

CIFP César Manrique.

Entornos de desarrollo 1º de Desarrollo de Aplicaciones Web

Profesora: Sofía del Carmen Hernández González

Documentación con Javadocs.



Esta obra está licenciada bajo la Licencia Creative Commons Atribución 4.0 Internacional.
Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0/> o
envíe una carta a Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Índice

Ejercicio.....	1
Enunciado.....	1
Desarrollo.....	1
Evidencia gráfica.....	2

Índice de imágenes

Clase Main y método para saludar.....	2
(1/4) Clase CalculadoraEjemplo con todos los métodos usados, documentados y lista para ser probada.....	3
(2/4) Clase CalculadoraEjemplo con todos los métodos usados, documentados y lista para ser probada.....	4
(3/4) Clase CalculadoraEjemplo con todos los métodos usados, documentados y lista para ser probada.....	5
(4/4) Clase CalculadoraEjemplo con todos los métodos usados, documentados y lista para ser probada.....	6
Muestra del funcionamiento de la app.....	7
Proceso de generación de la documentación y directorio con el resultado.....	8
Vista de los archivos generados.....	8
Vista de la página index.html.....	9
(1/4) Documentación generada de la clase CalculadoraEjemplo.....	9
(2/4) Documentación generada de la clase CalculadoraEjemplo.....	10
(3/4) Documentación generada de la clase CalculadoraEjemplo.....	11
(4/4) Documentación generada de la clase CalculadoraEjemplo.....	11

Ejercicio

Enunciado

Crea una clase denominada CalculadoraEjemplo que tenga los atributos operando1, operando2, resultado y métodos que implementen las operaciones básicas (Suma, Resta, Multiplicación y División). Documenta esta clase conforme a los estándares JavaDoc y comprueba cómo se visualiza la documentación.

[Volver al índice](#)

Desarrollo

Se procedió a escribir la clase solicitada, además del método de entrada de la aplicación o método Main, luego se comentó y se procedió a correr la aplicación, una vez comprobado su funcionamiento se procedió a generar la documentación con Javadocs se adjuntan los resultados.

[Volver al índice](#)

