

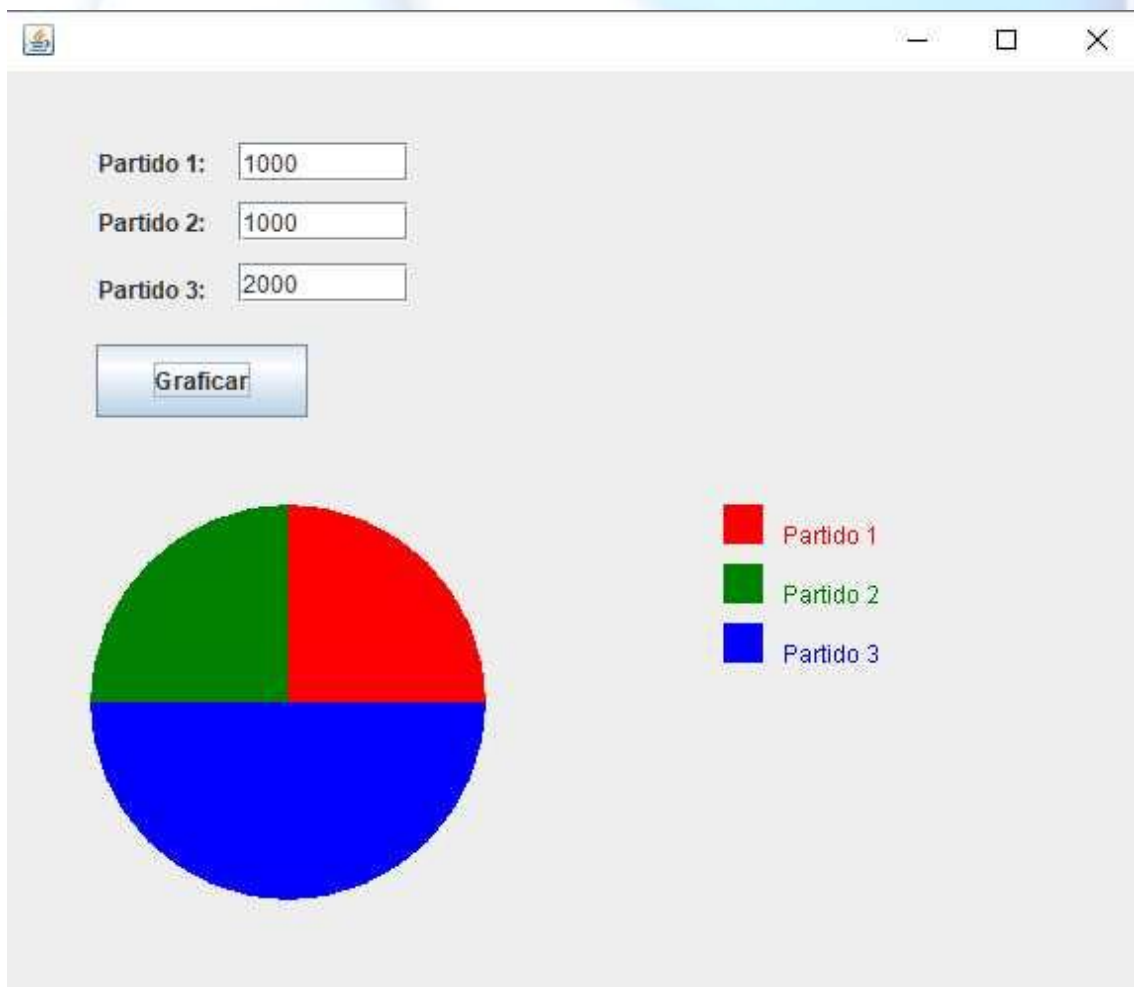
Curso de Java a distancia

Clase 19: Gráficos estadísticos

El objetivo de este concepto es la implementación de algoritmos para mostrar gráficos estadísticos.

Problema 1

Crear una aplicación que solicite el ingreso de tres valores por teclado que representan las cantidades de votos obtenidas por tres partidos políticos. Luego mostrar un gráfico de tartas:



El algoritmo es:

```
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Graphics;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class GraficoTarta extends JFrame {

    private JPanel contentPane;
    private JTextField tf1;
    private JTextField tf2;
    private JTextField tf3;
    private boolean bandera=false;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    GraficoTarta frame = new GraficoTarta();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public GraficoTarta() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 800, 600);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblPartido = new JLabel("Partido 1:");
        lblPartido.setBounds(46, 39, 61, 14);
        contentPane.add(lblPartido);

        JLabel lblPartido_1 = new JLabel("Partido 2:");
        lblPartido_1.setBounds(46, 69, 61, 14);
        contentPane.add(lblPartido_1);

        JLabel lblPartido_2 = new JLabel("Partido 3:");
        lblPartido_2.setBounds(46, 103, 61, 14);
        contentPane.add(lblPartido_2);
    }
}
```

```

tf1 = new JTextField();
tf1.setBounds(117, 36, 86, 20);
contentPane.add(tf1);
tf1.setColumns(10);

tf2 = new JTextField();
tf2.setBounds(117, 66, 86, 20);
contentPane.add(tf2);
tf2.setColumns(10);

tf3 = new JTextField();
tf3.setBounds(117, 97, 86, 20);
contentPane.add(tf3);
tf3.setColumns(10);

JButton btnGraficar = new JButton("Graficar");
btnGraficar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        bandera=true;
        repaint();
    }
});
btnGraficar.setBounds(45, 138, 107, 37);
contentPane.add(btnGraficar);
}

public void paint(Graphics g)
{
    super.paint(g);
    if (bandera==true)
    {
        String s1=tf1.getText();
        String s2=tf2.getText();
        String s3=tf3.getText();
        int v1=Integer.parseInt(s1);
        int v2=Integer.parseInt(s2);
        int v3=Integer.parseInt(s3);
        int suma=v1+v2+v3;
        int grados1=v1*360/suma;
        int grados2=v2*360/suma;
        int grados3=v3*360/suma;

        g.setColor(new Color(255,0,0));
        g.fillArc(50,250,200,200,0,grados1);
        g.fillRect(370,250,20,20);
        g.drawString("Partido 1", 400, 270);

        g.setColor(new Color(0,128,0));
        g.fillArc(50,250,200,200,grados1,grados2);
        g.fillRect(370,280,20,20);
        g.drawString("Partido 2", 400, 300);

        g.setColor(new Color(0,0,255));
        g.fillArc(50,250,200,200,grados1+grados2,grados3);
        g.fillRect(370,310,20,20);
        g.drawString("Partido 3", 400, 330);
    }
}
}

```

Disponemos un if en el método paint para controlar que se haya presionado el botón graficar:

```
public void paint(Graphics g)
{
    super.paint(g);
    if (bandera==true)
    {
```

El atributo bandera se inicializa cuando se define con el valor false, esto hace que cuando se ejecute por primera vez el método paint no ingrese al if:

```
private boolean bandera=false;
```

Se definen tres objetos de la clase JTextField para ingresar los tres valores por teclado:

```
private JTextField tf1;
private JTextField tf2;
private JTextField tf3;
```

Cuando se presiona el botón se cambia el estado del atributo bandera por el valor true y se llama al método repaint (recordemos que este método borra el JFrame y llama al método paint para que ahora dibuje el gráfico de tarta):

```
btnGraficar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        bandera=true;
        repaint();
    }
});
```

Lo primero que hacemos para graficar la tarta es rescatar los tres valores ingresados en los controles JTextField:

```
if (bandera==true)
{
    String s1=tf1.getText();
    String s2=tf2.getText();
    String s3=tf3.getText();
```

Los convertimos a tipo de dato entero:

```
int v1=Integer.parseInt(s1);
int v2=Integer.parseInt(s2);
int v3=Integer.parseInt(s3);
```

Sumamos los tres valores:

```
int suma=v1+v2+v3;
```

Seguidamente calculamos los grados que le corresponde a cada trozo de tarta (teniendo en cuenta que tenemos 360 grados para repartir):

Cada trozo de tarta lo obtenemos mediante la ecuación:

tamaño trozo= cantidad de votos del partido/ total de votos * 360

si observamos la ecuación podemos imaginar que la división:

cantidad de votos del partido / total de votos

generará un valor menor a uno (salvo que el partido haya obtenido todos los votos de la elección en cuyo caso la división genera el valor uno)

Esta división nos genera el porcentaje de votos que le corresponde al partido y luego dicho porcentaje lo multiplicamos por la cantidad de grados a repartir (que son 360 grados)

Luego como la división generará un valor menor a uno y al tratarse de dos variables enteras el resultado será cero. Para evitar este problema procedemos primero a multiplicar por 360 y luego dividir por la variable suma:

```
int grados1=v1*360/suma;  
int grados2=v2*360/suma;  
int grados3=v3*360/suma;
```

Si quisiéramos primero dividir y luego multiplicar por 360 debemos proceder a anteceder a cada operación el tipo de resultado a obtener:

```
int grados1=(int)((float)v1/suma*360);  
int grados2=(int)((float)v2/suma*360);  
int grados3=(int)((float)v3/suma*360);
```

Como podemos ver es más complicado si queremos primero efectuar la división y luego el producto.

