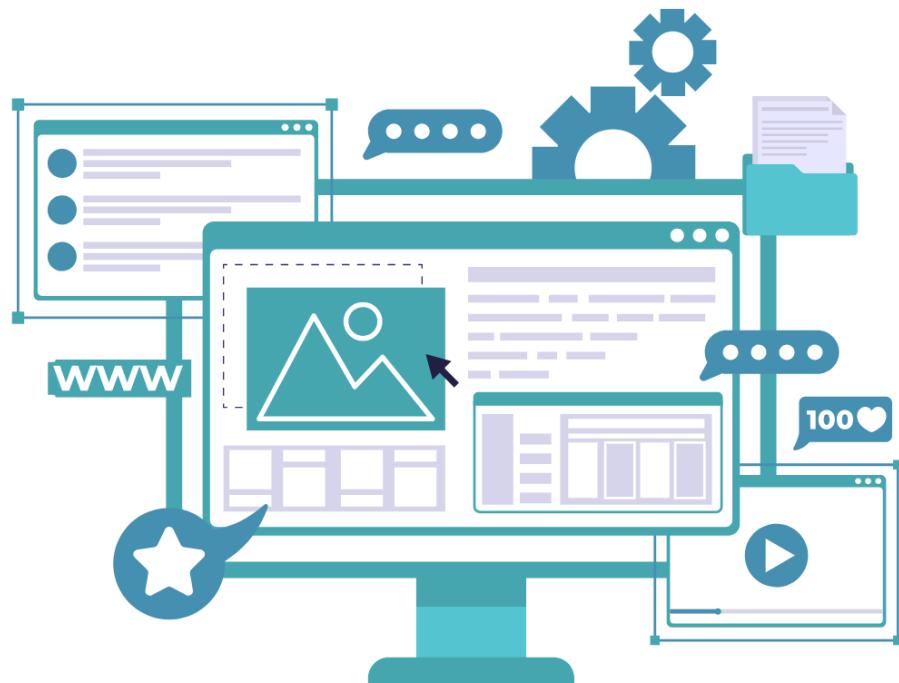




Estructura de Datos



Manual de prácticas

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

Elaborado por:

Ing. Alvarado Ornelas José Guadalupe
 Mtro. Dávila Guerrero Jorge Luis
 Mtra. Hernández Del Rio Andrea Alejandra
 Mtro. Medina Reyes Gerardo Joel
 Ing. Ortega Estrada Gilberto
 Dra. Gemma Alejandra Corona Núñez

Presidente de la Academia.

Mtro. Gerardo Joel Medina Reyes

Responsable del Laboratorio de Programación.

Mtro. Jorge Luis Dávila Guerrero

Jefe del Departamento de Ingenierías.

Dr. Domingo Velázquez Pérez

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

OBJETIVO GENERAL.

El manual de prácticas tiene como objetivo ser un apoyo para el estudiante con respecto a lo visto en clases presenciales. Utilizando el lenguaje de programación en Java, se aplica los conocimientos aprendidos en la clase de Estructura de Datos, donde aplicarán a si mismo sus conocimientos aprendidos en semestres anteriores. Además de tener una mejor comprensión de la programación de estructura de datos, se debe de practicar los conceptos aprendidos durante la clase presencial.

CONSIDERACIONES GENERALES

El estudiante debe cumplir con el Reglamento General de Uso de Laboratorios publicado en el “Compendio de reglamentos del Departamento de Ingeniería”

SEGURIDAD E HIGIENE EN EL USO DEL LABORATORIO

En caso de alguna contingencia (sismo, incendio o cualquier evento que ponga en riesgo su integridad) evacúe el laboratorio inmediatamente, siguiendo las normas de seguridad implementadas en los simulacros.

Así mismo es de suma importancia que las personas que hagan uso de las instalaciones de los laboratorios, conozcan las ubicaciones de los extintores, botiquines de primeros auxilios y salidas de emergencia.

Es importante resaltar los siguientes puntos referentes a la seguridad e higiene que se deben seguir para el uso de laboratorio y que se encuentran plasmados en el Reglamento Interno del laboratorio de Computo:

1. Mantener y dejar limpia su área de trabajo.
2. No arrojar papeles ni basura al piso.
3. No introducir alimentos y bebidas.
4. No fumar.
5. El alumno deberá dejar su mochila y/o bolsa en los estantes designados para los mismos, respetando todo objeto ajeno que allí se encuentre.
6. No utilizar dispositivos de reproducción de música y audio.