Evidencia gráfica

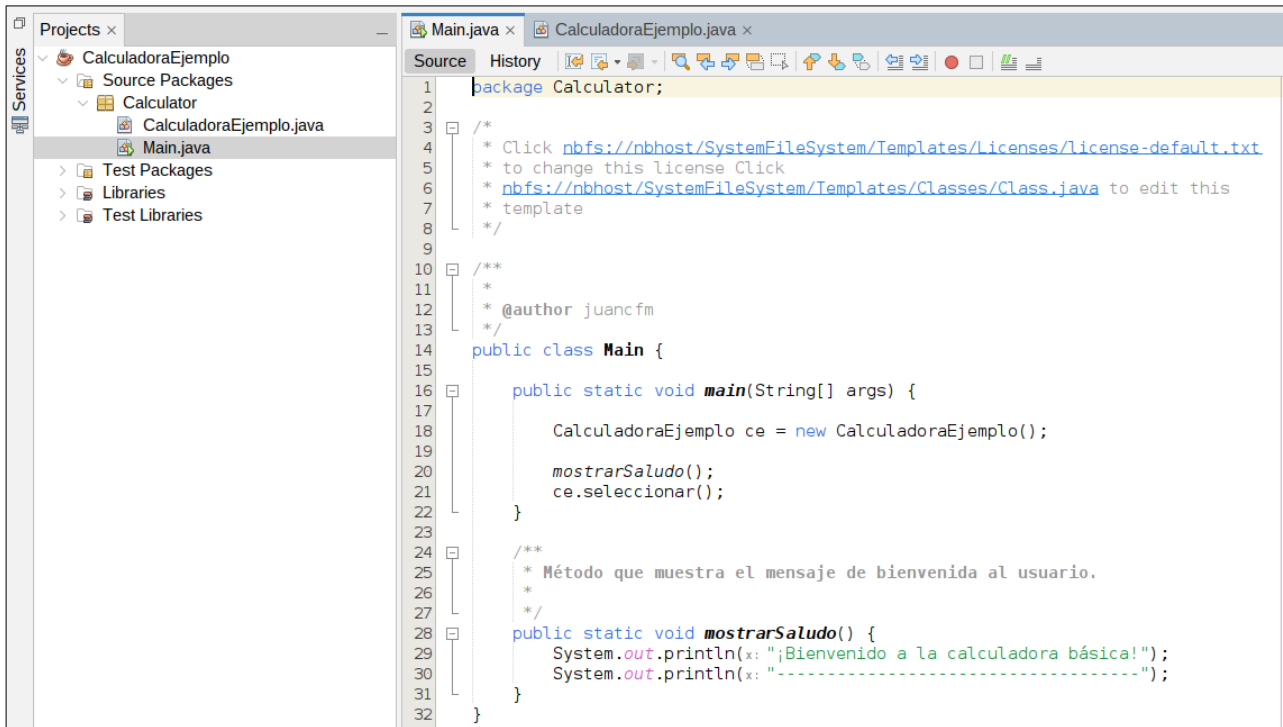


Imagen 1: Clase Main y método para saludar

[Volver al índice](#)

