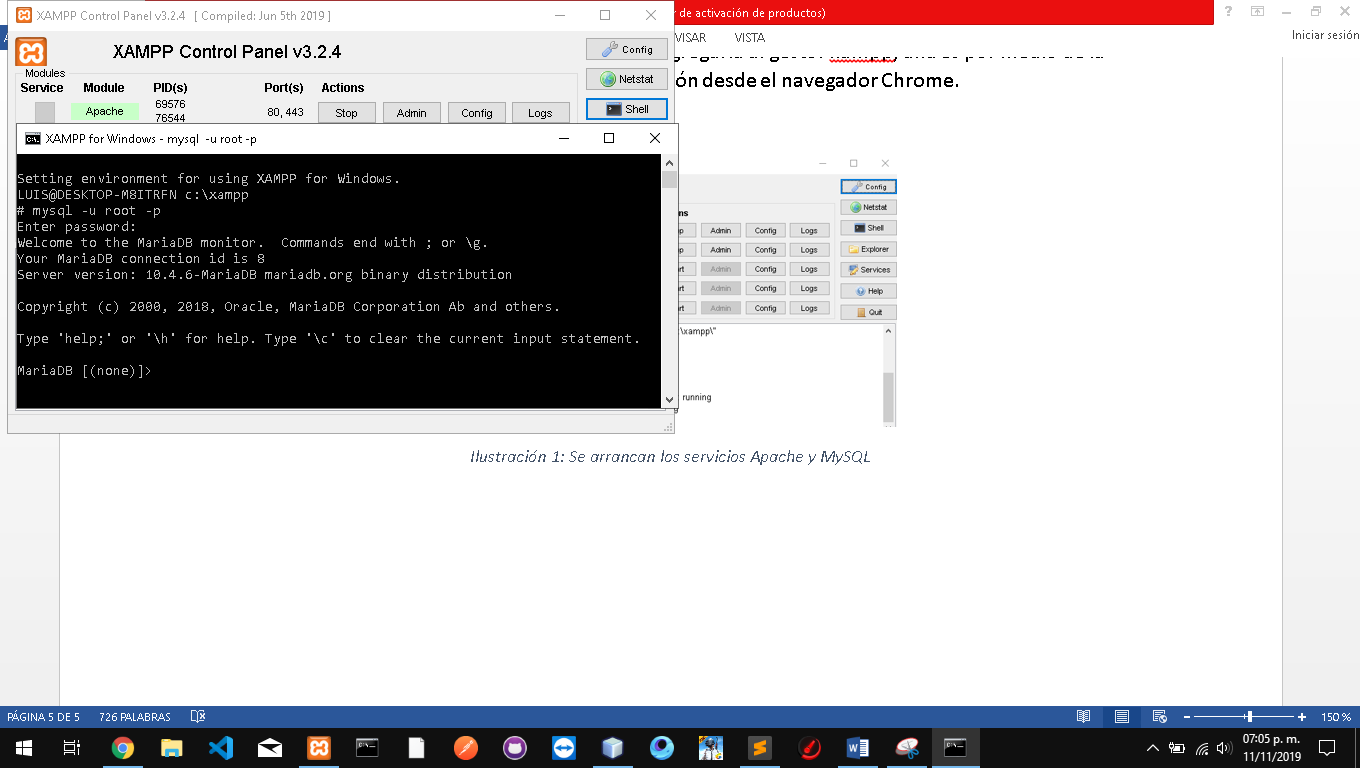


Ilustración 2: Se abre consola y se accesa a MariaDB

A partir de aquí se puede consultar si la base de datos existe con el comando Show databases; para visualizar las BD existentes y verificar que nuestra BD no exista para que no existan problemas al crearse.

Posteriormente se copia el texto de (código) SQL para crear la BD desde consola.