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

7. Está prohibido sentarse sobre las mesas de trabajo o pararse en las sillas.
8. El alumno debe comportarse siguiendo el Código de Ética de la Universidad de Guadalajara dentro de las instalaciones del laboratorio, hacer uso apropiado del lenguaje oral y escrito; respetar a sus profesores, compañeras y compañeros de clase.
9. Antes de iniciar las prácticas, el maestro inspeccionará las condiciones físicas del laboratorio y de encontrar situaciones que representen riesgo grave, deberá reportar dicha situación al responsable del laboratorio y/o al asistente o auxiliar del mismo, para que sea corregida, en caso de que no exista la posibilidad de atención inmediata, la práctica será suspendida.
10. Si durante la práctica surgiera una condición que ponga en riesgo grave la seguridad y salud de las personas, equipos, materiales o instalaciones, se procederá a suspender la práctica debiendo informar de la situación al responsable de laboratorio, asistente o auxiliar del mismo, elaborando por escrito el reporte correspondiente.
11. El profesor deberá cumplir con el uso del equipo de protección personal básico de laboratorio. El maestro que no cumpla con estos requisitos no podrá realizar la práctica. El auxiliar notificará la situación al responsable de laboratorio y/o al jefe de departamento quien elaborará un reporte de faltas al reglamento.
12. Es responsabilidad del profesor verificar que antes de iniciar la práctica, todos los alumnos cuenten con el equipo de protección personal y el código de vestimenta necesario para realizar la práctica. El alumno que no cumpla con los requisitos anteriores no podrá realizar la práctica.
13. El profesor deberá asegurarse que los alumnos utilicen adecuadamente el equipo de protección personal durante el desarrollo de la práctica.
14. El profesor llevará un registro de los alumnos que sean observados sin usar su equipo de protección personal o usándolo de manera inadecuada, cada registro contará como una falta al reglamento del laboratorio.
15. La acumulación de 4 faltas al reglamento del laboratorio implica la suspensión para el alumno de la práctica en el semestre y la no acreditación de la misma.

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

16. El profesor deberá permanecer en el laboratorio durante todo el desarrollo de la práctica.

17. Por razones de seguridad y orden está prohibido en el laboratorio:

- Correr.
- Usar sombrero, gorra y/o pañoleta en la cabeza.
- Ingresar personas ajenas a la institución o al grupo que desarrolla la práctica.
- Usar calzado inadecuado.
- Usar el cabello largo (las personas con esta característica deberán recoger su cabello y sujetarlo adecuadamente, como medida de prevención para evitar riesgos).
- Usar pantalón corto o bermuda.
- Y en general todo acto y/o conducta que incite al desorden.

18. Todo alumno que sufra una lesión deberá reportarla al maestro encargado de la práctica y de no encontrarse éste, deberá dirigirse con el responsable de laboratorio y/o asistente del mismo.

19. Todo trabajador universitario que sufra una lesión deberá reportarla a su jefe inmediato.

20. Todo accidente ocurrido en los laboratorios deberá ser atendido para su control, por la primera persona capacitada y enterada de la situación.

21. Al término de la práctica, el maestro será responsable de supervisar que los alumnos ordenen y limpien su lugar de trabajo. Asegurando que el laboratorio sea entregado al encargado en condiciones óptimas.

22. La persona que se presente bajo el influjo de alcohol o drogas, que incurra en actos de violencia, daño a la propiedad intencional o negligencia o tome objetos o valores sin autorización, será reportado de manera inmediata ante la H. Comisión de Responsabilidades y Sanciones del CU Costa Sur.

SEGURIDAD EN LA EJECUCIÓN DE LAS PRÁCTICAS.

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

Para el desarrollo de las prácticas se pueden presentar los siguientes peligros y su riesgo asociado y es importante que el estudiante los considere y tome las medidas de prevención pertinentes:

No.	Peligro o fuente de energía	Riesgo asociado
1	Manejo de corriente alterna.	Electrochoque, daño a los equipos.
2	Manejo de corriente continua.	Electrochoque, daño a los equipos.

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

Contenido

Contenido	7
PRÁCTICA 1 – Arreglos Unidimensionales	8
Objetivo General	9
Objetivo Específico	9
Materiales y Equipo	10
Metodología o Desarrollo	10
Resultado y Conclusiones	10
Referencias	11
PRÁCTICA 2 – Arreglo unidimensional, asignación tamaño	11
Objetivo General	12
Objetivo Específico	12
Materiales y Equipo	13
- 13	
- 13	
Metodología o Desarrollo	13
Resultado y Conclusiones	14
Referencias	15
PRÁCTICA 3 – Arreglos como contadores	15
Objetivo General	16
Objetivo Específico	16
Materiales y Equipo	17
Metodología o Desarrollo	17
Resultado y Conclusiones	17
Referencias	17
PRÁCTICA 4 – Arreglos bidimensionales	17
Objetivo General	18
Objetivo Específico	18
Materiales y Equipo	19
Metodología o Desarrollo	19
Resultado y Conclusiones	19

	UNIVERSIDAD DE GUADALAJARA	Academia de Redes e Informática
	CENTRO UNIVERSITARIO DE LA COSTA SUR	
	DEPARTAMENTO DE INGENIERÍAS	Fecha de Revisión
	Manual de Prácticas	Julio 2025
	Estructura de Datos	

PRÁCTICA 5 - Arreglos bidimensionales “Reservaciones”	19
Objetivo General	20
Objetivo Específico	21
Materiales y Equipo	21
Metodología o Desarrollo	21
Resultado y Conclusiones	23
Referencias	24
PRÁCTICA 6 – Arreglos bidimensionales “Cuadro Magico”	25
Objetivo General	26
Objetivo Específico	26
Materiales y Equipo	26
Metodología o Desarrollo	26
Resultado y Conclusiones	29
Referencias	29

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

PRÁCTICA 1 – Arreglos Unidimensionales

Carrera:	INGENIERÍA EN MECATRÓNICA
Nombre de la materia:	Estructura de Datos

Código	Nombre completo de los alumnos

Fecha:	
--------	--

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

Objetivo General

El alumno comprenderá la importancia de las estructuras de datos utilizando los objetos arreglo. En los cuales podrá agregar información relacionada de un mismo tipo.