Procedemos ahora a graficar los trozos de tarta y leyenda con el valor ingresado (activamos el color rojo y mediante el método fillArc creamos un trozo de tarta que se inicia en el grado 0 y avanza tantos grados como indica la variable grados1. Luego mediante el método drawString mostramos la leyenda del partido respectivo con un cuadradito también de color rojo):

```
g.setColor(new Color(255,0,0));  
g.fillArc(50,250,200,200,0,grados1);  
g.fillRect(370,250,20,20);  
g.drawString("Partido 1", 400, 270);
```

El segundo trozo comienza en grados1 y avanza tantos grados como lo indica la variable grados2. Activamos el color verde para diferenciar del trozo anterior:

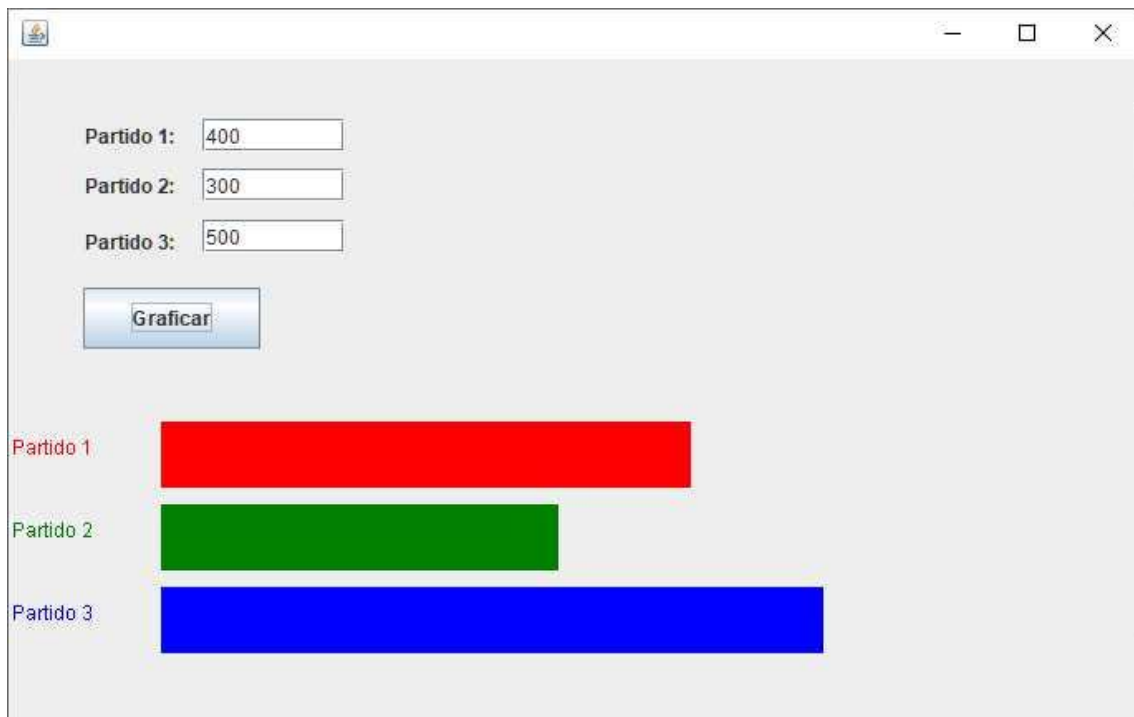
```
g.setColor(new Color(0,128,0));  
g.fillArc(50,250,200,200,grados1,grados2);  
g.fillRect(370,280,20,20);  
g.drawString("Partido 2", 400, 300);
```

El último trozo lo graficamos a partir de la suma de grados1+grados2 y avanzamos tantos grados como lo indica la variable grados3:

```
g.setColor(new Color(0,0,255));  
g.fillArc(50,250,200,200,grados1+grados2,grados3);  
g.fillRect(370,310,20,20);  
g.drawString("Partido 1", 400, 330);
```

Problema 2

Crear una aplicación que solicite el ingreso de tres valores por teclado que representan las cantidades de votos obtenidas por tres partidos políticos. Luego mostrar un gráfico de barras horizontales:



El algoritmo es:

```
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Graphics;
```

```
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
```

```
public class GraficoBarraHorizontal extends JFrame {
```

```
    private JPanel contentPane;
    private JTextField tf3;
    private JTextField tf1;
    private JTextField tf2;
    private boolean bandera=false;
```

```
    /**
```

```
     * Launch the application.
```

```
    */
```

```
    public static void main(String[] args) {
```

```
        EventQueue.invokeLater(new Runnable() {
```

```
            public void run() {
```

```
                try {
```

```
                    GraficoBarraHorizontal frame = new GraficoBarraHorizontal();
```

```
                    frame.setVisible(true);
```

```
                } catch (Exception e) {
```

```
                    e.printStackTrace();
```

```
                }
```

```
            }
```



```

    });
}

/**
 * Create the frame.
 */
public GraficoBarraHorizontal() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 800, 600);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblPartido = new JLabel("Partido 1:");
    lblPartido.setBounds(46, 39, 61, 14);
    contentPane.add(lblPartido);

    JLabel lblPartido_1 = new JLabel("Partido 2:");
    lblPartido_1.setBounds(46, 69, 61, 14);
    contentPane.add(lblPartido_1);

    JLabel lblPartido_2 = new JLabel("Partido 3:");
    lblPartido_2.setBounds(46, 103, 61, 14);
    contentPane.add(lblPartido_2);

    tf1 = new JTextField();
    tf1.setBounds(117, 36, 86, 20);
    contentPane.add(tf1);
    tf1.setColumns(10);

    tf2 = new JTextField();
    tf2.setBounds(117, 66, 86, 20);
    contentPane.add(tf2);
    tf2.setColumns(10);

    tf3 = new JTextField();
    tf3.setBounds(117, 97, 86, 20);
    contentPane.add(tf3);
    tf3.setColumns(10);

    JButton btnGraficar = new JButton("Graficar");
    btnGraficar.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            bandera=true;
            repaint();
        }
    });
    btnGraficar.setBounds(45, 138, 107, 37);
    contentPane.add(btnGraficar);
}

```

```

public void paint(Graphics g)
{
    super.paint(g);
    if (bandera==true)
    {
        String s1=tf1.getText();
        String s2=tf2.getText();
        String s3=tf3.getText();
    }
}

```

```

int v1=Integer.parseInt(s1);
int v2=Integer.parseInt(s2);
int v3=Integer.parseInt(s3);
int mayor=retornarMayor(v1,v2,v3);

int largo1=v1*400/mayor;
int largo2=v2*400/mayor;
int largo3=v3*400/mayor;

g.setColor(new Color(255,0,0));
g.fillRect(100,250,largo1,40);
g.drawString("Partido 1", 10, 270);

g.setColor(new Color(0,128,0));
g.fillRect(100,300,largo2,40);
g.drawString("Partido 2", 10, 320);

g.setColor(new Color(0,0,255));
g.fillRect(100,350,largo3,40);
g.drawString("Partido 3", 10, 370);
}
}

private int retornarMayor(int v1,int v2,int v3)
{
    if (v1>v2 && v1>v3)
        return v1;
    else
        if (v2>v3)
            return v2;
        else
            return v3;
}
}