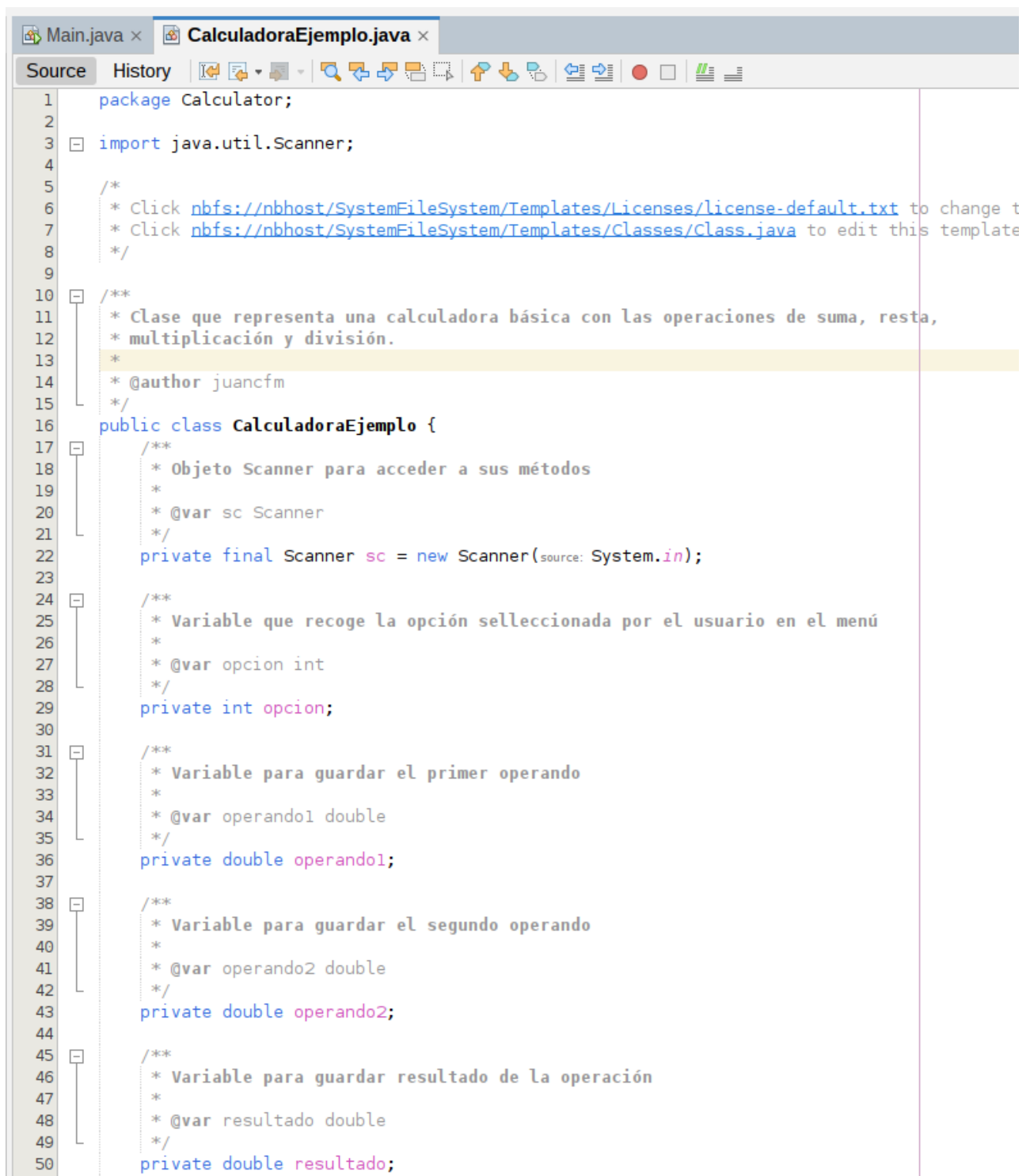


Imagen 2: (1/4) Clase CalculadoraEjemplo con todos los métodos usados, documentados y lista para ser probada

[Volver al índice](#)

```
51
52
53 /**
54  * Constructor que inicializa los operandos en cero.
55  */
56 public CalculadoraEjemplo() {
57     this.operando1 = 0;
58     this.operando2 = 0;
59     this.resultado = 0;
60 }
61
62 /**
63  * Método que pregunta al usuario qué operación desea realizar y llama al
64  * método correspondiente.
65  */
66 public void seleccionar() {
67     try (Scanner scanner = new Scanner(System.in)) {
68         while (true) {
69             System.out.println(x: "Seleccione la operación a realizar:");
70             System.out.println(x: "1. Suma");
71             System.out.println(x: "2. Resta");
72             System.out.println(x: "3. Multiplicación");
73             System.out.println(x: "4. División");
74             System.out.println(x: "5. Salir");
75
76             opcion = scanner.nextInt();
77
78             if (opcion == 5) {
79                 break;
80             }
81
82             System.out.println(x: "Introduzca el primer número:");
83             this.operando1 = pedirNumeroDecimal(scanner: sc);
84             System.out.println(x: "Introduzca el segundo número:");
85             this.operando2 = pedirNumeroDecimal(scanner: sc);
86
87             switch (opcion) {
88                 case 1 -> {
89                     this.resultado = sumar(operando1: this.operando1, operando2: this.operando2);
90                     presentarResultado(operación: "suma", numero: this.resultado);
91                 }
92                 case 2 -> {
93                     this.resultado = restar(operando1: this.operando1, operando2: this.operando2);
94                     presentarResultado(operación: "resta", numero: this.resultado);
95                 }
96                 case 3 -> {
97                     this.resultado = multiplicar(operando1: this.operando1, operando2: this.operando2);
98                     presentarResultado(operación: "multiplicación", numero: this.resultado);
99                 }
100                 case 4 -> {
101                     this.resultado = dividir(operando1: this.operando1, operando2: this.operando2);
102                     presentarResultado(operación: "división", numero: this.resultado);
103                 }
104                 default ->
105                     System.out.println(x: "Operación no válida.");
106             }
107         }
108     }
109 }
```

Imagen 3: (2/4) Clase CalculadoraEjemplo con todos los métodos usados, documentados y lista para ser probada