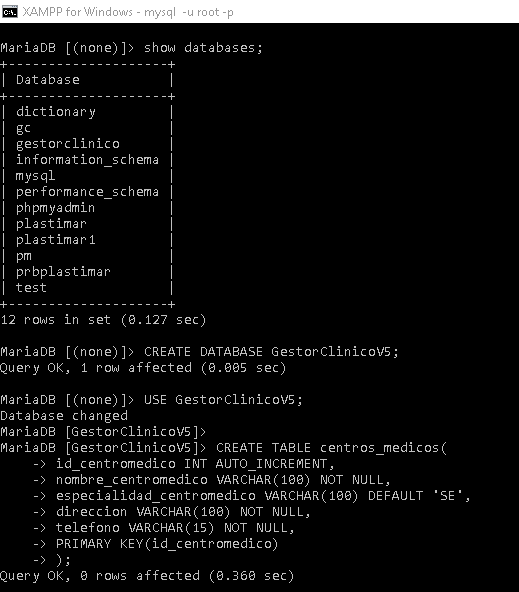


Ilustración 3: Consulta e inserción de tablas

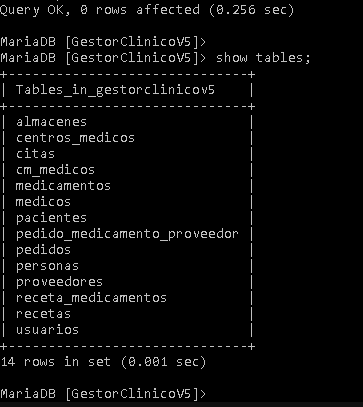


Ilustración 4: Consulta de tablas creadas

2.- Desde el navegador

Ya teniendo los servicios activos de MySQL y Apache en el navegador se teclea localhost o 127.0.0.1 para abrir la dirección del servidor phpMyAdmin, que es lo que nos permitirá la administración de la BD por un medio gráfico.

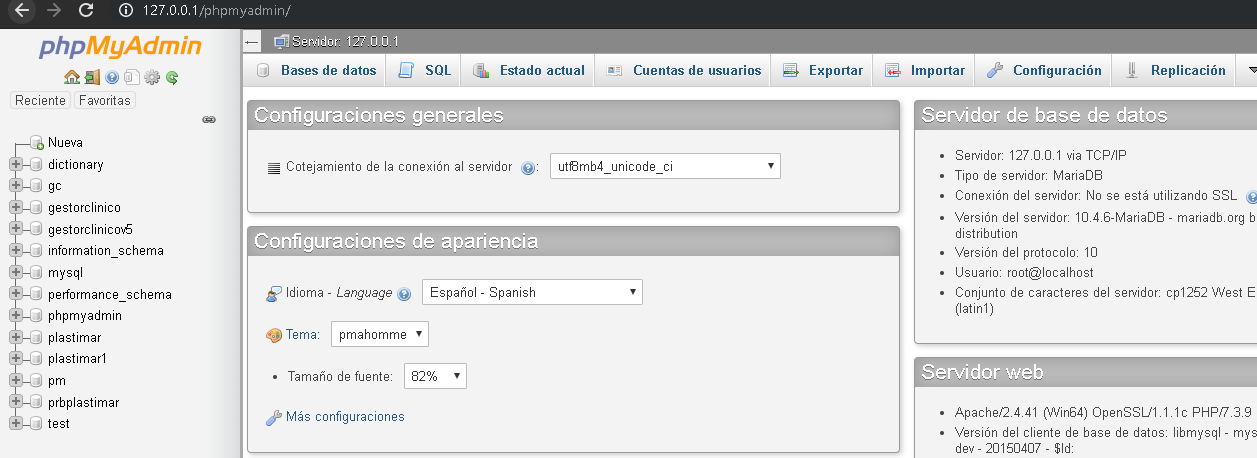


Ilustración 5: Acceso a localhost desde el navegador.

La BD se puede insertar directamente desde el apartado de SQL, o generando una nueva BD e importando las tablas, aunque esta opción es un poco más recomendada para tablas que ya contienen datos.

**MONTAR BD EN SERVIDOR WEB 000WEBHOST**

Lo primero requerido es contar con un usuarios (este nos permite crear uno gratuitamente con un mínimo de servicios).

