



**LICENCIATURA EN INGENIERÍA DE TECNOLOGÍAS DE LA
INFORMACIÓN E INNOVACIÓN DIGITAL**

Desarrollo de software

Unidad III.

Estructuras de datos básicas

Actividad U3Ejercicio Guiado Swing

Materia

Estructura de datos avanzadas

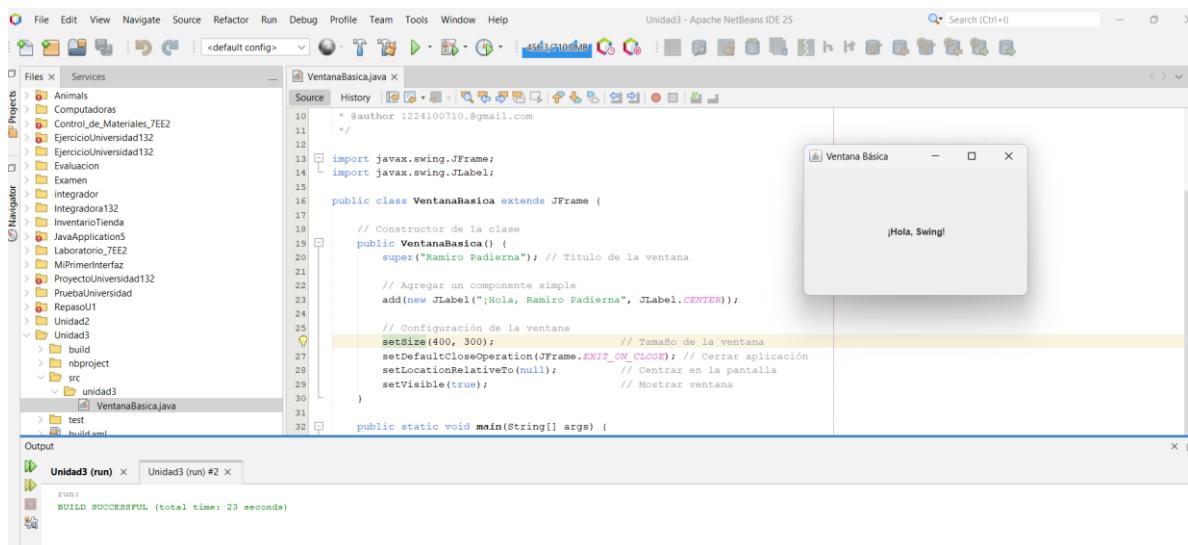
Alumno:

Ramiro Padierna Delgado **1224100710**

Profesor:

Gabriel Barrón Rodríguez

Creación de una ventana básica swing



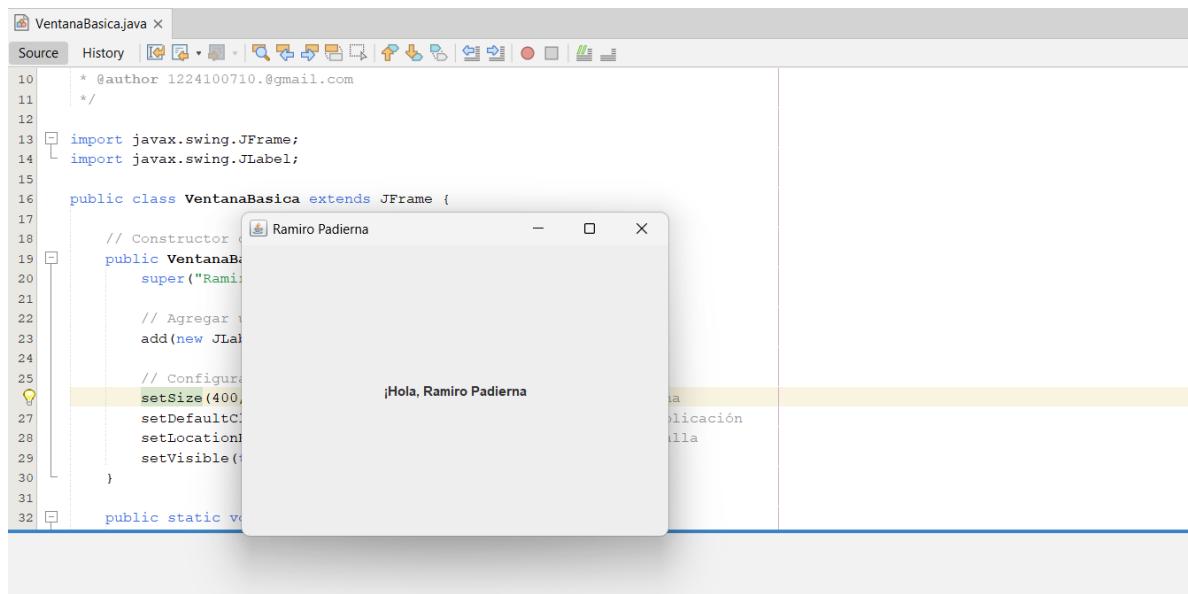
The screenshot shows the Apache NetBeans IDE interface. The left sidebar displays a project tree with several Java projects. The main area shows the code editor for a file named 'VentanaBasica.java'. The code defines a simple Swing application:

```
10 * @author 1224100710@gmail.com
11 */
12
13 import javax.swing.JFrame;
14 import javax.swing.JLabel;
15
16 public class VentanaBasica extends JFrame {
17
18     // Constructor de la clase
19     public VentanaBasica() {
20         super("Ramiro Padierna"); // Título de la ventana
21
22         // Agregar un componente simple
23         add(new JLabel("Hola, Ramiro Padierna", JLabel.CENTER));
24
25         // Configuración de la ventana
26         setSize(400, 300); // Tamaño de la ventana
27         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // Cerrar aplicación
28         setLocationRelativeTo(null); // Centrar en la pantalla
29         setVisible(true); // Mostrar ventana
30     }
31
32     public static void main(String[] args) {
33
34     }
35 }
```