[Volver al índice](#)

```
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161

/**
 * Método que pide al usuario que introduzca un número decimal.
 *
 * @param scanner
 * @return el número decimal introducido por el usuario
 */
public static double pedirNumeroDecimal(Scanner scanner) {
    return scanner.nextDouble();
}

/**
 * Método para sumar los operandos.
 *
 * @param operando1 Primer operando.
 * @param operando2 Segundo operando.
 * @return Resultado de la suma de los operandos.
 */
public double sumar(double operando1, double operando2) {
    this.operando1 = operando1;
    this.operando2 = operando2;
    this.resultado = operando1 + operando2;
    return resultado;
}

/**
 * Método para restar los operandos.
 *
 * @param operando1 Primer operando.
 * @param operando2 Segundo operando.
 * @return Resultado de la resta de los operandos.
 */
public double restar(double operando1, double operando2) {
    this.operando1 = operando1;
    this.operando2 = operando2;
    this.resultado = operando1 - operando2;
    return resultado;
}

/**
 * Método para multiplicar los operandos.
 *
 * @param operando1 Primer operando.
 * @param operando2 Segundo operando.
 * @return Resultado de la multiplicación de los operandos.
 */
public double multiplicar(double operando1, double operando2) {
    this.operando1 = operando1;
    this.operando2 = operando2;
    this.resultado = operando1 * operando2;
    return resultado;
}
```

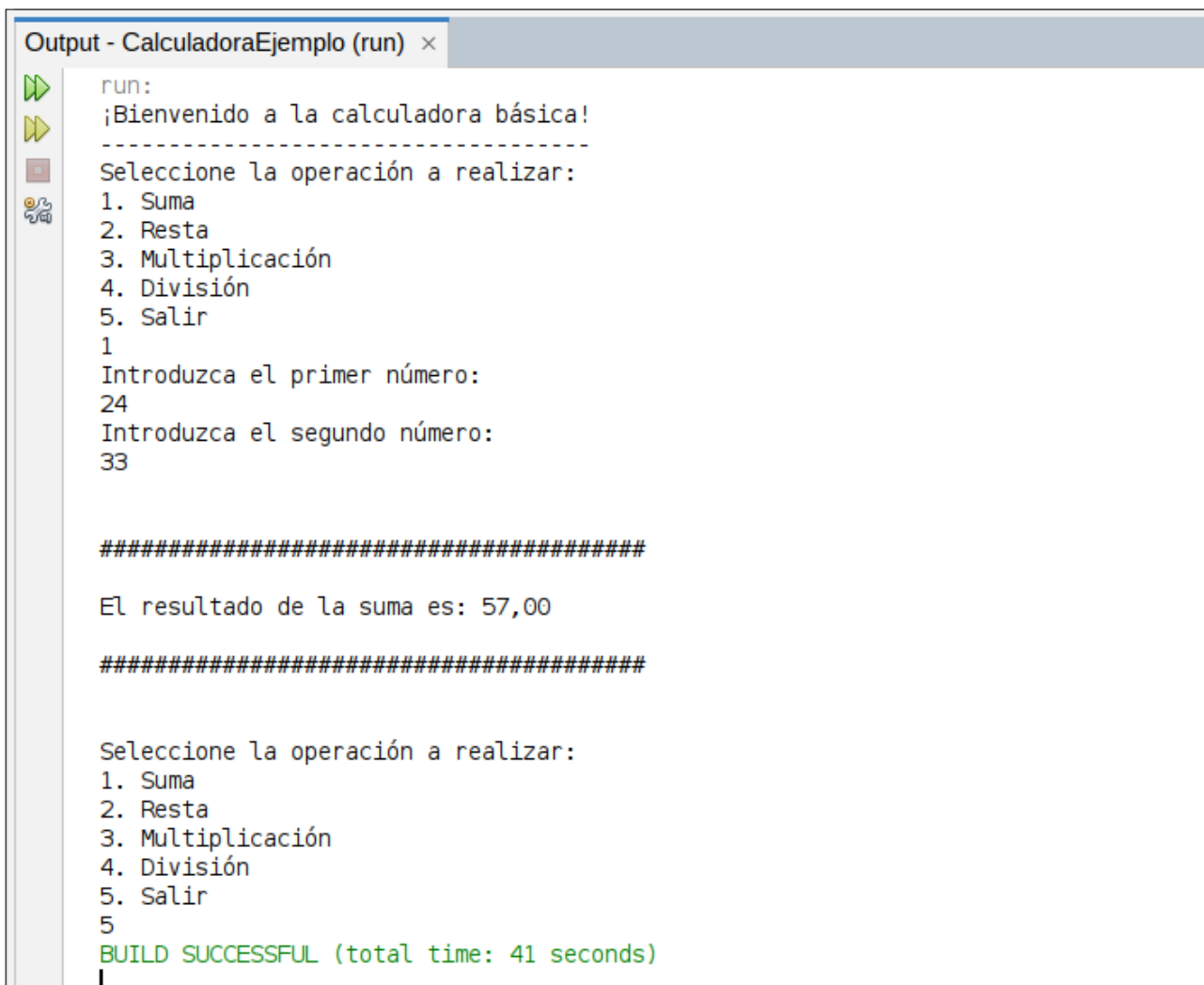
Imagen 4: (3/4) Clase CalculadoraEjemplo con todos los métodos usados, documentados y lista para ser probada