```

La metodología es similar al problema anterior. Ahora no tenemos grados para repartir, por lo que implementaremos el siguiente algoritmo:

Consideraremos que el partido que obtuvo más votos le corresponde una barra de 400 píxeles de largo y los otros dos partidos se les entregará en forma proporcional. Lo primero que hacemos es obtener el mayor de los tres valores ingresados por teclado, para esto implementamos un método privado que retorne el mayor de tres valores enteros:

```

private int retornarMayor(int v1,int v2,int v3)
{
    if (v1>v2 && v1>v3)
        return v1;
    else
        if (v2>v3)
            return v2;
        else
            return v3;
}

```

El el método paint llamamos al método retornarMayor:

```

int mayor=retornarMayor(v1,v2,v3);

```


La ecuación para obtener el largo de la barra será:
 $\text{largo} = \text{votos del partido} / \text{votos del partido con mas votos} * 400 \text{ píxeles}.$
Como podemos ver esta división generará un valor menor a uno salvo para el partido que tiene más votos (en este caso la división genera el valor 1) luego multiplicamos por 400.
Comenzamos a calcular el largo de cada rectángulo:

```
int largo1=v1*400/mayor;
```

Como podemos ver primero multiplicamos por 400 y lo dividimos por el mayor de los tres valores ingresados.

Nuevamente primero multiplicamos y luego dividimos con el objetivo que el resultado de la división no nos redondee a cero.

El primer trozo lo graficamos a partir de la columna 100, fila 250 y con un ancho indicado en la variable largo1 (el alto de la barra es de 40 píxeles):

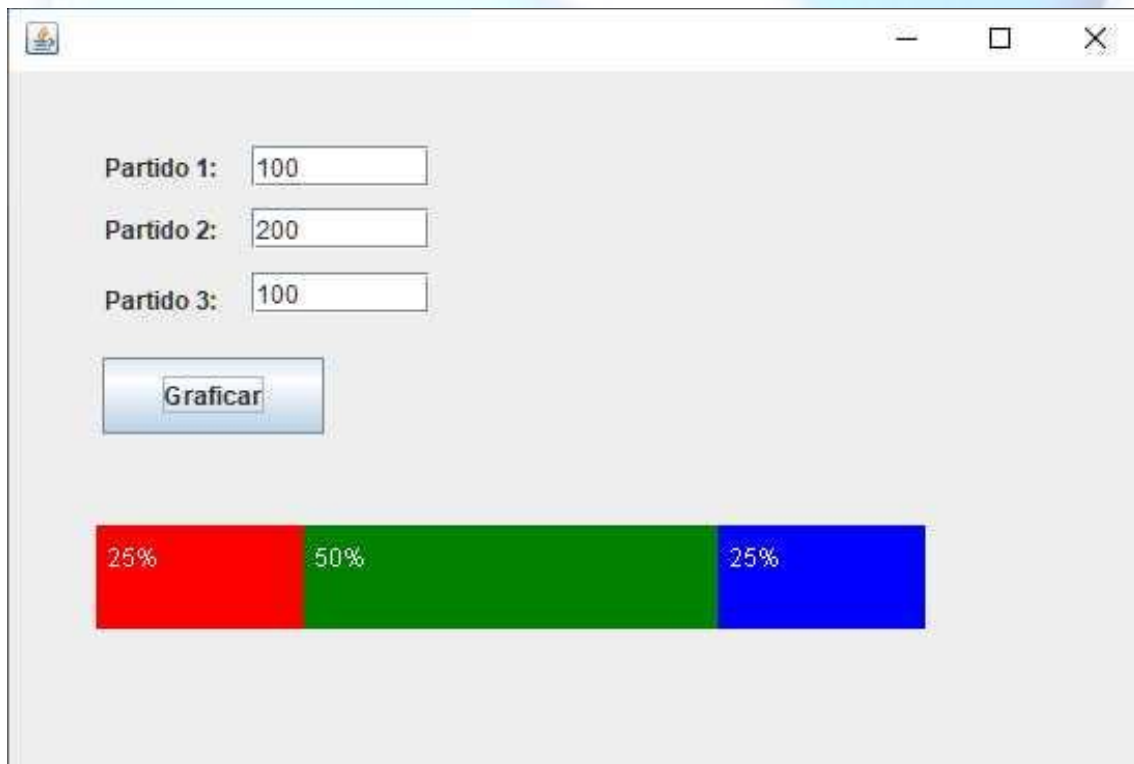
```
g.setColor(new Color(255,0,0));  
g.fillRect(100,250,largo1,40);  
g.drawString("Partido 1", 10, 270);
```

En forma similar graficamos los otros dos trozos de barra:

```
g.setColor(new Color(0,128,0));  
g.fillRect(100,300,largo2,40);  
g.drawString("Partido 2", 10, 320);  
  
g.setColor(new Color(0,0,255));  
g.fillRect(100,350,largo3,40);  
g.drawString("Partido 3", 10, 370);
```

Problema propuesto

1. Implementar un gráfico estadístico de tipo "Barra Porcentual":



Solución

```
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
```

```
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;
```

```
public class BarraPorcentual extends JFrame {

    private JPanel contentPane;
    private JTextField tf1;
    private JTextField tf2;
    private JTextField tf3;
    private boolean bandera=false;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    BarraPorcentual frame = new BarraPorcentual();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public BarraPorcentual()
    {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 800, 600);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblPartido = new JLabel("Partido 1:");
        lblPartido.setBounds(46, 39, 61, 14);
        contentPane.add(lblPartido);

        JLabel lblPartido_1 = new JLabel("Partido 2:");
        lblPartido_1.setBounds(46, 69, 61, 14);
        contentPane.add(lblPartido_1);
    }
}
```

```

JLabel lblPartido_2 = new JLabel("Partido 3:");
lblPartido_2.setBounds(46, 103, 61, 14);
contentPane.add(lblPartido_2);

tf1 = new JTextField();
tf1.setBounds(117, 36, 86, 20);
contentPane.add(tf1);
tf1.setColumns(10);

tf2 = new JTextField();
tf2.setBounds(117, 66, 86, 20);
contentPane.add(tf2);
tf2.setColumns(10);

tf3 = new JTextField();
tf3.setBounds(117, 97, 86, 20);
contentPane.add(tf3);
tf3.setColumns(10);

JButton btnGraficar = new JButton("Graficar");
btnGraficar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        bandera=true;
        repaint();
    }
});
btnGraficar.setBounds(45, 138, 107, 37);
contentPane.add(btnGraficar);
}

```

```

public void paint(Graphics g)
{
    super.paint(g);
    if (bandera==true)
    {
        String s1=tf1.getText();
        String s2=tf2.getText();
        String s3=tf3.getText();
        int v1=Integer.parseInt(s1);
        int v2=Integer.parseInt(s2);
        int v3=Integer.parseInt(s3);
        int suma=v1+v2+v3;
        int largo1=v1*400/suma;
        int largo2=v2*400/suma;
        int largo3=v3*400/suma;
        int porc1=v1*100/suma;
        int porc2=v2*100/suma;
        int porc3=v3*100/suma;

        g.setColor(new Color(255,0,0));
        g.fillRect(50,250,largo1,50);
        g.setColor(new Color(255,255,255));
        g.drawString(porc1+"%",55,270);

        g.setColor(new Color(0,128,0));
        g.fillRect(50+largo1,250,largo2,50);
        g.setColor(new Color(255,255,255));
        g.drawString(porc2+"%",55+largo1,270);
    }
}

```

```
g.setColor(new Color(0,0,255));  
g.fillRect(50+largo1+largo2,250,largo3,50);  
g.setColor(new Color(255,255,255));  
g.drawString(porc3+"%",55+largo1+largo2,270);  
  
}  
}  
}
```



JDBC con MySQL

JDBC son las siglas en ingles de Java Database Connectivity. Es un conjunto de clases que nos permite acceder a diversos gestores de bases de datos en forma transparente.