To the right of the code editor, a preview window titled 'Ventana Básica' shows a simple window with the text '¡Hola, Swing!'. Below the code editor, the 'Output' tab shows the build log:

```
run:
BUILD SUCCESSFUL (total time: 23 seconds)
```

Realice modificaciones en el nombre al igual que el tamaño de la pantalla



The screenshot shows the Apache NetBeans IDE interface with the same project structure and code editor as the previous screenshot. However, the preview window now displays a larger window titled 'Ramiro Padierna' with the text '¡Hola, Ramiro Padierna'. This indicates that the code has been modified to change the window title and size.

Se agrego un nuevo código (JPanel) para hacer que la ventana se muestre en color azul

```

13 import java.awt.Color;
14 import javax.swing.JFrame;
15 import javax.swing.JLabel;
16 import javax.swing.JPanel;
17
18 public class VentanaBasica extends JFrame {
19
20     // Constructor de la clase
21     public VentanaBasica() {
22         super("Ramiro Padierna"); // Título de la ventana
23
24         // Agregar un componente simple
25         add(new JLabel("Hola, Ramiro Padierna", JLabel.CENTER));
26
27         // Configuración de la ventana
28         setSize(400, 300); // Tamaño de la ventana
29         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // Cerrar aplicación
30         setLocationRelativeTo(null); // Centrar en la pantalla
31         setVisible(true); // Mostrar ventana
32
33         // Crear un panel y ponerle color de fondo azul
34         JPanel panelAzul = new JPanel();
35         panelAzul.setBackground(Color.BLUE);
36         // Agregar el panel a la ventana
37         add(panelAzul);
38     }
39
40     public static void main(String[] args) {
41         // Crear y mostrar la ventana
42         new VentanaBasica();
43     }

```

Agregamos un TextFile que contenga el nombre.

```

32         panelAzul.setBackground(Color.BLUE);
33         panelAzul.setLayout(new FlowLayout());
34
35         JLabel etiqueta = new JLabel("Nombre: ");
36         etiqueta.setForeground(Color.WHITE); // Texto blanco
37         JTextField campoNombre = new JTextField(15);
38
39         panelAzul.add(etiqueta);
40         panelAzul.add(campoNombre);
41
42         add(panelAzul, BorderLayout.CENTER); // Panel azul en el centro
43
44         // Configuración de la ventana
45         setSize(400, 200);
46         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
47         setLocationRelativeTo(null);
48         setVisible(true);
49
50         // Configuración de la ventana
51         setSize(400, 300); // Tamaño de la ventana
52         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // Cerrar aplicación
53         setLocationRelativeTo(null); // Centrar en la pantalla
54         setVisible(true); // Mostrar ventana
55
56     }
57
58     public static void main(String[] args) {
59         // Crear y mostrar la ventana
60         new VentanaBasica();
61     }