[Volver al índice](#)


```
162  /**
163   * Método para dividir los operandos.
164   *
165   * @param operando1 Primer operando.
166   * @param operando2 Segundo operando.
167   * @return Resultado de la división de los operandos.
168   * @throws ArithmeticException Si el segundo operando es cero.
169   */
170  public double dividir(double operando1, double operando2) throws ArithmeticException {
171      if (operando2 == 0) {
172          throw new ArithmeticException(s: "No se puede dividir por cero.");
173      }
174      this.operando1 = operando1;
175      this.operando2 = operando2;
176      this.resultado = operando1 / operando2;
177      return resultado;
178  }
179
180  /**
181   * Método para obtener el valor del primer operando.
182   *
183   * @return Valor del primer operando.
184   */
185  public double getOperando1() {
186      return operando1;
187  }
188
189  /**
190   * Método para obtener el valor del segundo operando.
191   *
192   * @return Valor del segundo operando.
193   */
194  public double getOperando2() {
195      return operando2;
196  }
197
198  /**
199   * Método para obtener el valor del resultado.
200   *
201   * @return Valor del resultado.
202   */
203  public double getResultado() {
204      return resultado;
205  }
206
207  /**
208   * Método que presenta el resultado de las operaciones y les da formato para
209   * presentar resultado con dos decimales.
210   *
211   * @param operacion el nombre de la operación efectuada
212   * @param numero el resultado obtenido
213   */
214  public void presentarResultado(String operacion, double numero) {
215      System.out.println(x: "\n\n#####\n");
216      System.out.print("El resultado de la " + operacion + " es: ");
217      System.out.printf(format: "%.2f", args: numero);
218      System.out.println(x: "");
219      System.out.println(x: "\n#####\n\n");
220  }
221 }
```

Imagen 5: (4/4) Clase CalculadoraEjemplo con todos los métodos usados, documentados y lista para ser probada

[Volver al índice](#)



```
Output - CalculadoraEjemplo (run) x
run:
¡Bienvenido a la calculadora básica!
-----
Seleccione la operación a realizar:
1. Suma
2. Resta
3. Multiplicación
4. División
5. Salir
1
Introduzca el primer número:
24
Introduzca el segundo número:
33

#####

El resultado de la suma es: 57,00

#####


Seleccione la operación a realizar:
1. Suma
2. Resta
3. Multiplicación
4. División
5. Salir
5
BUILD SUCCESSFUL (total time: 41 seconds)
```

Imagen 6: Muestra del funcionamiento de la app

[Volver al índice](#)

Procedemos ahora a generar nuestra documentación con Javadocs accediendo con el botón derecho sobre el nombre de nuestro proyecto y seleccionando en el menú desplegable la opción “Generate Javadocs”, o bien desde el menú run accedemos también a la opción “Generate Javadocs” y accedemos a la página generada.