Objetivo Específico

- Identificara las distintas maneras de declarar un arreglo
- Identificará la forma de acceder a los índices a través de una estructura de control

Materiales y Equipo

- Equipo de Computo
- IDE para desarrollo

Metodología o Desarrollo

Las estructuras de datos son colecciones de elementos de datos relacionados. Los objetos arreglo son estructuras de datos que consisten en elementos de datos relacionados, del mismo tipo. Los arreglos facilitan el procesamiento de grupos de valores relacionados. Los arreglos conservan la misma longitud una vez creados.

Para la siguiente práctica, se creará el siguiente programa declarando arreglos de distinto tipos de datos, así como 3 formas distintas de declarar un arreglo y acceder a la información almacenada dentro de este.

```

10  * @author Gilbert
11  */
12  public class ArreglosIMEC {
13
14  /**
15  * @param args the command line arguments
16  */
17  public static void main(String[] args) {
18      // TODO code application logic here
19      String[] datos = new String[5]; //Se declarará un arreglo de 5 posiciones de memoria
20      String[] estados = {"Jalisco", "México", "Oaxaca", "Estado de México", "Colima"}; //Se inicializa un arreglo con datos almacenados
21      int[] dias = new int[6];// Se declara un arreglo entero de 6 posiciones
22      // 
23      dias[0] = 10;
24      dias[1] = 4; // Llenamos el arreglo accediendo a una posición de memoria a la vez
25      dias[2] = 12;
26      dias[3] = 1;
27      dias[4] = 65;
28      dias[5] = 10;
29
30      for (int indice = 0; indice < 5; indice++) { // A traves de un ciclo for podemos imprimir cada posición de memoria
31          System.out.println(estados[indice]); //Impresión del arreglo estados
32      }
33
34      System.out.printf("%s%8s%n", "Indice", "Valor"); //Utilizamos el Format para darle formato a la impresión
35      for (int indice = 1; indice < dias.length; indice++) {
36          System.out.printf("%5d%8d%n", indice, dias[indice]);
37      }
38  }
39
40 }
41

```

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

Resultado y Conclusiones

Al finalizar la práctica, el alumno podrá relacionar lo visto teóricamente con respecto a las estructuras de datos y la aplicación de arreglos.

Se realizará la modificación de los datos ingresados. Así como la declaración de un nuevo arreglo de 20 posiciones al cual a través de un ciclo for se debe llenar del 1 al 20, en imprimirllo.

Referencias

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

PRÁCTICA 2 - Arreglo unidimensional, asignación tamaño

Carrera:	INGENIERÍA EN MECATRÓNICA
Nombre de la materia:	Estructura de Datos

Código	Nombre completo de los alumnos

Fecha:	
--------	--

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

Objetivo General

- Relacionar los conocimientos teóricos de la clase con la práctica
- El alumno conocerá la declaración de arreglos a través de la interfaz del usuario.

Objetivo Específico

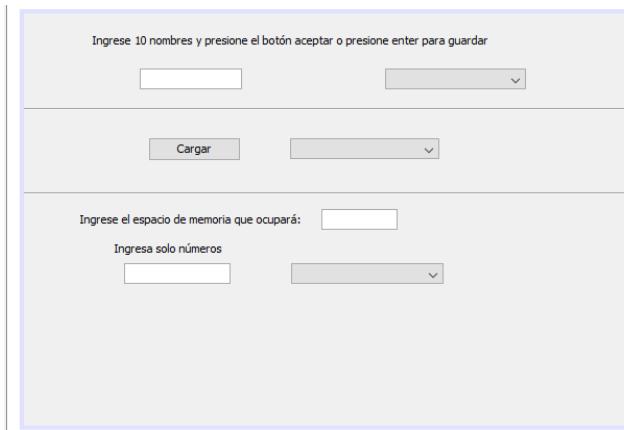
- Identificara las formas de llenar información dentro de un arreglo.
- Podrá indicarle al usuario que controle el tamaño del arreglo para los datos que asignará dentro del arreglo.
- Identificará el uso de algunas estructuras de control y eventos.
-

Materiales y Equipo

- Equipo de computo
- IDE de desarrollo

Metodología o Desarrollo

Crea la siguiente interfaz gráfica con un JFRAME llamado interfaz



A los objetos que tenemos en el JFRAME indicará las siguientes nombres de variables:

1. Al primer JTextField le indicará el nombre de variable txtNombres
2. Al primer combo box le indicará el nombre de cboNombres

En la segunda sección

3. Al JButton le asignará el nombre de btnCargar
4. Al JComboBox le asignará el nombre de cboEstados

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

En la tercera sección

5. Al JTextField le asignará el nombre de txtTamArreglo
6. Al JTextField le asignará el nombre de txtNumeros
7. Al JComboBox le asignará el nombre de cboNumeros