```

Se agregaron nuevos campos que fueron Apellidos y Ciudad de nacimiento ambos JTextField

```

// --- Ciudad ---
JPanel filaCiudad = new JPanel(new FlowLayout(FlowLayout.LEFT));
filaCiudad.setBackground(Color.BLUE);
JLabel etiquetaCiudad = new JLabel("Ciudad de nacimiento: ");
etiquetaCiudad.setForeground(Color.WHITE);
JTextField campoCiudad = new JTextField(8);
filaCiudad.add(etiquetaCiudad);
filaCiudad.add(campoCiudad);

// Agregar filas al panel azul
panelAzul.add(filaNombre);
panelAzul.add(filaApellidos);
panelAzul.add(filaCiudad);

// Agregar panel azul al JFrame
add(panelAzul, BorderLayout.CENTER);

// Configuración de la ventana
setSize(400, 300); // Tamaño de la ventana
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // Cerrar aplicación
 setLocationRelativeTo(null); // Centrar en la pantalla
setVisible(true); // Mostrar ventana
}

public static void main(String[] args) {
    // Crear y mostrar la ventana
    new VentanaBasica();
}

```

Se agregaron nuevas cosas como JComboBox, JRadioButtons y un JOptionPane que mandara un mensaje, todo esto para lograr algo más mejorado y dejar de lado lo básico.

```

VentanaBasica.java X
Source History
28     panelAzul.setBackground(Color.BLUE);
29     GridBagConstraints gbc = new GridBagConstraints();
30     gbc.insets = new Insets(5,5,5,5);
31     gbc.fill = GridBagConstraints.HORIZONTAL;
32
33     // Nombre
34     JLabel etiquetaNombre = new JLabel("Nombre: ");
35     etiquetaNombre.setForeground(Color.WHITE);
36     JTextField campoNombre = new JTextField(10);
37     gbc.gridx = 0; gbc.gridy = 0;
38     panelAzul.add(etiquetaNombre, gbc);
39     gbc.gridx = 1; gbc.gridy = 0;
40     panelAzul.add(campoNombre, gbc);
41
42     // Apellidos
43     JLabel etiquetaApellidos = new JLabel("Apellidos: ");
44     etiquetaApellidos.setForeground(Color.WHITE);
45     JTextField campoApellidos = new JTextField(10);
46     gbc.gridx = 0; gbc.gridy = 1;
47     panelAzul.add(etiquetaApellidos, gbc);
48     gbc.gridx = 1; gbc.gridy = 1;
49     panelAzul.add(campoApellidos, gbc);
50
51     // Municipio (46 de Guanajuato)
52     JLabel etiquetaMunicipio = new JLabel("Municipio: ");
53     etiquetaMunicipio.setForeground(Color.WHITE);
54     String[] municipios = {
55         "Abasolo", "Acámbaro", "Allende (San Miguel de Allende)", "Apaseo el Alto",
56         "Apaseo el Grande", "Atarjea", "Celaya", "Manuel Doblado",
57         "Comonfort", "Coroneo", "Cortazar", "Cuerámaro", "Doctor Mora",
58     };
59
60     JComboBox comboBoxMunicipio = new JComboBox(municipios);
61     gbc.gridx = 1; gbc.gridy = 2;
62     panelAzul.add(comboBoxMunicipio, gbc);
63
64     // Fecha de Nacimiento
65     JLabel etiquetaFechaNacimiento = new JLabel("Fecha de Nacimiento: ");
66     etiquetaFechaNacimiento.setForeground(Color.WHITE);
67     JTextField campoFechaNacimiento = new JTextField(10);
68     gbc.gridx = 0; gbc.gridy = 3;
69     panelAzul.add(etiquetaFechaNacimiento, gbc);
70     gbc.gridx = 1; gbc.gridy = 3;
71     panelAzul.add(campoFechaNacimiento, gbc);
72
73     // Género
74     JPanel panelGenero = new JPanel();
75     panelGenero.setLayout(new GridLayout(1, 3));
76     panelGenero.add(new JRadioButton("Masculino"));
77     panelGenero.add(new JRadioButton("Femenino"));
78     panelGenero.add(new JRadioButton("Otro"));
79     gbc.gridx = 1; gbc.gridy = 4;
80     panelAzul.add(panelGenero, gbc);
81
82     // Botón Registrar
83     JButton botonRegistrar = new JButton("Registrar");
84     gbc.gridx = 1; gbc.gridy = 5;
85     panelAzul.add(botonRegistrar, gbc);
86
87     // Configuración de la ventana
88     setSize(400, 300); // Tamaño de la ventana
89     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // Cerrar aplicación
90     setLocationRelativeTo(null); // Centrar en la pantalla
91     setVisible(true); // Mostrar ventana
92 }

public static void main(String[] args) {
    // Crear y mostrar la ventana
    new VentanaBasica();
}

```

Y el mensaje lanzado fue el siguiente.