Documentación con Javadocs.



```
Output - CalculadoraEjemplo (javadoc) x
public static void main(String[] args) {
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/Calculator/package-summary.html...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/Calculator/package-tree.html...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/Calculator/class-use/CalculadoraEjemplo.html...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/Calculator/class-use/Main.html...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/Calculator/package-use.html...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/overview-tree.html...
Building index for all classes...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/allclasses-index.html...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/allpackages-index.html...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/index-files/index-1.html...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/index-files/index-2.html...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/index-files/index-3.html...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/index-files/index-4.html...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/index-files/index-5.html...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/index-files/index-6.html...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/index-files/index-7.html...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/search.html...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/index.html...
Generating /home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/help-doc.html...
4 warnings
Browsing: file:/home/juancfm/NetBeansProjects/CalculadoraEjemplo/dist/javadoc/index.html
javadoc:
BUILD SUCCESSFUL (total time: 3 seconds)
```

Imagen 7: Proceso de generación de la documentación y directorio con el resultado

[Volver al índice](#)

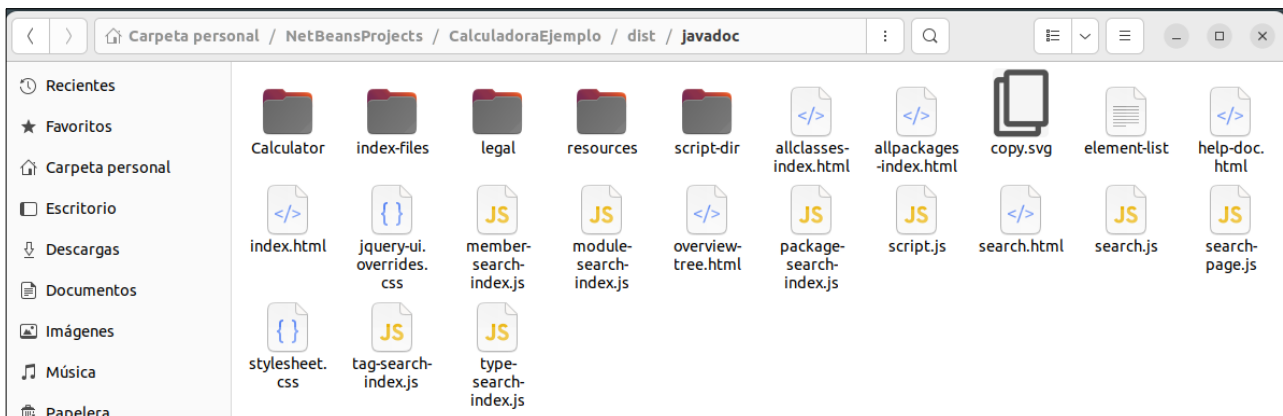


Imagen 8: Vista de los archivos generados

[Volver al índice](#)

Documentación con Javadocs.