Una vez creada nuestra cuenta accedemos al panel principal y creamos un nuevo proyecto (Create New Site).

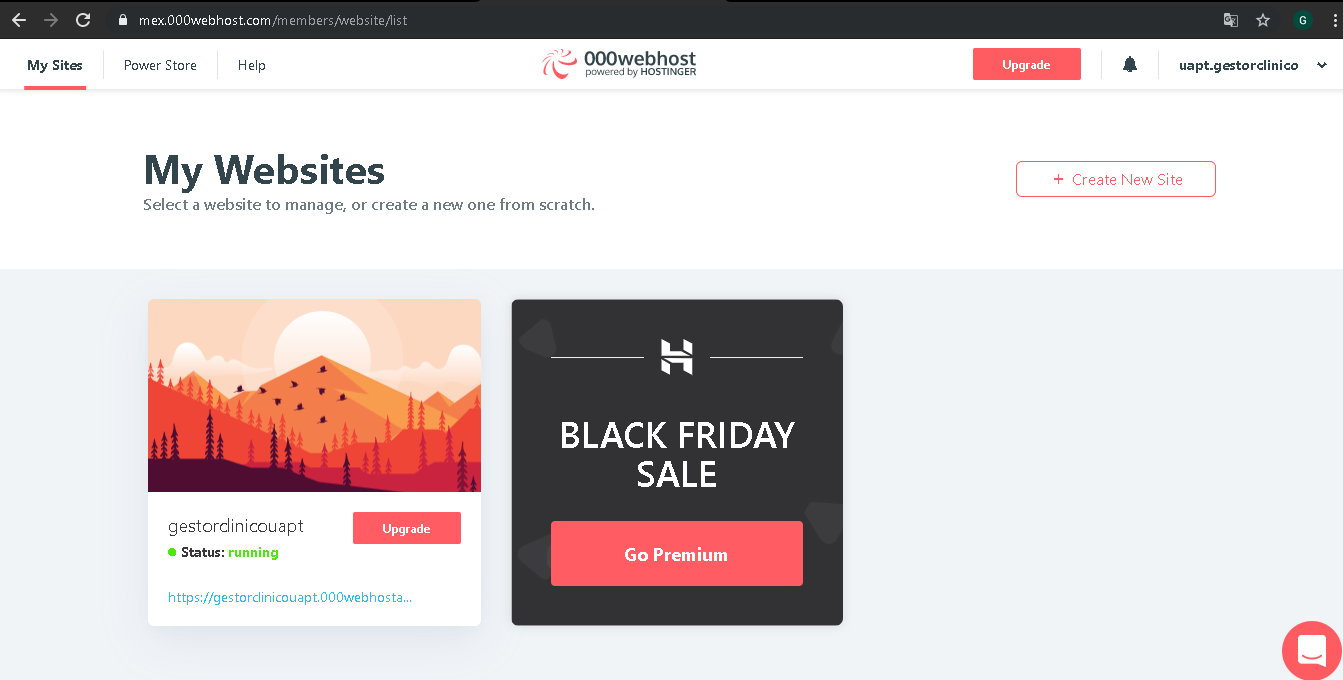


Ilustración 6: Panel principal de 000webhost

Al crear el sitio nos permitirá gestionar el sistema a partir de aquí, en este caso veremos la inserción en la BD.