Al inicio de la clase agregará el siguiente código, así como en el constructor de la clase

```

9  * @author Gilbert
10 */
11 public class interfaz extends javax.swing.JFrame {
12
13     String nombres[] = new String[10];//Declaramos un arreglo de 10 posiciones
14     String estados[] = {"Baja California", "Aguascalientes", "Jalisco", "Oaxaca", "Durango");//Inicializamos
15     int contador = 0;//Variable de tipo entero
16     int contador2 = 0;
17     int numerosDecimales[];//Declaramos arreglo de tipo entero, pero sin indicar su tamaño
18
19     public interfaz() {
20         initComponents();
21         setLocationRelativeTo(this);//Indicamos que nuestra ventana aparezca centrada
22     }
23

```

Creará los siguientes eventos de tipo actionPerformed en los siguientes objetos:

- txtNombre

```

144     private void txtNombresActionPerformed(java.awt.event.ActionEvent evt) {
145         if (contador <= 9) {
146             nombres[contador] = txtNombres.getText();
147             cboNombres.addItem(nombres[contador]);
148             txtNombres.setText("");
149             contador++;
150         }
151
152

```

- btnCargar

```

155     private void btnCargarActionPerformed(java.awt.event.ActionEvent evt) {
156         for (int indice = 0; indice < estados.length; indice++) {
157             cboEstados.addItem(estados[indice]);
158         }
159

```

- txtTamArreglo

```

162     private void txtTamArregloActionPerformed(java.awt.event.ActionEvent evt) {
163         numerosDecimales = new int[Integer.parseInt(txtTamArreglo.getText())];
164         txtTamArreglo.setEnabled(false);
165         System.out.println(numerosDecimales.length);
166

```

- TxtNumeros

```

168     private void txtNumerosActionPerformed(java.awt.event.ActionEvent evt) {
169         if (contador2 < numerosDecimales.length) {
170             numerosDecimales[contador2] = Integer.parseInt(txtNumeros.getText());
171             cboNumeros.addItem(String.valueOf(numerosDecimales[contador2]));
172             contador2++;
173         }

```

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

Resultado y Conclusiones

El alumno relacionará lo visto en clase con los resultados obtenido de la practica que acaba de realizar.

Así mismo, realizará las siguientes modificaciones,

1. Asignar nuevos arreglos de distintos tipos de datos
2. Prevenir que el usuario solo ingrese valores enteros al asignar el tamaño

Referencias

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

PRÁCTICA 3 - Arreglos como contadores

Carrera:	INGENIERÍA EN MECATRÓNICA
Nombre de la materia:	Estructura de Datos

Código	Nombre completo de los alumnos

Fecha:	
--------	--

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

Objetivo General

El alumno podrá hacer uso de los arreglos unidimensionales para el conteo de registro de datos, y no solo como una forma de almacenar valores.

Objetivo Específico

- Identificara el uso de la librería SecureRandom
- Identificará el uso de un arreglo unidimensional como contador.

Materiales y Equipo

- Equipo de computo
- IDE de desarrollo

Metodología o Desarrollo

Para la practica creará el siguiente programa para ejecutarse en consola

```

3
4  import java.security.SecureRandom;
5
6  /**
7   * @author Gilbert
8   */
9  public class ArreglosIMEC3 {
10
11
12  public static void main(String[] args) {
13      SecureRandom aRandom = new SecureRandom(); // Creamos una variable para generar números aleatorios
14      int[] repeticiones = new int[10]; //Arreglo de tipo entero de 10 posiciones
15      int numAleatorio; //Variable de tipo entero sin inicializar
16      for (int i = 0; i < 100; i++) { //Ciclo for que dará 100 vueltas
17          numAleatorio = aRandom.nextInt(10); //Indicamos que se generen números aleatorios del 0 al 9
18          ++repeticiones[numAleatorio]; //El número aleatorio generado lo asignamos de posición de memoria
19          //los signo ++ nos permiten sumar en 1 dependiendo la posición de memoria
20      }
21      System.out.printf("%s%10s%n", "Indice", "Repeticiones"); //Damos formato para imprimir el resultado
22      for (int indice = 0; indice < repeticiones.length; indice++) { //Ciclo for para imprimir los valores
23          System.out.printf("%d%10d%n", indice, repeticiones[indice]);
24      }
25  }
26
27 }

```

Resultado y Conclusiones

Una vez terminado, el alumno pondrá en uso lo relacionado con la practica. Para aplicar el código realizado en otros programas, que pueden servir para encuestas, etc.

Referencias

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

PRÁCTICA 4 - Arreglos bidimensionales

Carrera:	INGENIERÍA EN MECATRÓNICA
Nombre de la materia:	Estructura de Datos

Código	Nombre completo de los alumnos

Fecha:	
--------	--

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

Objetivo General

El alumno conocerá las formas en que puede declarar un arreglo bidimensional, y como acceder a cada una de las posiciones de memoria a través del un iterador anidado en este caso en particular, la estructura de control for.

Objetivo Específico

- Declarar arreglos bidimensionales
- Iterar a través de los arreglos mediante el ciclo for anidado
- Inicializar, y hacer referencia a los elementos.