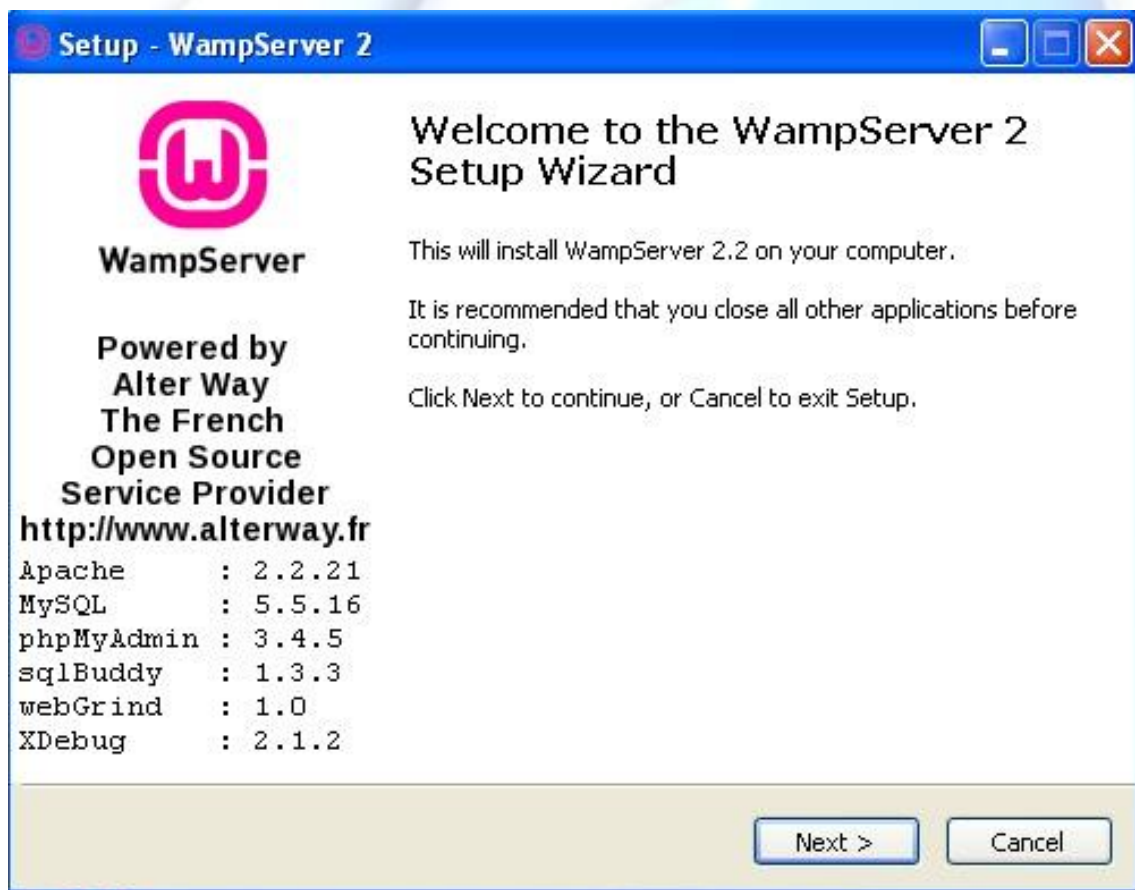
Veremos como conectarnos con el motor de base de datos MySQL.

INSTALACIÓN DE MYSQL INTEGRADO EN EL WAMP SERVER

Utilizaremos esta herramienta en lugar de descargar solo el MySQL con la finalidad de facilitar la instalación y configuración del motor de base de datos (la instalación de esta herramienta es sumamente sencilla), además utilizaremos otra software que provee el WampServer que es el PhpMyAdmin que nos facilitará la creación de la base de datos.

Procedemos a descargar el WampServer de la siguiente página: [aquí](#).

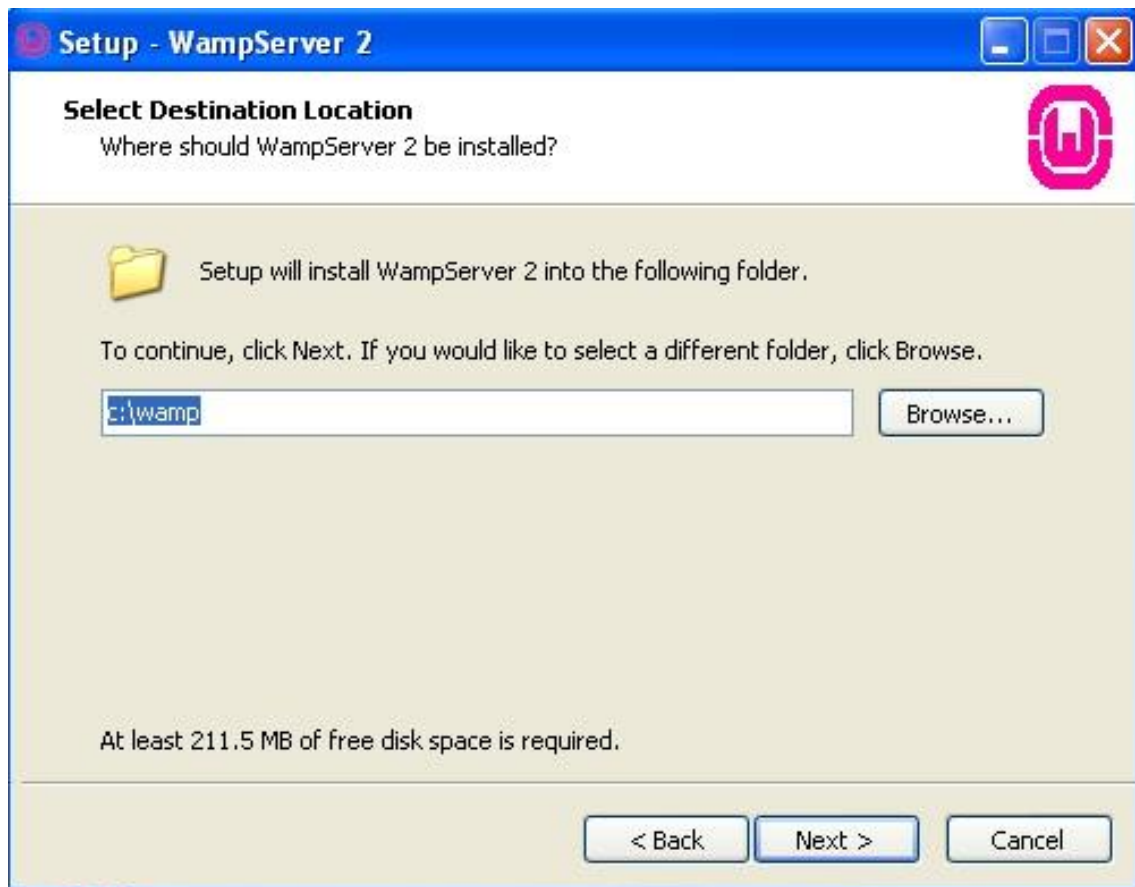
Luego de descargarlo procedemos a ejecutar el instalador:



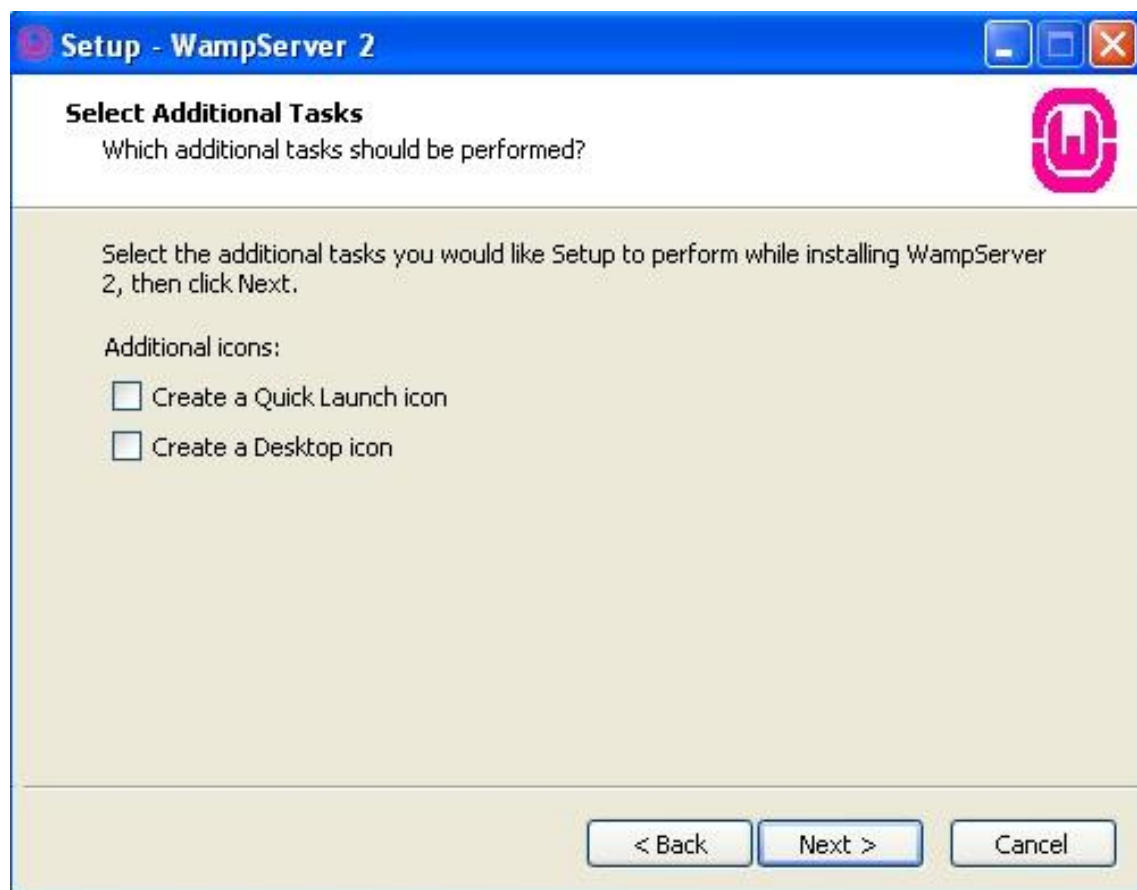
Aceptamos los términos y condiciones:



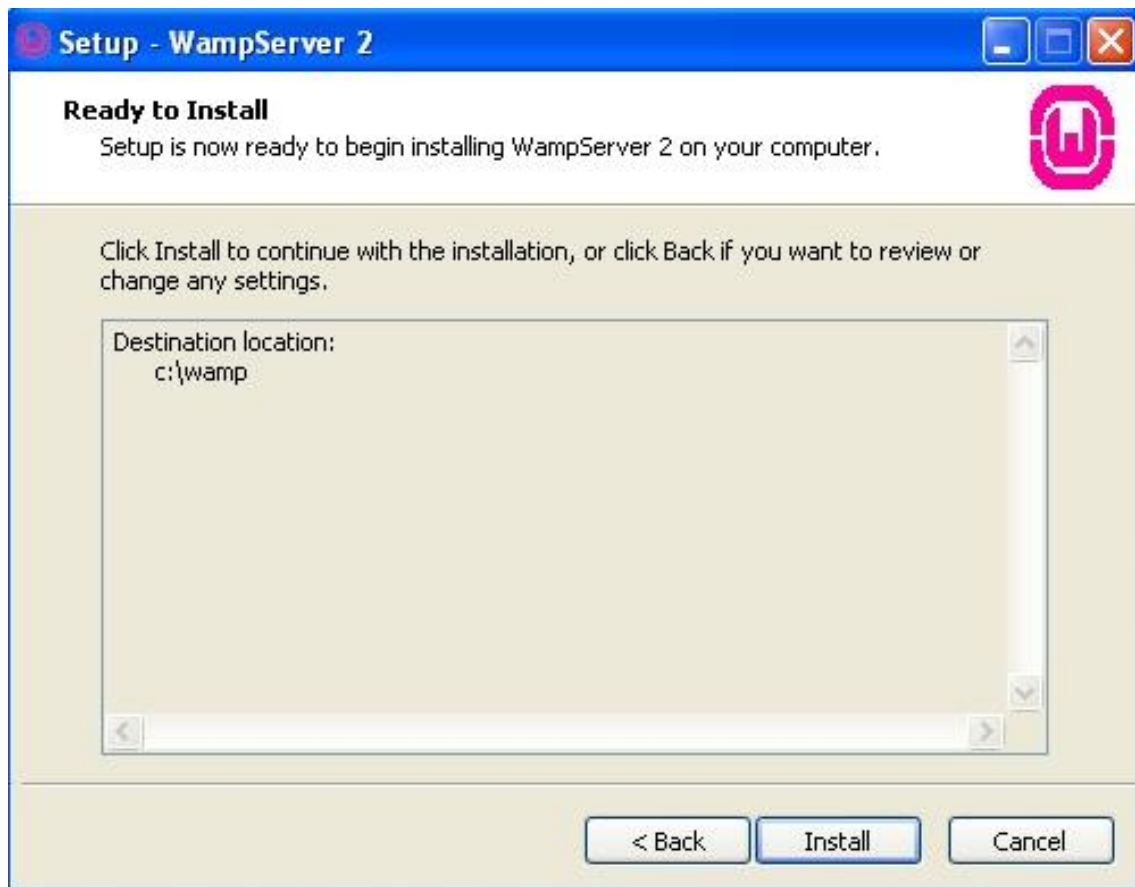
Elegimos el directorio donde se instalará":



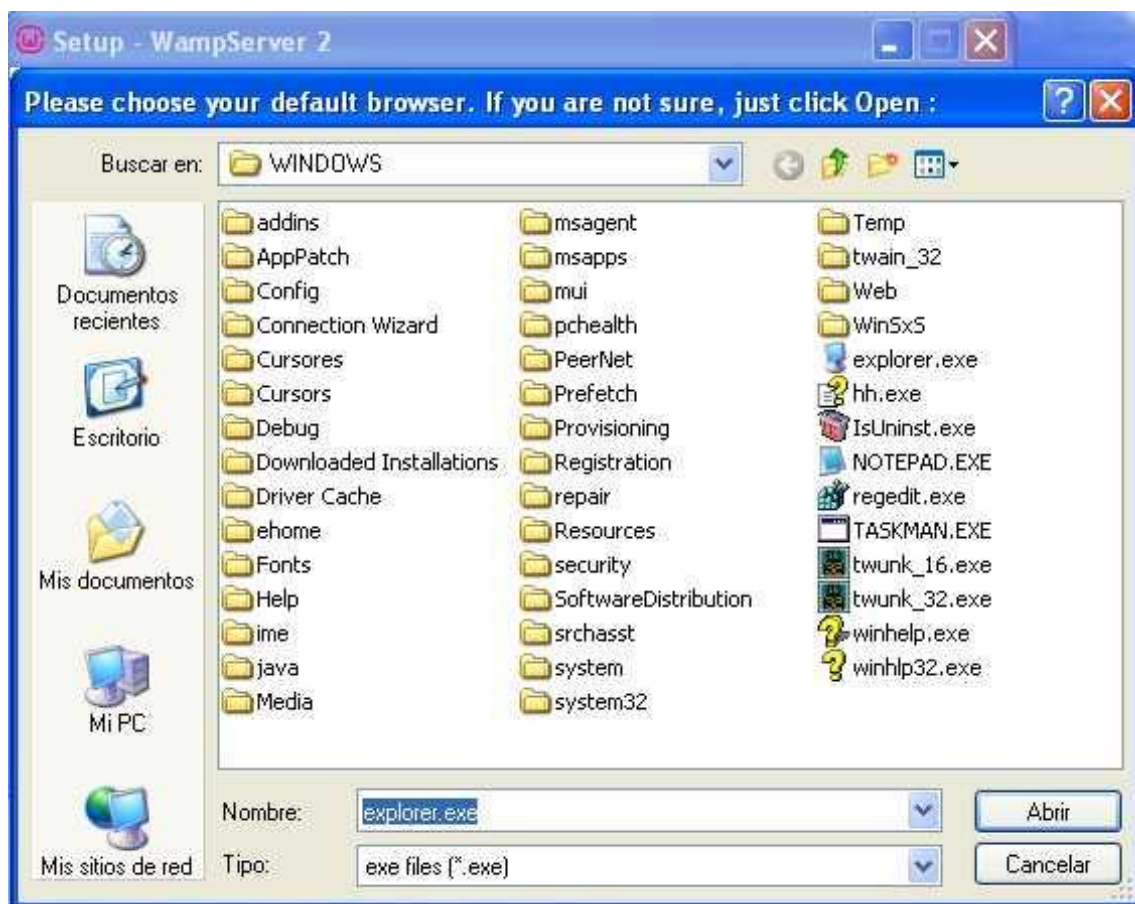
Indicamos si queremos que se cree un ícono en el escritorio:



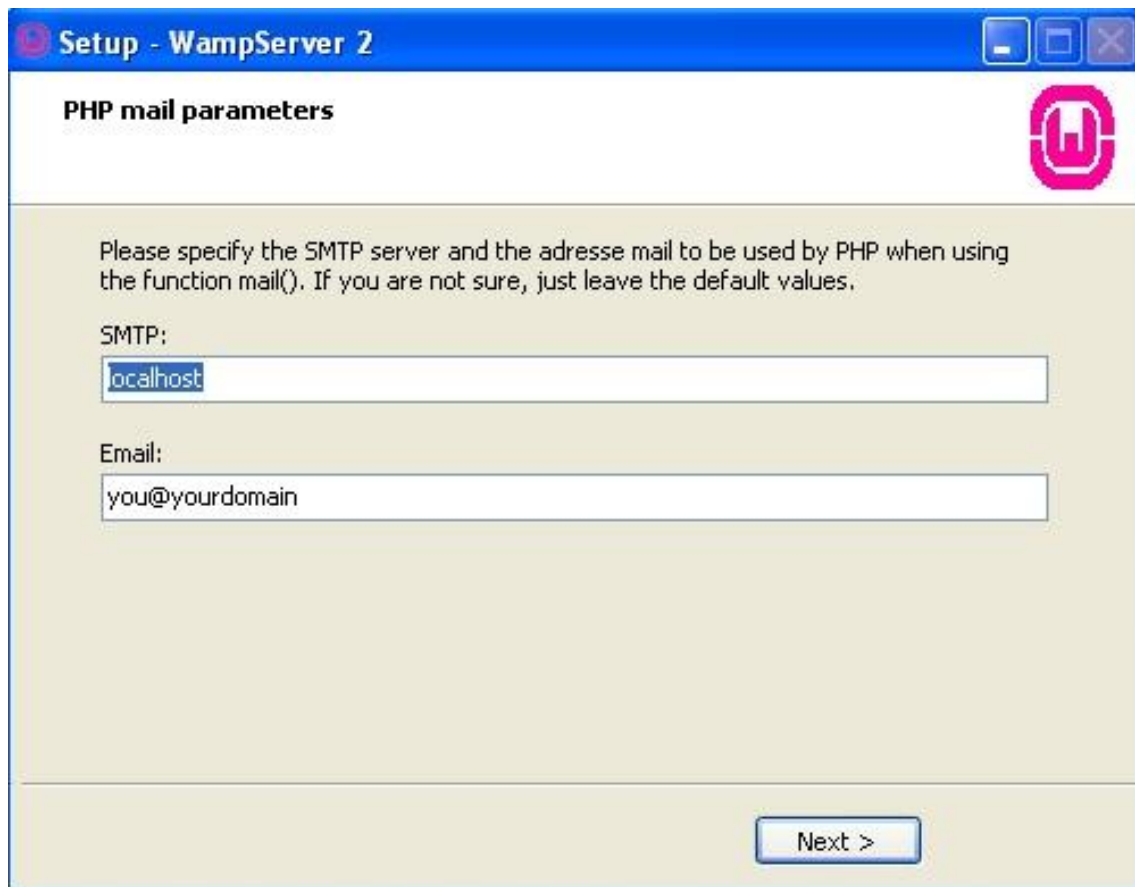
Procedemos a presionar el botón "Install":



Luego de instalarse nos solicita que navegador abrirá por defecto cuando ejecutemos el PhpMyAdmin (para la creación de la base de datos de MySQL):



En el siguiente diálogo dejamos los datos por defecto:



Setup - WampServer 2

PHP mail parameters

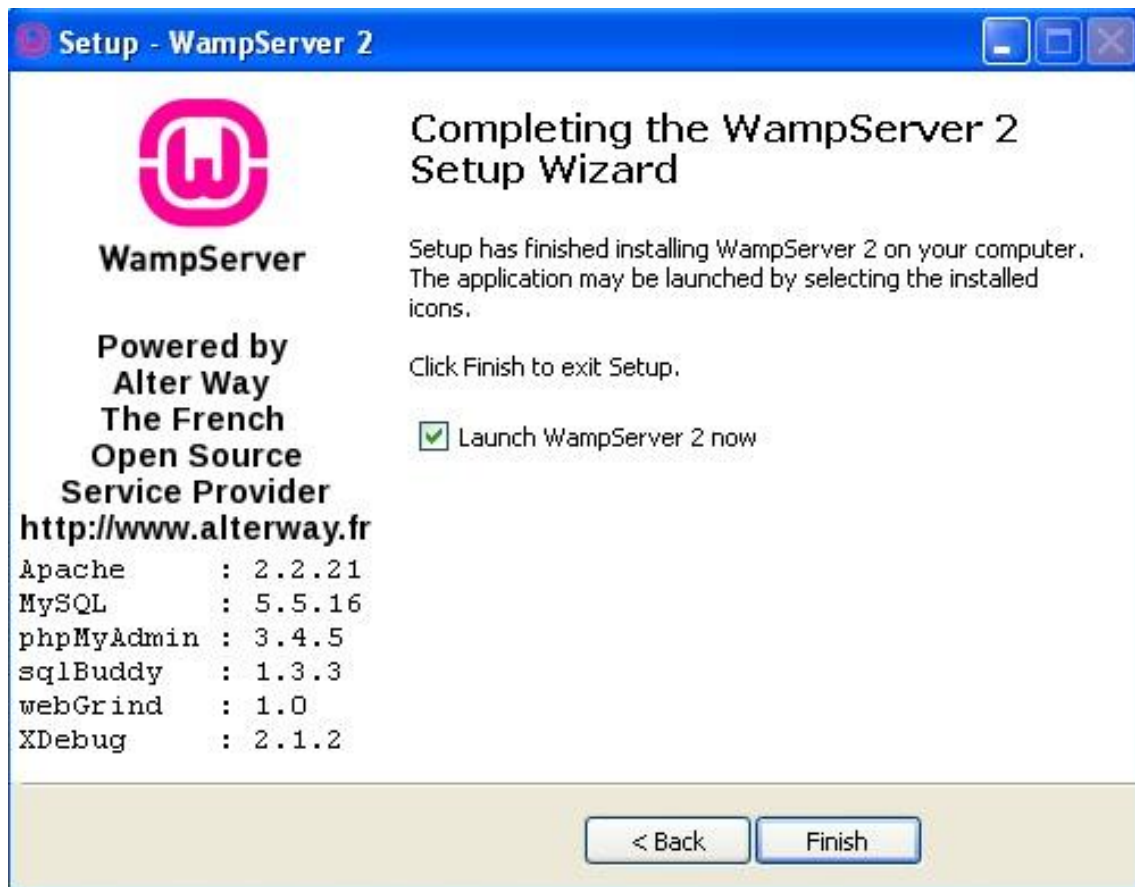
Please specify the SMTP server and the adresse mail to be used by PHP when using the function mail(). If you are not sure, just leave the default values.