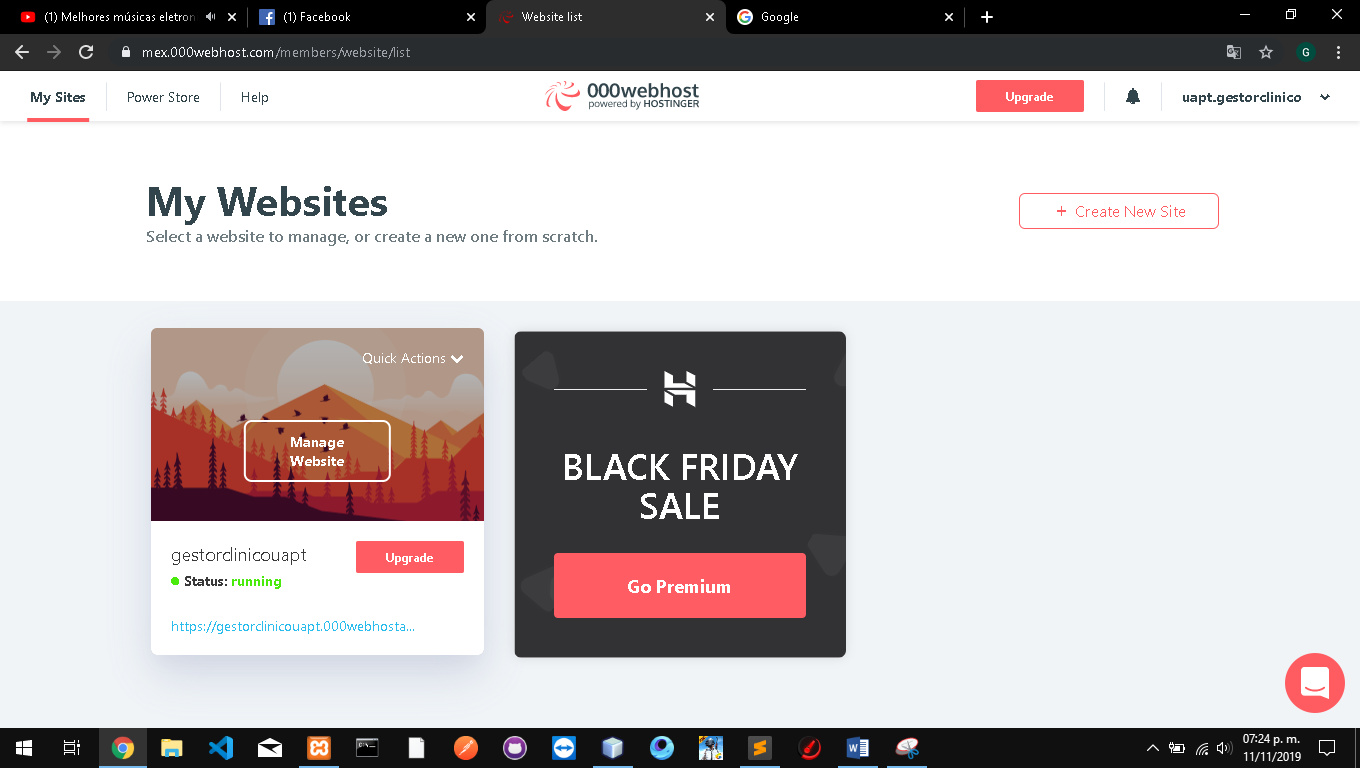


Ilustración 7: Seleccionamos administrar sitio web

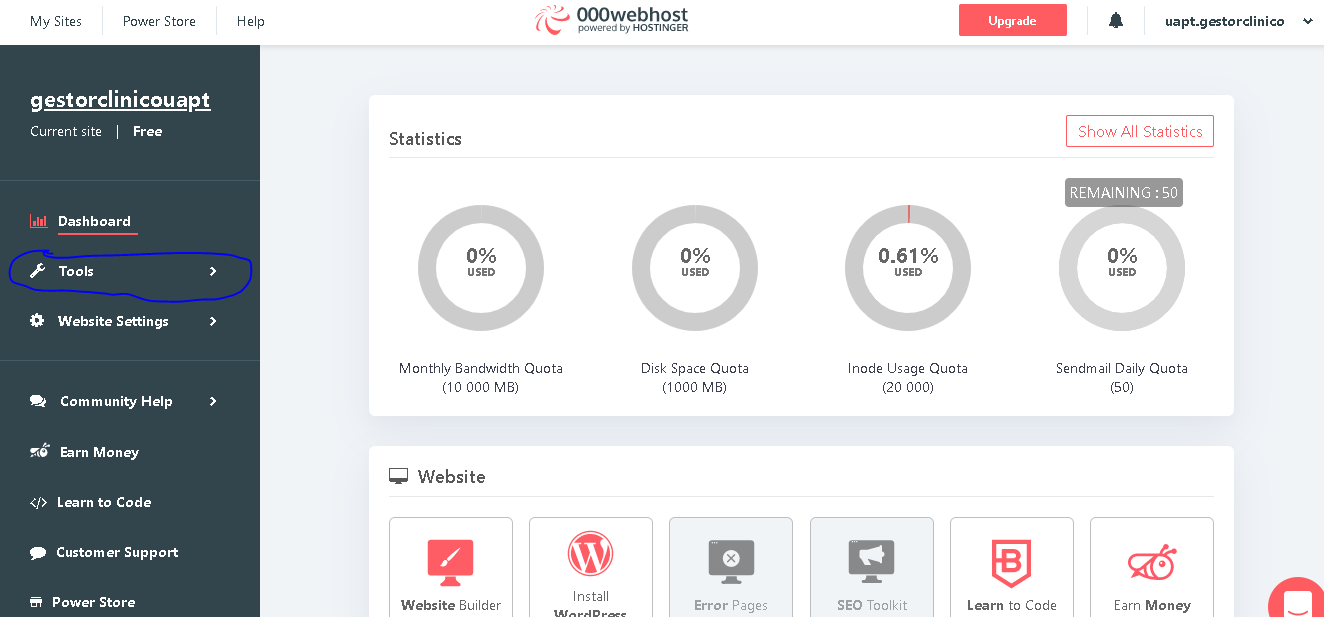


Ilustración 8 : Al acceder a la administración del sitio nos aparecerán algunas herramientas de gestión (seleccionamos Tools).

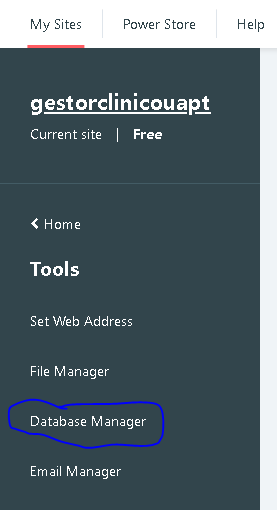


Ilustración 9 : Seleccionamos DataBase Manager

Podemos crear una BD nueva o trabajar en la que ya tenemos .