Materiales y Equipo

- Equipo de computo
- IDE de desarrollo

Metodología o Desarrollo

Los arreglos multidimensionales de dos dimensiones se utilizan con frecuencia para representar tablas de valores, con datos ordenados en filas y columnas. Para identificar un elemento específico de una tabla, debemos especificar dos índices.

Por convención, el primero identifica la fila y el segundo la columna. Los arreglos que requieren dos índices para identificar un elemento específico se llama arreglos bidimensionales.

Para poner en práctica lo anterior, crearemos la siguiente interfaz grafica

```

10  * @author Gilbert
11  */
12  public class ArreglosIMEC6 {
13
14  public static void main(String[] args) {
15      int arreglo[][] = new int[10][10];//Declaramos un arreglo de 10 x 10 de tipo entero
16
17      for (int fila = 0; fila < arreglo.length; fila++) {//Creamos un ciclo for anidado para acceder a cada fila y columna del arreglo
18          //E indicamos el método length para indicar el límite de este ciclo por la fila
19          for (int columna = 0; columna < arreglo[fila].length; columna++) {//ciclo for para acceder a cada columna del arreglo bidimensional
20              //utilizamos el método length después de los corchetes de la fila para indicar el tamaño de las columnas
21              arreglo[fila][columna] = (fila + 1) * (columna + 1); //guardamos los valores resultado de la operación por fila y columna
22              System.out.printf("%10d", arreglo[fila][columna]); //imprimimos los valores con formato
23
24          }
25          System.out.println();
26      }
27  }
28
29
30 }

```

Resultado y Conclusiones

Una vez terminada la práctica, el alumno relacionará lo visto en clase presencial teórica, con lo implementado dentro de la práctica. Podrá sacar conclusiones entre el uso de un arreglo de una sola dimensión y de dos dimensiones. Modificando los valores de entrada.

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

PRÁCTICA 5 - Arreglos bidimensionales “Reservaciones”

Carrera:	INGENIERÍA EN MECATRÓNICA
Nombre de la materia:	Estructura de Datos

Código	Nombre completo de los alumnos

Fecha:	
--------	--

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

Objetivo General

El alumno pondrá en práctica sus conocimientos, implementando el arreglo bidimensional para almacenar datos en un registro de reservaciones de camiones. En la forma de como guardar y accede a esta misma información, así como aplicar conocimiento aprendidos anteriormente, como métodos, arreglos unidimensionales, paso de referencia.

Objetivo Específico

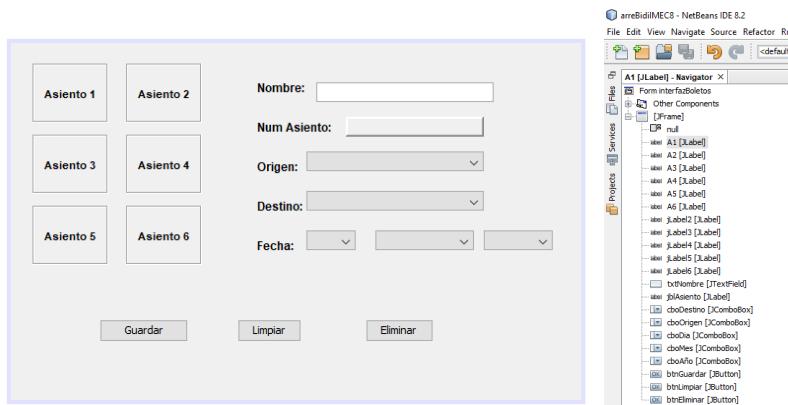
- Utilizar arreglos bidimensionales para almacenar datos
- Obtendrá listas y tabla de valores
- Aplicar iterador para el desplazamiento dentro del arreglo

Materiales y Equipo

- Equipo de computo.
- IDE de desarrollo.

Metodología o Desarrollo

Para poner en práctica el almacenamiento de información a través de los arreglos bidimensionales, creamos la siguiente interfaz, declarando las siguientes variables a los objetos.



En el código empezaremos con lo siguiente:

```

2  import java.awt.Color;
3  import javax.swing.JComboBox;
4  import javax.swing.JLabel;
5
6  /**
7   * @author Gilbert
8   */
9
10 public class interfazBoletos extends javax.swing.JFrame {
11
12   String boletos[][] = new String[6][7];//Declaramos un arreglo bidimensional de 6 x 7
13   JLabel etiquetas[] = new JLabel[6];//Creamos un arreglo unidimensional de tipo JLabel de 6 posiciones
14   String origenDestino[] = {"Seleccione", "Guadalajara", "Aatlán", "Casimiro Castillo", "La Huerta", "El Grullo", "Melaque", "Cd. Guzman"};
15   String mes[] = {"Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"};
16   JComboBox comboBox[] = new JComboBox[5];//Creamos un arreglo de tipo JComboBox de 5 posiciones
17   Color colorEtiquetas[] = {Color.LIGHT_GRAY, Color.DARK_GRAY, Color.RED};//Creamos un arreglo de tipo COLOR e inicializamos los siguientes valores
18   int asientoSeleccionado = 0;