The screenshot shows a Java application window titled "VentanaBasica.java". The code in the editor is for a basic Swing application. It includes sections for setting up a blue panel, adding a name label and field, and adding a last name label and field. It also includes a municipality dropdown, a date of birth date picker, and a gender radio group. A "Registrar" button is present at the bottom. Below the code, there is a "Registro" message box containing an information icon, a message saying "Hola Ramiro Padierna Delgado, eres de San Diego de la Unión, naciste el 13/Agosto/2005 y eres del género Masculino. ¡Saludos!", and an "OK" button.

```

28     panelAzul.setBackground(Color.BLUE);
29     GridBagConstraints gbc = new GridBagConstraints();
30     gbc.insets = new Insets(5,5,5,5);
31     gbc.fill = GridBagConstraints.HORIZONTAL;
32
33     // Nombre
34     JLabel etiquetaNombre = new JLabel("Nombre: ");
35     etiquetaNombre.setForeground(Color.WHITE);
36     JTextField campoNombre = new JTextField(10);
37     gbc.gridx = 0; gbc.gridy = 0;
38     panelAzul.add(etiquetaNombre, gbc);
39     gbc.gridx = 1; gbc.gridy = 0;
40     panelAzul.add(campoNombre, gbc);
41
42     // Apellidos
43     JLabel etiquetaApellidos = new JLabel("Apellidos: ");
44     etiquetaApellidos.setForeground(Color.WHITE);
45     JTextField campoApellidos = new JTextField(10);
46     gbc.gridx = 0; gbc.gridy = 1;
47
48
49
50
51
52
53
54     String[] municipios = {
55         "Abasolo", "Acámbaro", "Allende (San Miguel de Allende)", "Apaseo el Alto",
56         "Apaseo el Grande", "Atarjea", "Celaya", "Manuel Doblado",
57         "Comonfort", "Coroneo", "Cortazar", "Cuerámaro", "Doctor Mora".

```

El autor fue:

```

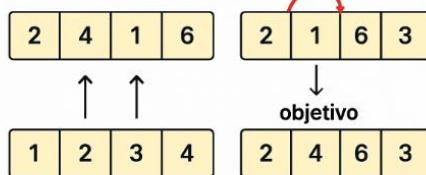
/**
 * Ventana basica Swing para aprender
 * @author Padierna Delgado Ramiro
 * @author 1224100710@gmail.com
 */

```

ALGORITMOS DE ORDENACIÓN Y BÚSQUEDA

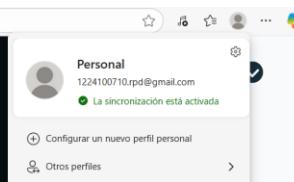
ALGORITMOS DE ORDENACIÓN Y BÚSQUEDA

Ordenación Búsqueda



Definición

- El **ordenamiento** (o clasificación de datos) consiste en organizar una colección de elementos en un orden específico —numérico o alfabético— según una clave.
- Este proceso puede realizarse de forma interna (en memoria principal) o externa (en archivos almacenados).
- La eficiencia de los algoritmos se mide por su número de comparaciones y complejidad temporal.



<https://app.nearpod.com/presentation?pin=E2KSI>

TIPOS DE ORDENACIÓN

Existen dos categorías:

- Métodos directos (básicos)
- Métodos indirectos (avanzados)

4 de 20 Abrir navegador ▲

<https://app.nearpod.com/presentation?pin=E2KSI>

MÉTODOS DIRECTOS (BÁSICOS)

Por intercambio
Compara e intercambia elementos adyacentes.
Sencillo pero ineficiente.

→ Complejidad: $O(n^2)$

Por selección
Busca el elemento menor y lo coloca en su posición definitiva.

→ Complejidad: $O(n^2)$

Por inserción
Inserta cada elemento en su posición correcta dentro de la sublista ya ordenada.