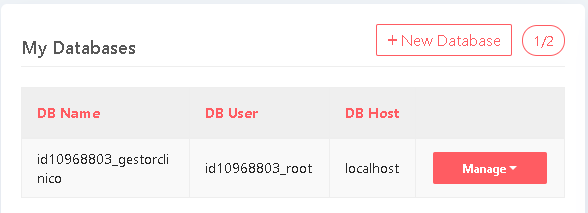


Ilustración 10: BD existentes

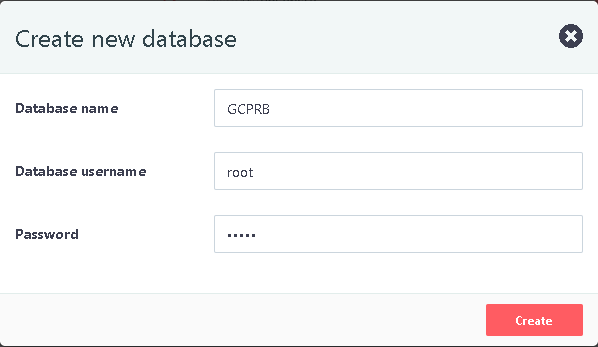


Ilustración 11: Creación de una nueva BD

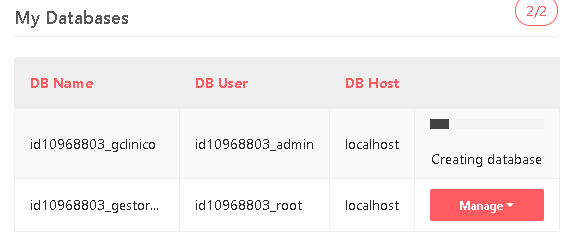


Ilustración 12: Nueva BD creada

Desde este momento podemos trabajar con la BD creada, seleccionamos Manage y la opción PhpMyAdmin y nos abrirá el gestor web de BD.