```

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

Creamos el método cargar, para guardar las variables creados de tipo JComboBox y JLabel a un arreglo.

```

29  public void cargar() { // Creamos el siguiente método para inicializar los arreglos de tipo JComboBox y JLabel
30    combo[0] = cboOrigen;
31    combo[1] = cboDestino;
32    combo[2] = cboDia;
33    combo[3] = cboMes;
34    combo[4] = cboAño;
35
36    etiquetas[0] = A1;
37    etiquetas[1] = A2;
38    etiquetas[2] = A3;
39    etiquetas[3] = A4;
40    etiquetas[4] = A5;
41    etiquetas[5] = A6;
42
43    for (int i = 0; i < origenDestino.length; i++) { //Cargar orígenes y destinos a los combo box
44      cboOrigen.addItem(origenDestino[i]);
45      cboDestino.addItem(origenDestino[i]);
46    }
47
48    for (int i = 0; i < mes.length; i++) {
49      cboMes.addItem(mes[i]);
50    }
51
52    for (int i = 0; i < 31; i++) {
53      cboDia.addItem(" " + (i + 1));
54    }
55    cboAño.addItem("2020");
56    cboAño.addItem("2021");
57
58    for (int i = 0; i < etiquetas.length; i++) {
59      etiquetas[i].setBackground(colorEtiquetas[0]);
60      etiquetas[i].setOpaque(true);
61    }
62
63  }

```

Creamos los siguientes métodos:

```

64  public void onOff(boolean actDes) { //Creamos un método con un parámetro de tipo booleano para activar y desactivar el formulario
65    txtNombre.setEnabled(actDes);
66    for (int i = 0; i < combo.length; i++) {
67      combo[i].setEnabled(actDes);
68    }
69    btnGuardar.setEnabled(actDes);
70    btnEliminar.setEnabled(actDes);
71    btnLimpiar.setEnabled(actDes);
72  }
73
74
75  public void limpiar() { //Creamos un método que nos permitirá limpiar los JTextField y posicionar en 0 los combo box
76    txtNombre.setText("");
77    jblAsiento.setText("");
78    for (int i = 0; i < combo.length; i++) {
79      combo[i].setSelectedIndex(0);
80    }
81  }
82
83
84  public void mostrar(int seleccion) { //Creamos un método que nos mostrará los datos guardados según la mesa seleccionada
85    txtNombre.setText(boletos[seleccion][0]);
86    jblAsiento.setText(boletos[seleccion][1]);
87
88    cboOrigen.setSelectedItem(boletos[seleccion][2]);
89    cboDestino.setSelectedItem(boletos[seleccion][3]);
90    cboDia.setSelectedItem(boletos[seleccion][4]);
91    cboMes.setSelectedItem(boletos[seleccion][5]);
92    cboAño.setSelectedItem(boletos[seleccion][6]);
93  }
94

```

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS Manual de Prácticas Estructura de Datos	Academia de Redes e Informática Fecha de Revisión Julio 2025
---	--	---

A cada etiqueta que representa la mesa, le crearemos un evento de tipo MouseClicked y agregaremos el siguiente código, tomando en cuenta que por cada mesa, en la variable asientoSeleccionado se indica el valor de 0, al siguiente el valor de 1 y así consecutivamente.

```

274  private void A1MouseClicked(java.awt.event.MouseEvent evt) {
275      asientoSeleccionado = 0;
276      onOff(true);
277
278      if (boletos[asientoSeleccionado][0] == null) {
279          etiquetas[asientoSeleccionado].setBackground(colorEtiquetas[1]);
280          jblAsiento.setText(etiquetas[asientoSeleccionado].getText());
281      } else {
282          mostrar(asientoSeleccionado);
283      }
284  }
285
286
287  private void A2MouseClicked(java.awt.event.MouseEvent evt) {
288      asientoSeleccionado = 1;
289      onOff(true);
290
291      if (boletos[asientoSeleccionado][0] == null) {
292          etiquetas[asientoSeleccionado].setBackground(colorEtiquetas[1]);
293          jblAsiento.setText(etiquetas[asientoSeleccionado].getText());
294      } else {
295          mostrar(asientoSeleccionado);
296      }
297  }
298
299  private void A3MouseClicked(java.awt.event.MouseEvent evt) {
300      asientoSeleccionado = 2;
301      onOff(true);
302
303      if (boletos[asientoSeleccionado][0] == null) {
304          etiquetas[asientoSeleccionado].setBackground(colorEtiquetas[1]);
305          jblAsiento.setText(etiquetas[asientoSeleccionado].getText());
306      } else {
307          mostrar(asientoSeleccionado);
308      }
309  }

```

Al botón btnGuardar le creamos un evento de tipo ActionPerformed y agregamos el siguiente código.

```

346
347  private void btnGuardarActionPerformed(java.awt.event.ActionEvent evt) {
348      boletos[asientoSeleccionado][0] = txtNombre.getText();
349      boletos[asientoSeleccionado][1] = jblAsiento.getText();
350      boletos[asientoSeleccionado][2] = (String) cboOrigen.getSelectedItem();
351      boletos[asientoSeleccionado][3] = (String) cboDestino.getSelectedItem();
352      boletos[asientoSeleccionado][4] = (String) cboDia.getSelectedItem();
353      boletos[asientoSeleccionado][5] = (String) cboMes.getSelectedItem();
354      boletos[asientoSeleccionado][6] = (String) cboAño.getSelectedItem();
355
356      etiquetas[asientoSeleccionado].setBackground(colorEtiquetas[2]);
357      onOff(false);
358      limpiar();
359  }

```

Inicializamos los métodos siguientes en el constructor de la clase.

```

public interfazBoletos() {
    initComponents();
    setSize(700, 400);
    setLocationRelativeTo(this);
    cargar();
    onOff(false);
}

```

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

Resultado y Conclusiones

El alumno en esta práctica hará las siguientes modificaciones.

