



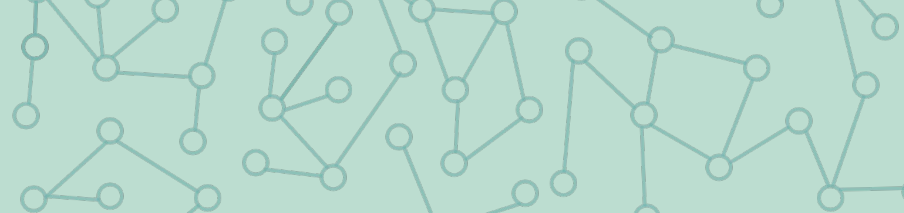
Clase N°2

Elementos básicos

Taller de programación Computines



Hoy aprenderemos



- **Crear un programa nuevo de Java que:**
 - Pida valores al usuario
 - El programa ejecute acciones con los valores
 - Se entregue al usuario los resultados de las acciones
 - Programar es frustrante, pero muy enriquecedor

Java

- Lenguaje de Programación Orientado a Objetos
- Objeto o clase:
 - Atributos (características)
 - Métodos (funcionalidades)
- Usaremos Java 8



Objetos

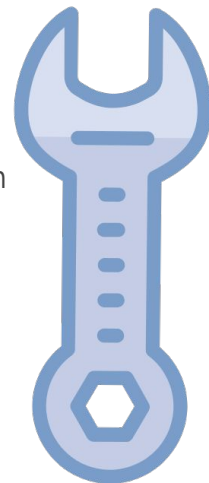
- **Martillo**

- **Atributos:**
 - Dimensiones
 - Material
 - Peso
- **Funciones:**
 - Clavar
 - Sacar un clavo



- **Llave**

- **Atributos:**
 - Dimensiones
 - Material
 - Tamaño de tuercas que sirven
- **Funciones:**
 - Apretar tuercas
 - Soltar tuercas



Nueva clase principal



Main es el objeto principal en el que escribimos nuestro código. Todo lo que se encuentra dentro de Main son distintas funciones.

La función main será la función principal que ejecutará el programa.

Nueva clase principal

Main es el objeto principal en el que escribimos nuestro código. Todo lo que se encuentra dentro de Main son distintas funciones.

La función main será la función principal que ejecutará el programa.

```
1  
Clase → 2 public class Main {  
3  
Función → 4 public static void main(String[] args) {  
5  
6  
7     }  
8  
9 }
```

Nueva clase principal

Main es el objeto principal en el que escribimos nuestro código. Todo lo que se encuentra dentro de Main son distintas funciones.

La función main será la función principal que ejecutará el programa.

```
1
Clase → 2 public class Main {
3
Función → 4 public static void main(String[] args) {
5
6
Acá escribe tu código → 7 }
8
9 }
```

Nueva clase

Main es el objeto principal
se encuentra dentro de

La función main será la

```
1  
2 Clase → public class  
3  
4 Función → public  
5  
6 Acá escribe tu código →  
7 }  
8  
9 }
```

```
FUNCTION()  
{  
  //CODE  
}
```

```
FUNCTION() {  
  //CODE  
}
```

estoy escribiendo este código. Todo lo que
está dentro de las llaves {}

es el código que ejecutará el programa.

```
args) {
```

MARVEL
CAPTAIN AMERICA
CIVIL WAR



El programa irá leyendo y ejecutando línea por línea

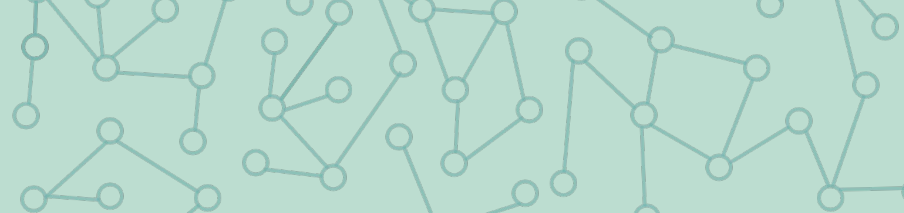
El orden de las instrucciones es crucial



Elementos básicos de Java



Variables



- Son elementos a los que se les asignan valores para poder trabajar con ellos
- Pueden tener distintos tipos

Variables - analogía



- Distintas cajas para distintos tipos
- Le puedo asignar una etiqueta a cada caja
- Puedo cambiar el contenido de la caja sin cambiar su etiqueta

Variables - conceptos

Crear	Asignar
Crea la variable, indicando nombre y el tipo. Solo se puede crear una variable con el mismo nombre y el nombre debe ser una sola palabra (sin espacios)	Le da un valor a la variable, debe existir la variable antes y el valor a asignar debe ser del mismo tipo que la variable creada.
<code>int number;</code>	<code>number = 0;</code>