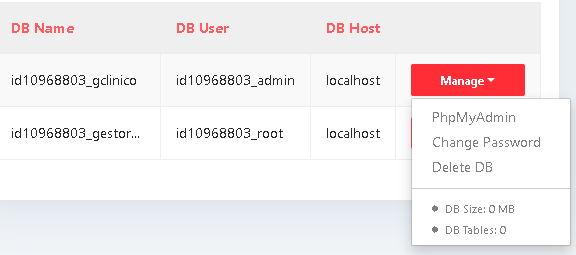


Ilustración 13: Seleccionar administrar y seleccionar PhpMyAdmin

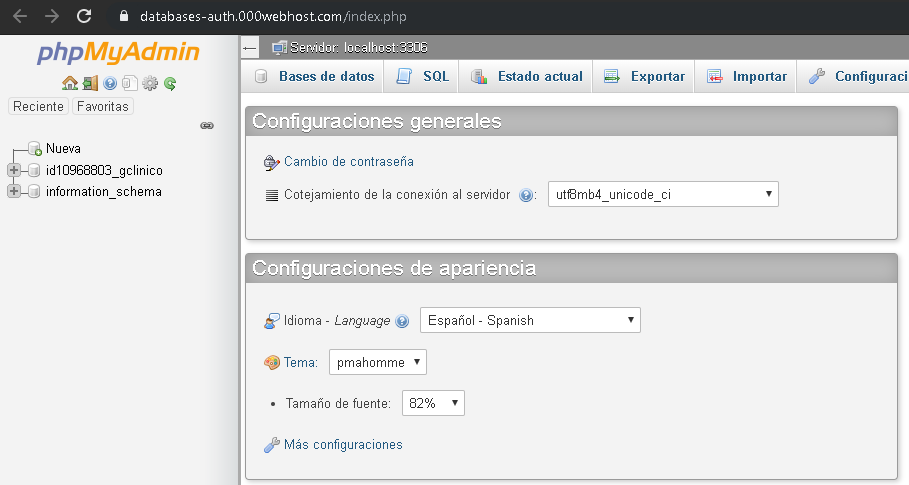


Ilustración 14: Vista del gestor web de BD

Para continuar insertamos el código desde el apartado SQL y le damos continuar.