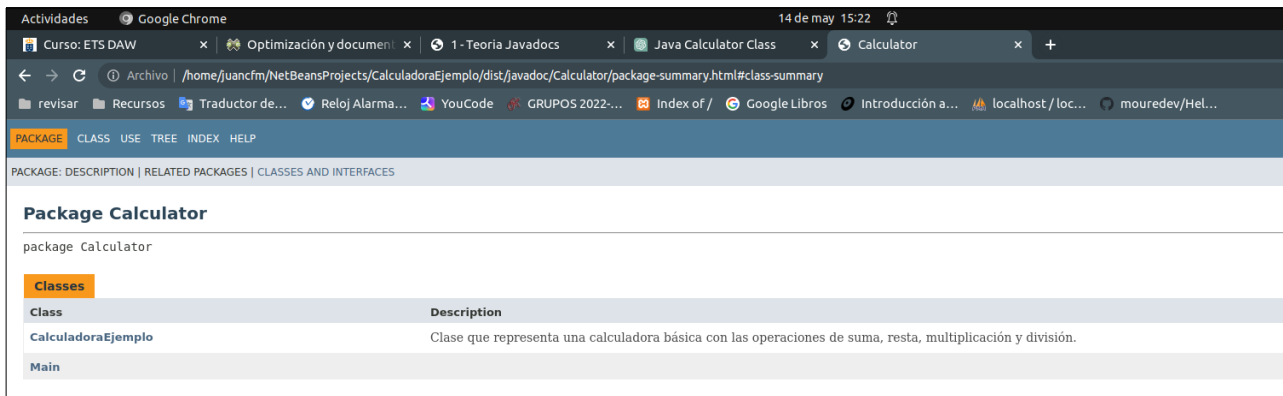


Imagen 9: Vista de la página index.html

[Volver al índice](#)