1. Debe verificar que el formulario este completamente lleno para poder guardar. (Evitar guardar valores vacíos)
2. Al ir seleccionando cualquier asiento este debe cambiar de color para indicar que el asiento ha sido seleccionado, si selecciono otro, el anteriormente seleccionado debe volver a su color normal.
3. Configurar los botones de limpiar y eliminar.
4. Al momento de eliminar un registro debe mostrar un JOptionPane en el cual pregunte si estás seguro que quieres eliminar, si es así el asiento debe volver a su color cuando no ha estado seleccionado y el formulario debe bloquearse de nueva cuenta.
5. Comentar el código.

Referencias

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión Julio 2025

PRÁCTICA 6 - Arreglos bidimensionales “Cuadro Magico”

Carrera:	INGENIERÍA EN MECATRÓNICA
Nombre de la materia:	Estructura de Datos

Código	Nombre completo de los alumnos

Fecha:	
--------	--

	<p style="text-align: center;">UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS</p> <p style="text-align: center;">Manual de Prácticas Estructura de Datos</p>	<p>Academia de Redes e Informática</p> <p>Fecha de Revisión</p> <p>Julio 2025</p>
---	---	--

Objetivo General

El alumno comprenderá de una mejor manera la manipulación de los arreglos bidimensionales, para realizar juegos, crear métodos, y combinar el uso de arreglos unidimensionales con arreglos bidimensionales

Objetivo Específico

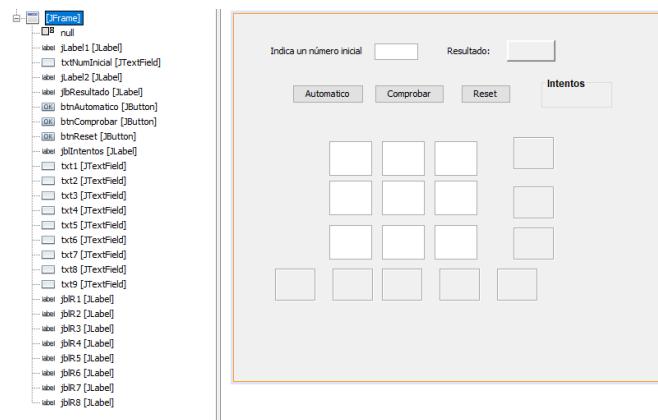
- Identificará como hacer manipulaciones comunes de arreglos bidimensionales y unidimensionales
- Identificará la forma de realizar sumas de elementos almacenados en los arreglos bidimensionales

Materiales y Equipo

- Equipo de computo
- IDE de desarrollo

Metodología o Desarrollo

Continuando con los ejercicios en donde aplicamos el uso de arreglos bidimensionales, y arreglos unidimensionales de tipo objetos, creamos la siguiente interfaz y declaramos los siguientes nombres de variables a los objetos.



	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

Importamos y declaramos lo siguiente:

```

2  import javax.swing.JLabel;
3  import javax.swing.JTextField;
4
5  /**
6   * @author Gilbert
7   */
8
9  public class cuadroMagico extends javax.swing.JFrame {
10
11     int serieNum[] = new int[9];// Declaramos un arreglo de tipo entero
12     int cuadroMagico[][] = new int[3][3]; //Declaramos un arreglo bidimensional
13     JLabel etiquetas[] = new JLabel[8];//Arreglo de tipo JLabel
14     JTextField cajasTexto[] = new JTextField[9];//Arreglo de tipo JTextField

```

Creamos el método cargar, para inicializar los arreglos

```

23
24  public void cargar() { //Cargamos los arreglos con los elementos de tipo JLabel y JTextField
25      etiquetas[0] = jblR1;
26      etiquetas[1] = jblR2;
27      etiquetas[2] = jblR3;
28      etiquetas[3] = jblR4;
29      etiquetas[4] = jblR5;
30      etiquetas[5] = jblR6;
31      etiquetas[6] = jblR7;
32      etiquetas[7] = jblR8;
33
34      cajasTexto[0] = txt1;
35      cajasTexto[1] = txt2;
36      cajasTexto[2] = txt3;
37      cajasTexto[3] = txt4;
38      cajasTexto[4] = txt5;
39      cajasTexto[5] = txt6;
40      cajasTexto[6] = txt7;
41      cajasTexto[7] = txt8;
42      cajasTexto[8] = txt9;
43
44  }