Ilustración 15: Inserción del código SQL

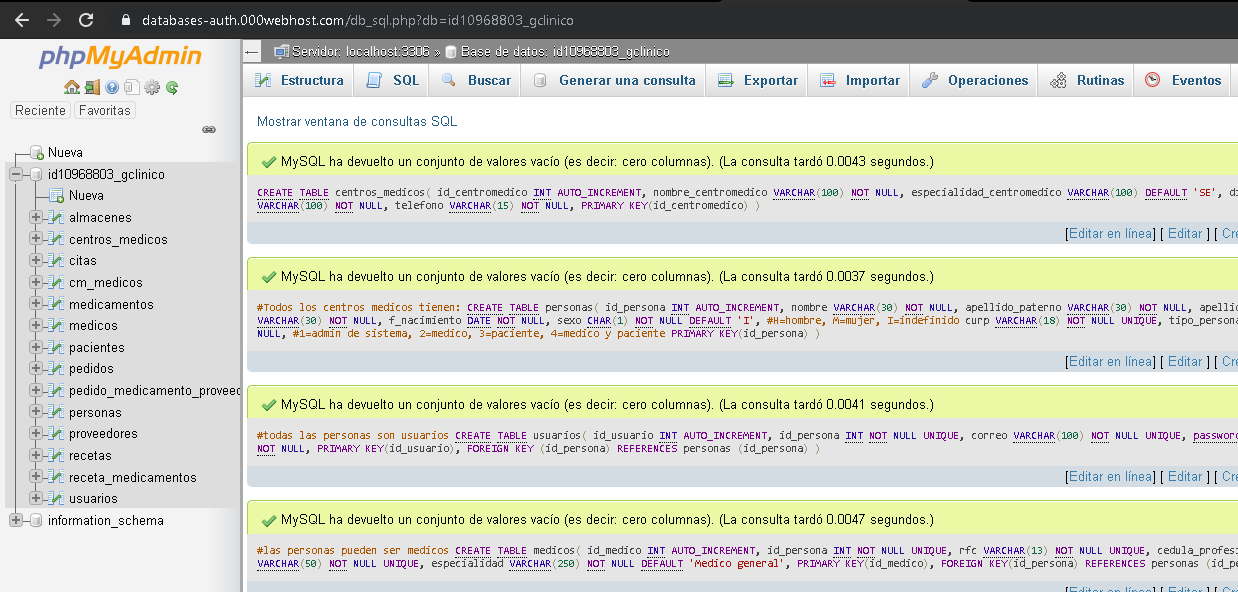


Ilustración 16: La pantalla nos muestra cuando las Tablas de la BD han sido creadas

Para poder administrar una BD web tenemos que crear un archivo de conexión, este puede ser uno que nos cree un camino remoto y que se modifique cada que un usuario lo utilice.

Este codigo nos permite esto, pero primero tendremos que configurar el entorno, y es un archivo punto Json.

{

    "remotePath": "./public\_html/",

    "host": "files.000webhost.com",

    "username": "gestorclinicouapt",

    "password": "adminsistema2019.",

    "port": 21,

    "secure": false,

    "protocol": "ftp",

    "uploadOnSave": false,

    "passive": false,

    "debug": false,

    "privateKeyPath": null,

    "passphrase": null,

    "agent": null,

    "allow": [],

    "ignore": [

        "\\.vscode",

        "\\.git",

        "\\.DS\_Store"

    ],

    "generatedFiles": {

        "extensionsToInclude": [

            ""

        ],

        "path": ""

    }

}

El segundo es una clase de conexión java y nos permite gestionar la BD a partir de la dirección del host en el que está esta almacenada en este caso realizamos pruebas con freemysqlhosting.

public class Conexiones {

    public static String driver = "com.mysql.cj.jdbc.Driver";

    public static String url = "jdbc:mysql://sql9.freemysqlhosting.net/sql9310189";

    public static String user = "sql9310189";

    public static String password = "TJTURy186b";

    public static Connection conectar() {

        try {

            // The newInstance() call is a work around for some

            // broken Java implementations

            Class.forName(driver).newInstance();  //Definicion del driver

        } catch (Exception ex) {

            // handle the error

        }

        Connection conn = null;

        try {

            conn = DriverManager.getConnection(url, user, password);

            if (conn != null) {

                System.out.println("Conexion exitosa");

            }

            // Do something with the Connection

        } catch (SQLException ex) {

            // handle any errors

            System.out.println("SQLException: " + ex.getMessage());

            System.out.println("SQLState: " + ex.getSQLState());

            System.out.println("VendorError: " + ex.getErrorCode());

        }

        return conn;

    }

    public void desconectar(Connection con) throws SQLException {

        con.close();

    }

    public Resultset consultas(Connection conn, String query) {

        Statement stmt = null;

        ResultSet rs = null;

        try {

            stmt = conn.createStatement();

            rs = stmt.executeQuery(query);//"SELECT \* FROM persona where id=2

            // or alternatively, if you don't know ahead of time that

            // the query will be a SELECT...

            if (stmt.execute(query)) {//"SELECT \* FROM persona where id=2"

                rs = stmt.getResultSet();

            }

            if (rs.next()) { // una sola linea

                JOptionPane.showMessageDialog(null, rs.getString("nombre") + " " + rs.getString("domicilio"));

            } else {

                JOptionPane.showMessageDialog(null, "No existen datos");

            }

            // Now do something with the ResultSet ....