SMTP:
localhost

Email:
you@yourdomain

Next >

Finalmente aparece el diálogo final donde se nos informa que se iniciará el WampServer (es decir que se cargará en memoria entre otras cosas el MySQL) :



Ahora podemos ver el iconos del WampServer en la bandeja del sistema de Windows (si se encuentra en color verde significa que el MySQL está ejecutándose correctamente):

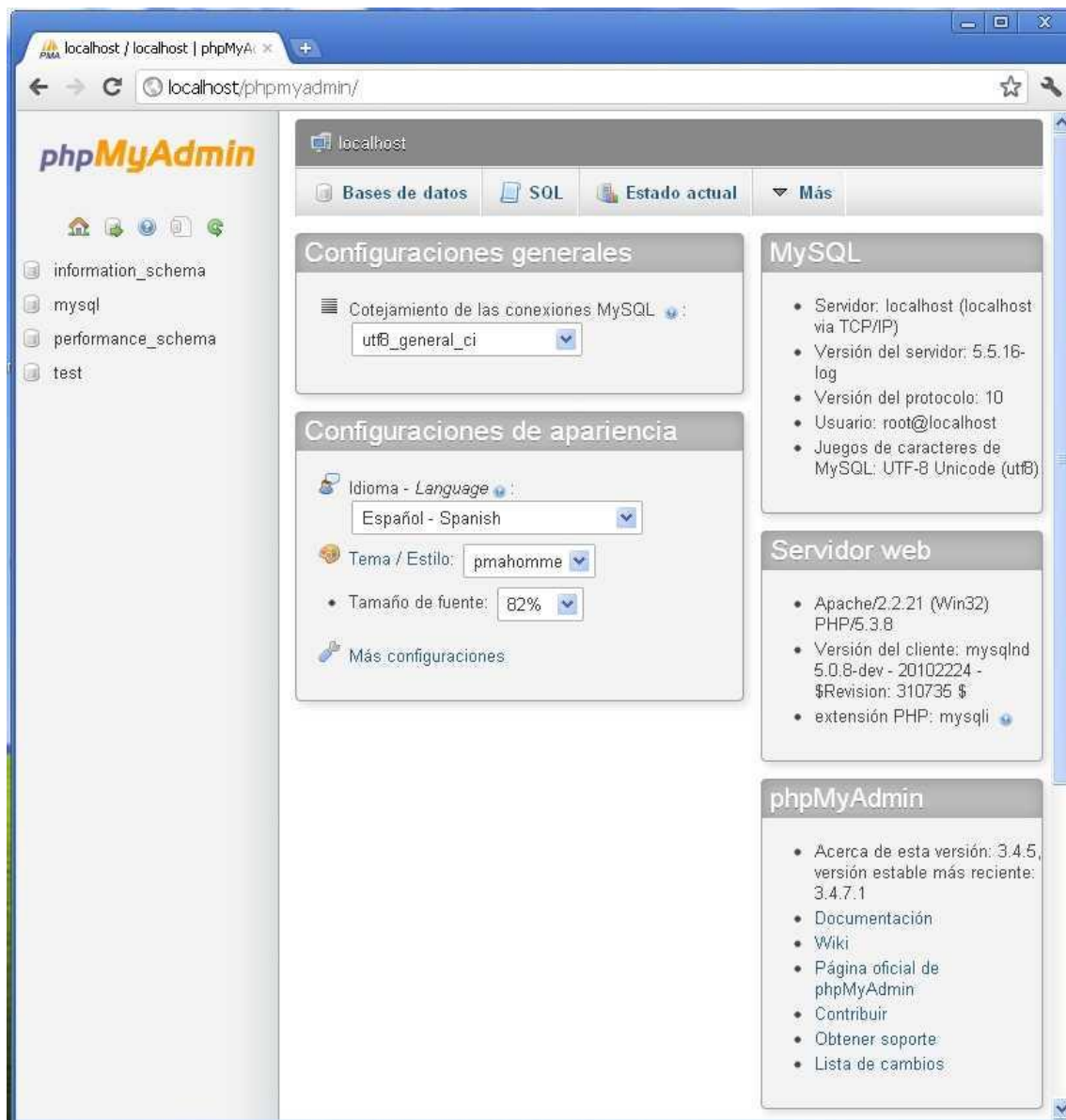


EJECUCIÓN DEL PHPMYADMIN PARA LA CREACIÓN DE LA BASE DE DATOS.

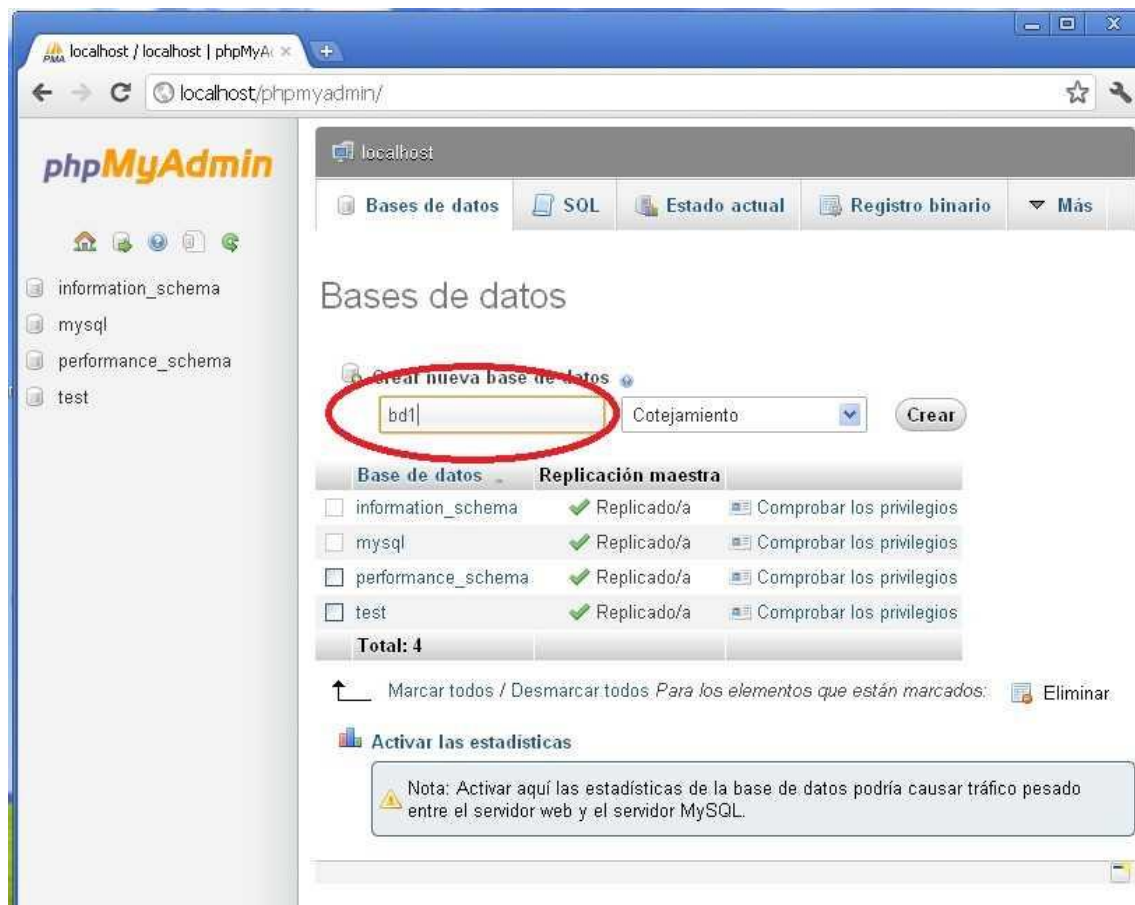
Haciendo clic sobre el ícono de la bandeja del sistema aparece un menú que nos permite lanzar el PhpMyAdmin para crear la base de datos de MySQL:



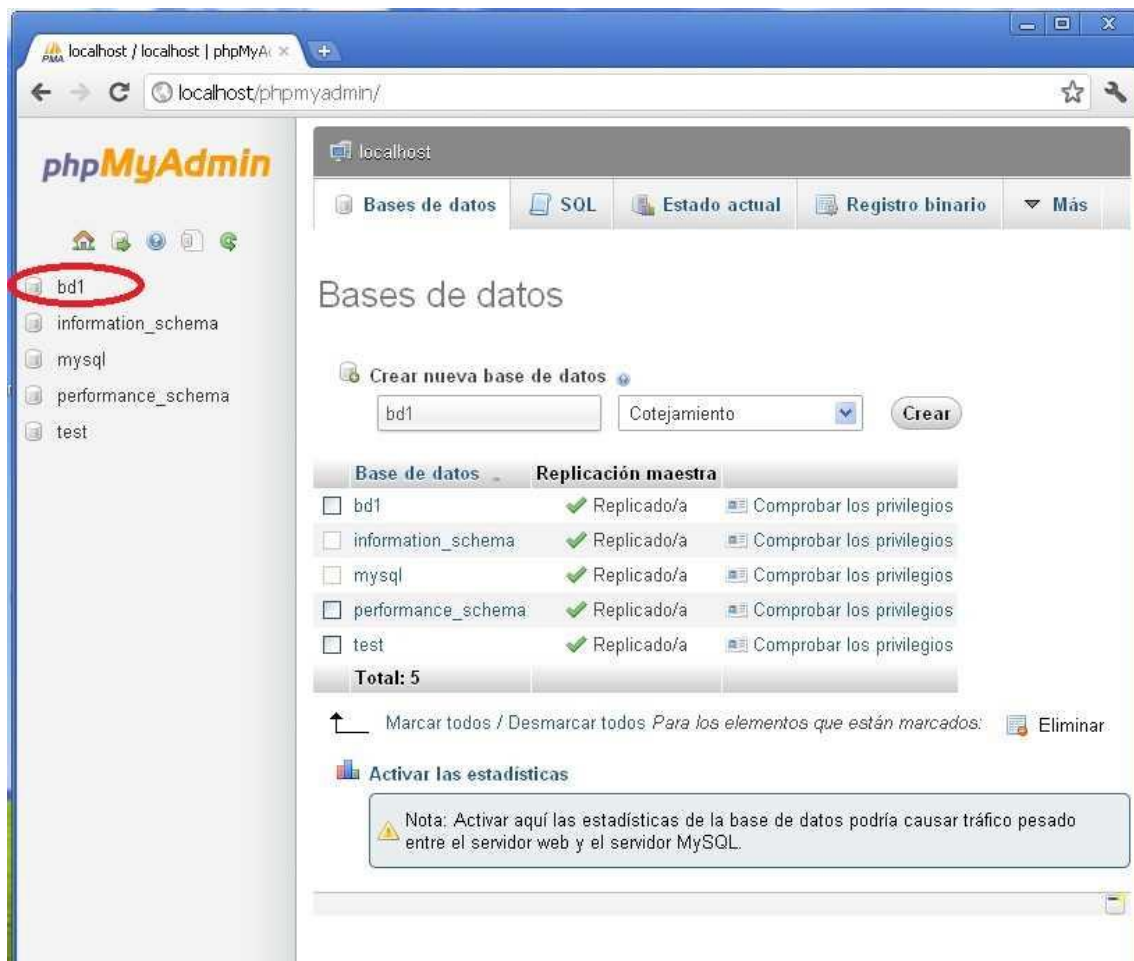
El PhpMyAdmin es un programa web que nos permite administrar las bases de datos del MySQL:



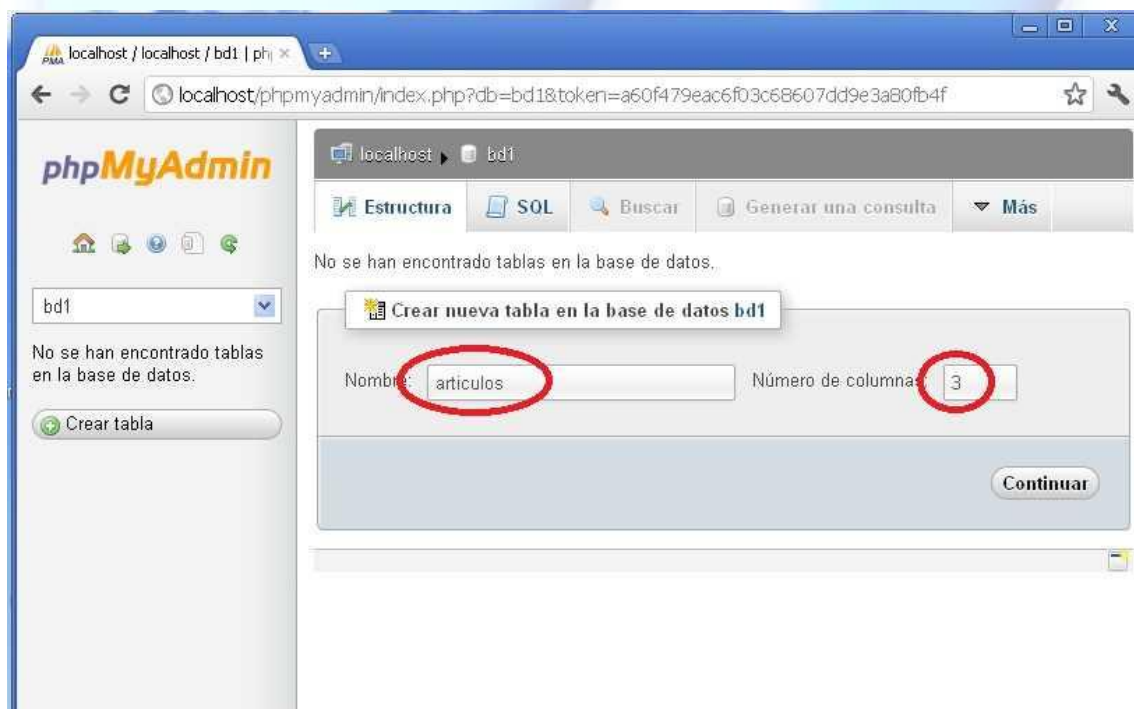
Seleccionamos la pestaña "Base de datos" y donde dice "Crear nueva base de datos" especificamos que nuestra base de datos se llamará "bd1":



Presionamos el botón "crear" y con esto ya tenemos nuestra base de datos creada:



Luego de seleccionar la base de datos "bd1" que figura a la izquierda procedemos a crear la primer tabla que contendrá (crearemos una tabla llamada "articulos" y que tendrá tres campos):



En la tabla "articulos" definimos el campo "codigo" de tipo int (este campo será el "primary key" y auto_increment lo tildamos para que el código se genere automáticamente), el segundo campo es la descripción que es de tipo varchar con un máximo de 50 caracteres y por último el campo precio que es de tipo float.

Luego de especificar los tres campos en la parte inferior de la misma ventana aparece un botón llamada "Guardar" para confirmar la estructura de la tabla:

The screenshot shows a 'Crear tabla' (Create Table) window. At the top, the table name is 'articulos'. Below, the 'Estructura' (Structure) section defines three columns:

Columna	Tipo	Longitud/Valores*1	Predeterminado2	Cotejamiento	Atributos	Nulo	Índice	AUTO_INCREMENT	Comentarios
codigo	INT		Ninguno			<input checked="" type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>	
descripción	VARCHAR	50	Ninguno			<input type="checkbox"/>	---	<input type="checkbox"/>	
precio	FLOAT		Ninguno			<input type="checkbox"/>	---	<input type="checkbox"/>	

Below the structure, there are sections for 'Comentarios de la tabla:', 'Motor de almacenamiento:' (set to InnoDB), and 'Cotejamiento:'. At the bottom, there is a 'definición de la PARTICIÓN:' section. The 'Guardar' (Save) button is highlighted with a red box, and it is followed by 'O Agregar 1 columna(s)' and a 'Continuar' button.

En el lado izquierdo del navegador podemos ver ahora que la base de datos "bd1" tiene una tabla llamada "articulos"



Hasta acá lo que nos ayuda el PhpMyAdmin (es decir creación de la base de datos y la tabla), de ahora en más todas las otras actividades las desarrollaremos desde nuestro programa en java (poblar o insertar datos, listar registros, consultar, modificar y borrar datos)

DESCARGA DEL DRIVER PARA PERMITIR CONECTAR NUESTRO PROGRAMA JAVA CON EL MYSQL

Como última actividad de configuración previo a implementar los programas en Java para acceder a MySQL es la descarga del Driver que nos permita conectarnos con la base de datos. El Driver lo podemos descargar de la página: [aquí](#)

Podemos descomprimir el archivo mysql-connector-java-5.1.18.zip que acabamos de descargar.

Luego veremos que en nuestro programa en java haremos referencias al archivo mysql-connector-java-5.1.18-bin.jar (que es el Driver propiamente dicho)

Muchas gracias hasta la próxima clase.

Alsina 16 [B1642FNB] San Isidro | Pcia. De Buenos Aires |Argentina |

TEL.: [011] 4742-1532 o [011] 4742-1665 |

www.institutosanisidro.com.ar info@institutosanisidro.com.ar