```

Creamos los siguientes métodos que estaremos utilizando:

```

46  public void onOff(boolean actDes) {// Método para habilitar o deshabilitar
47      for (int indice = 0; indice < cajasTexto.length; indice++) {
48          cajasTexto[indice].setEnabled(actDes);
49      }
50  }
51
52  public int calcular(int numeroInicial) { //Método de tipo entero calcular los número dependiendo el valor inicial
53      int resultado = 0;
54      serieNum[0] = numeroInicial;
55      for (int i = 1; i < serieNum.length; i++) { //agregar los 9 números al arreglo serieNum
56          serieNum[i] = serieNum[i - 1] + 1;
57          System.out.println(" " + serieNum[i]);
58      }
59
60      for (int i = 0; i < serieNum.length; i++) { //suma los números del arreglo serieNum
61          resultado = resultado + serieNum[i];
62      }
63      resultado = resultado / 3;
64      cargarArrBidimensional();
65
66      return resultado;
67  }

```

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

Creamos el método cargarArrBidimensional para asignar la posición de los números según la matriz

```

68
69  public void cargarArrBidimensional() {//Método para llenar los textField con el resultado correcto
70    cuadroMagico[0][0] = serieNum[1];
71    cuadroMagico[0][1] = serieNum[8];
72    cuadroMagico[0][2] = serieNum[3];
73    cuadroMagico[1][0] = serieNum[6];
74    cuadroMagico[1][1] = serieNum[4];
75    cuadroMagico[1][2] = serieNum[2];
76    cuadroMagico[2][0] = serieNum[5];
77    cuadroMagico[2][1] = serieNum[0];
78    cuadroMagico[2][2] = serieNum[7];
79
80  }
  
```

Creamos el método mostrarResultado, el cual nos servirá para llenar de forma automática nuestro JTextField y mostrar el resultado del cuadro mágico

```

82
83  public void mostrarResultado() //Método para cargar el resultado a los textField
84    int posicion = 0;
85    int temp;
86    for (int fila = 0; fila < cuadroMagico.length; fila++) {
87      temp = 0;
88      for (int columna = 0; columna < cuadroMagico[0].length; columna++) {
89        cajastexto[posicion].setText(" " + cuadroMagico[fila][columna]);
90        temp = temp + cuadroMagico[fila][columna];
91        etiquetas[fila].setText(" " + temp);
92        posicion++;
93        System.out.println("posicion " + posicion);
94      }
95      for (int columna = 0; columna < cuadroMagico[0].length; columna++) {
96        temp = 0;
97        for (int fila = 0; fila < cuadroMagico.length; fila++) {
98          temp = temp + cuadroMagico[fila][columna];
99          etiquetas[columna + 3].setText(" " + temp);
100        }
101      }
102      temp = 0;
103      for (int i = 0; i < cuadroMagico.length; i++) {
104        temp = temp + cuadroMagico[i][i];
105      }
106      etiquetas[6].setText(" " + temp);
107      for (int i = 2; i > cuadroMagico.length; i--) {
108        temp = temp + cuadroMagico[i][i];
109      }
110      etiquetas[7].setText(" " + temp);
111    }
  
```

Creamos un evento de tipo ActionPerfomred en el JTextField txtNumInicial, el cual nos permitirá indicar el número con el cual queremos iniciar nuestro cuadro mágico.

```

289
290  private void txtNumInicialActionPerformed(java.awt.event.ActionEvent evt) {
291    int res = calcular(Integer.parseInt(txtNumInicial.getText()));
292    jlbResultado.setText(" " + res);
293    onOff(true);
294    txtNumInicial.setEnabled(false);
295
296  private void btnAutomaticoActionPerformed(java.awt.event.ActionEvent evt) {
297    mostrarResultado();
298  }
  
```

	UNIVERSIDAD DE GUADALAJARA CENTRO UNIVERSITARIO DE LA COSTA SUR DEPARTAMENTO DE INGENIERÍAS	Academia de Redes e Informática
	Manual de Prácticas Estructura de Datos	Fecha de Revisión
		Julio 2025

Inicializamos los métodos en el constructor de la clase.

```

15
16  public cuadroMagico() {
17    initComponents();
18    setSize(500, 500);
19    setLocationRelativeTo(this);
20    cargar();
21    onOff(false);
22  }

```

Resultado y Conclusiones

El alumno deberá realizar las siguientes modificaciones

1. Corregir la suma de la diagonal de izquierda a derecha, mencionada en el video.
2. Cuando el programa inicie los botones deben estar bloqueados y se deben activar cuando el usuario indique el número con el cual quiere iniciar.
3. El usuario debe llenar los JTextField del cuadro mágico y dar clic en el botón comprobar los números se deben guardar en un arreglo bidimensional en donde se realizará la suman (similar a lo que se hizo en el botón automático) para verificar si su resultado es correcto, en los JLabel se debe mostrar el total sumado-
4. Si el resultado es incorrecto, en el JLabel de intentos se debe incrementar en 1, hasta un límite de 3 intentos.
5. Si el resultado es correcto, debe aparecer un mensaje que diga que lo ha logrado.
6. Al terminar los 3 intentos ya no podrá seguir jugando, y se activa el botón automatico para que el usuario pueda ver la respuesta correcta.
7. El botón reset limpia todos los TextField, bloquea todo y desbloquea el JTextField para indicar un nuevo número y comenzar a jugar de nuevo.

Referencias