        } catch (SQLException ex) {

            // handle any errors

            System.out.println("SQLException: " + ex.getMessage());

            System.out.println("SQLState: " + ex.getSQLState());

            System.out.println("VendorError: " + ex.getErrorCode());

        }

        return (Resultset) rs;

    }

    public static Boolean conss(String nombre, String contra) {

        Connection con = Conexiones.conectar();

        PreparedStatement ps;

        ResultSet rs;

        try {

            ps = con.prepareStatement("INSERT INTO cat\_usuario (usuario,contrasenia,semestre\_actual,bactive,correo) VALUES (?,?,?,?,?)");

            ps.setString(1, nombre);

            ps.setString(2, contra);

            ps.setInt(3, 1);

            ps.setBoolean(4, true);

            ps.setString(5, contra);

            int res = ps.executeUpdate();

            if (res > 0) {

                System.out.println("Se registro exitosamente");

                con.close();

                return true;

            } else {

                System.out.println("error");

                con.close();

                return false;

            }

        } catch (Exception e) {

            System.out.println(e.toString());

        }

        return false;

    }

}

Y en este otro caso realizamos pruebas con 000webhost

public class Conexiones {

    public static String driver = "com.mysql.cj.jdbc.Driver";

    public static String url = "jdbc:mysql://mysql2.files.000webhost.com/gestorclinicouapt";

    public static String user = "gestorclinicouapt";

    public static String password = "adminsistema2019.";

    public static Connection conectar() {

        try {

            // The newInstance() call is a work around for some

            // broken Java implementations

            Class.forName(driver).newInstance();  //Definicion del driver

        } catch (Exception ex) {

            // handle the error

        }

        Connection conn = null;

        try {

            conn = DriverManager.getConnection(url, user, password);

            if (conn != null) {

                System.out.println("Conexion exitosa");

            }

            // Do something with the Connection

        } catch (SQLException ex) {

            // handle any errors

            System.out.println("SQLException: " + ex.getMessage());

            System.out.println("SQLState: " + ex.getSQLState());

            System.out.println("VendorError: " + ex.getErrorCode());

        }

        return conn;

    }

    public void desconectar(Connection con) throws SQLException {

        con.close();

    }

    public Resultset consultas(Connection conn, String query) {

        Statement stmt = null;

        ResultSet rs = null;

        try {

            stmt = conn.createStatement();

            rs = stmt.executeQuery(query);//"SELECT \* FROM persona where id=2

            // or alternatively, if you don't know ahead of time that

            // the query will be a SELECT...

            if (stmt.execute(query)) {//"SELECT \* FROM persona where id=2"

                rs = stmt.getResultSet();

            }

            if (rs.next()) { // una sola linea

                JOptionPane.showMessageDialog(null, rs.getString("nombre") + " " + rs.getString("domicilio"));

            } else {

                JOptionPane.showMessageDialog(null, "No existen datos");

            }

            // Now do something with the ResultSet ....

        } catch (SQLException ex) {

            // handle any errors

            System.out.println("SQLException: " + ex.getMessage());

            System.out.println("SQLState: " + ex.getSQLState());

            System.out.println("VendorError: " + ex.getErrorCode());

        }

        return (Resultset) rs;

    }

    public static Boolean conss(String nombre, String contra) {

        Connection con = Conexiones.conectar();

        PreparedStatement ps;

        ResultSet rs;

        try {

            ps = con.prepareStatement("INSERT INTO cat\_usuario (usuario,contrasenia,semestre\_actual,bactive,correo) VALUES (?,?,?,?,?)");

            ps.setString(1, nombre);

            ps.setString(2, contra);

            ps.setInt(3, 1);

            ps.setBoolean(4, true);

            ps.setString(5, contra);

            int res = ps.executeUpdate();

            if (res > 0) {

                System.out.println("Se registro exitosamente");

                con.close();

                return true;

            } else {

                System.out.println("error");

                con.close();

                return false;

            }

        } catch (Exception e) {

            System.out.println(e.toString());

        }

        return false;

    }

}

Sin embargo para este segundo sitio causa errores por la conexión ftp.