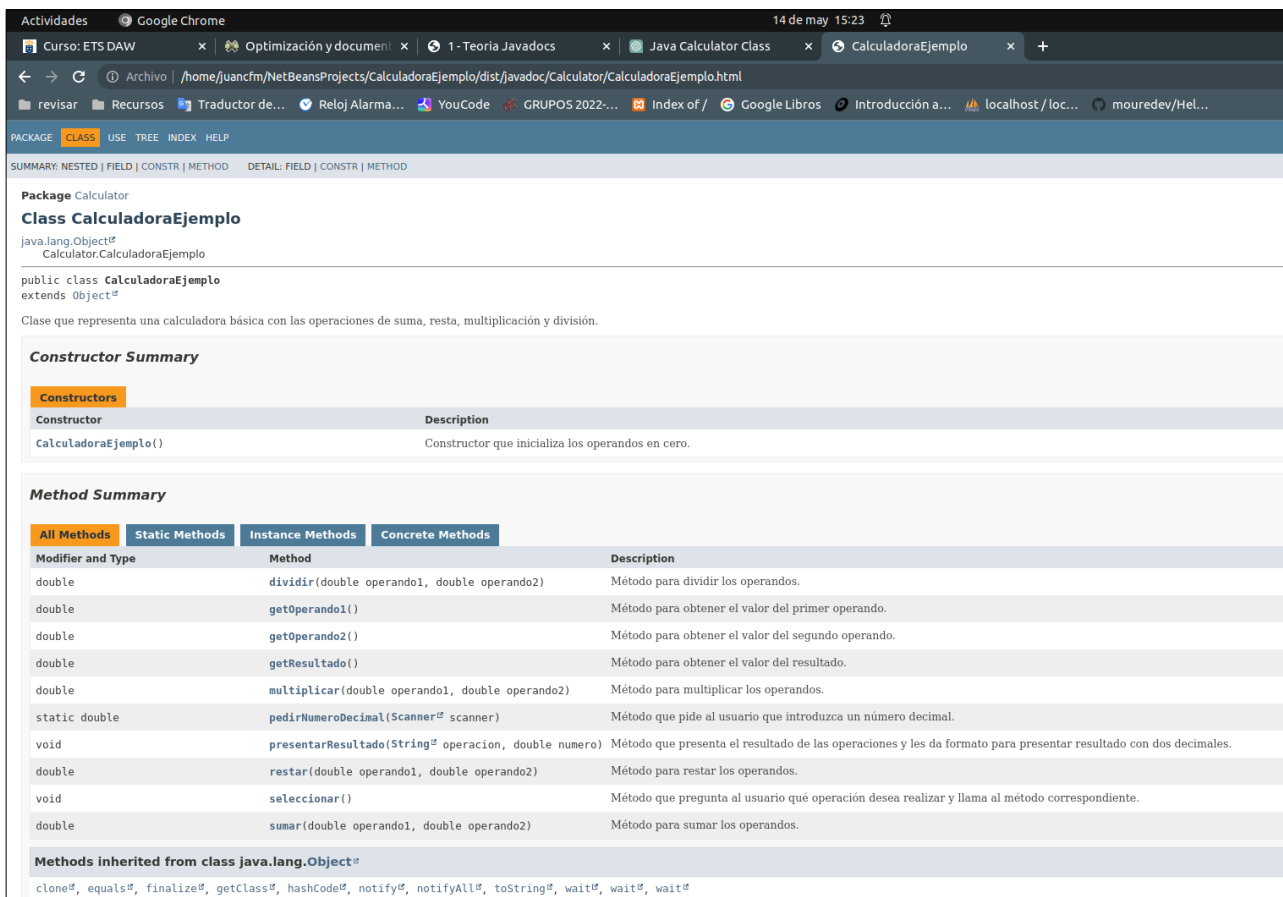


Imagen 10: (1/4) Documentación generada de la clase CalculadoraEjemplo

[Volver al índice](#)

Constructor Details
CalculadoraEjemplo
<pre>public CalculadoraEjemplo()</pre> <p>Constructor que inicializa los operandos en cero.</p>
Method Details
seleccionar
<pre>public void seleccionar()</pre> <p>Método que pregunta al usuario qué operación desea realizar y llama al método correspondiente.</p>
pedirNumeroDecimal
<pre>public static double pedirNumeroDecimal(Scanner scanner)</pre> <p>Método que pide al usuario que introduzca un número decimal.</p> <p>Parameters: scanner -</p> <p>Returns: el número decimal introducido por el usuario</p>
sumar
<pre>public double sumar(double operando1, double operando2)</pre> <p>Método para sumar los operandos.</p> <p>Parameters: operando1 - Primer operando. operando2 - Segundo operando.</p> <p>Returns: Resultado de la suma de los operandos.</p>

Imagen 11: (2/4) Documentación generada de la clase CalculadoraEjemplo

[Volver al índice](#)

Documentación con Javadocs.

restar
<pre>public double restar(double operando1, double operando2)</pre> <p>Método para restar los operandos.</p> <p>Parameters: operando1 - Primer operando. operando2 - Segundo operando.</p> <p>Returns: Resultado de la resta de los operandos.</p>
multiplicar
<pre>public double multiplicar(double operando1, double operando2)</pre> <p>Método para multiplicar los operandos.</p> <p>Parameters: operando1 - Primer operando. operando2 - Segundo operando.</p> <p>Returns: Resultado de la multiplicación de los operandos.</p>
dividir
<pre>public double dividir(double operando1, double operando2) throws ArithmeticException</pre> <p>Método para dividir los operandos.</p> <p>Parameters: operando1 - Primer operando. operando2 - Segundo operando.</p> <p>Returns: Resultado de la división de los operandos.</p> <p>Throws: <code>ArithmeticException</code> - Si el segundo operando es cero.</p>

Imagen 12: (3/4) Documentación generada de la clase CalculadoraEjemplo

[Volver al índice](#)

getOperando1
<pre>public double getOperando1()</pre> <p>Método para obtener el valor del primer operando.</p> <p>Returns: Valor del primer operando.</p>
getOperando2
<pre>public double getOperando2()</pre> <p>Método para obtener el valor del segundo operando.</p> <p>Returns: Valor del segundo operando.</p>
getResultado
<pre>public double getResultado()</pre> <p>Método para obtener el valor del resultado.</p> <p>Returns: Valor del resultado.</p>
presentarResultado
<pre>public void presentarResultado(String operacion, double numero)</pre> <p>Método que presenta el resultado de las operaciones y les da formato para presentar resultado con dos decimales.</p> <p>Parameters: operacion - el nombre de la operación efectuada numero - el resultado obtenido</p>

Imagen 13: (4/4) Documentación generada de la clase CalculadoraEjemplo

[Volver al índice](#)