→ Complejidad: $O(n^2)$

5 de 20 Abrir navegador ▲

https://app.nearpod.com/presentation?pin=E2KSI

ORDENAMIENTO POR INTERCAMBIO

Método de ordenamiento por intercambio

El algoritmo compara e intercambia elementos adyacentes que están fuera de orden.

Buscar el número 9

Paso 2: Nueva sublista: 9 11

Elemento encontrado

7 de 20

Abrir navegador ▲

https://app.nearpod.com/presentation?pin=E2KSI

ORDENAMIENTO POR SELECCIÓN

Método de ordenamiento por selección

El algoritmo busca el menor y lo coloca en su posición definitiva.

Paso 2: Nueva sublista: 1

8 de 20

Abrir navegador ▲

Métodos Directos

ACERTASTE 5 DE 5

100 CORRECTA(S)

0 INCORRECTA(S)

0 SIN RESPUESTA

Mis respuestas

¿Qué es un algoritmo de ordenamiento por intercambio?

Un método que organiza elementos mediante comparación y intercambio ✓

9 de 20 Abrir navegador ▲

MÉTODOS INDIRECTOS

Shell
Variante mejorada de la inserción con incrementos decrecientes, que reduce comparaciones.

→ Complejidad media: $O(n^{1/2})$

Quicksort (ordenación rápida)
Utiliza la técnica divide y vencerás: selecciona un pivote para dividir la lista en dos sublistas (menores y mayores).

pivote

1, 2 5, 7

→ Complejidad media: $O(n \log n)$

→ Peor caso: $O(n^2)$

10 de 20 Abrir navegador ▲

The screenshot shows a presentation slide from the Nearpod app. The title 'COSAS POR CONOCER' is displayed in large white text on a black background. Below the title, there is yellow text: 'Otros métodos como fusión (MergeSort) y montículo (HeapSort) también alcanzan complejidad O(n log n) y se emplean en listas muy grandes.' At the bottom left, it says '11 de 20'. At the bottom right, there is a link 'Abrir navegador ▲'.

COSAS POR CONOCER

Otros métodos como fusión (MergeSort) y montículo (HeapSort) también alcanzan complejidad O($n \log n$) y se emplean en listas muy grandes.

11 de 20

Abrir navegador ▲



Mis respuestas

¿Cuál es la principal ventaja de usar Quicksort?

- Es el más estable
- Es fácil de implementar
- No requiere memoria adicional
- Es rápido en la práctica ✓

12 de 20

Abrir navegador ▲

A screenshot of a Nearpod slide. The title 'ORDENACIÓN DE OBJETOS' is displayed in large white text on a black background. Below the title, there is a list of bullet points in yellow text:

- Para ordenar objetos en Java, se define un criterio de comparación mediante una interfaz (Comparable) que implementa métodos como menorQue() o mayorQue().
- Una vez definidos, se puede aplicar cualquier algoritmo de ordenación (por ejemplo, burbuja) adaptado a objetos.

The slide has a dark theme with blue circular arrows for navigation. The bottom navigation bar shows '13 de 20' and 'Abrir navegador ▲'.

The screenshot shows a presentation slide with a black background and white text. The title 'BÚSQUEDAS' is at the top in large, bold, white capital letters. Below it is a subtitle in smaller white text: 'La búsqueda consiste en localizar un elemento que cumpla una condición dada (clave).'. At the bottom left is the text '14 de 20' and at the bottom right is 'Abrir navegador ▲'. The browser interface is visible around the slide.

BÚSQUEDAS

La búsqueda consiste en localizar un elemento que cumpla una condición dada (clave).

14 de 20 Abrir navegador ▲

This screenshot shows another presentation slide with a black background and white text. The title 'BÚSQUEDA secuencial' is at the top in large, bold, white capital letters. Below it is a subtitle: 'Recorre cada elemento hasta encontrar el valor buscado.' followed by two bullet points: '→ Util en listas pequeñas o no ordenadas.' and '→ Complejidad: O(n)'. At the bottom left is the text '15 de 20' and at the bottom right is 'Abrir navegador ▲'. The browser interface is visible around the slide.

BÚSQUEDA secuencial

Recorre cada elemento hasta encontrar el valor buscado.
→ Util en listas pequeñas o no ordenadas.
→ Complejidad: O(n)

15 de 20 Abrir navegador ▲