Variables

En este programa estamos creando una variable de tipo **int** (número entero o **Integer** en inglés), asignándole el valor 0.

```
public class Main {  
    public static void main(String[] args) {  
  
        int number;  
        number = 0;  
  
    }  
}
```

Variables

Crear y asignar un valor:

```
public class Main {  
    public static void main(String[] args) {  
  
        int number = 0;  
  
    }  
}
```

Variables

Cambiar el valor:

```
public class Main {  
    public static void main(String[] args) {  
  
        int number = 0;  
        number = 42;  
  
    }  
}
```


Tipos de datos

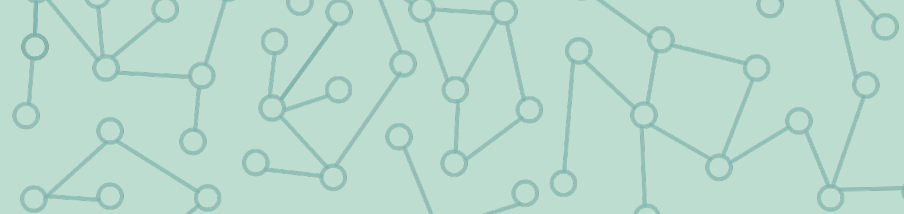
Atributos asociados a cada dato que indican al computador cómo administrarlos. Se destacan:

- 1) Número entero (**int**)
- 2) Números reales (**double**)
- 3) Caracteres (**char**)
- 4) Texto (**String**)
- 5) Valores de verdad (**boolean**)

Ejemplo de asignaciones

```
public class Main {  
    public static void main(String[] args) {  
  
        int myNumber = 0;  
  
        String myText = "Soy un texto";  
        myText = "Ahora soy otro texto";  
  
        boolean myBoolean;  
        myBoolean = true;  
        myBoolean = false;  
    }  
}
```

Expresiones



Expresiones básicas para hacer operaciones entre variables

- 1) Suma, resta, multiplicación y división (+, -, *, /)
- 2) Módulo, que es el resto de la división entera entre dos números (%)
- 3) Comparaciones (==, <=, >=, >, <)

Ejemplo de expresiones

```
public class Main {  
    public static void main(String[] args) {  
        int a = 7;  
        int b = 4;  
  
        int result = a + b;  
        // El valor de result es 11  
  
        result = a - b;  
        // Ahora el valor de result es 3  
  
        result = 2 * a;  
        // Ahora el valor de result es 14  
  
        result = a / b;  
        // Ahora el valor de result es 1  
  
        result = a % b;  
        // ahora el valor de result es 3  
  
    }  
}
```

Actividad en grupo

```
public class Main {  
    public static void main(String[] args) {  
        int a = 7;  
        int b = 4;  
  
        int result = a + b;  
        // El valor de result es 11  
  
        result = a - b;  
        // Ahora el valor de result es 3  
  
        result = 2 * a;  
        // Ahora el valor de result es 14  
  
        result = a / b;  
        // Ahora el valor de result es 1  
  
        result = a % b;  
        // ahora el valor de result es 3  
  
    }  
}
```

¿Cuáles patrones pueden observar en éste código?

1. ¿Cómo terminan algunas líneas? ¿Por qué es así?
2. Identifica los siguientes elementos:
 - a. Creaciones de una variable
 - b. Asignaciones de una variable
 - c. Comentarios
 - d. Expresiones
3. ¿Cuál es el valor de a y b al final del código?
4. ¿Cuál es la diferencia entre las primeras tres líneas? ¿Por qué?

Expresiones

La división entre dos números enteros debe ser un entero. Si deseamos obtener la división real podemos hacer lo siguiente:

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int a = 7;  
        int b = 4;  
  
        double result = a / b;  
        // El valor de result es 1  
  
        double result2 = ((double) a) / ((double) b)  
        // El valor de result es 1.75  
  
    }  
  
}
```

Asignación a sí mismo

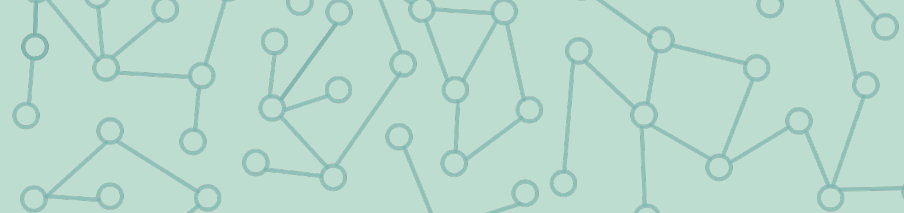
Veamos un ejemplo donde defino el valor de una variable en base a sí misma, con ayuda de expresiones aritméticas

```
public class Main {  
    public static void main(String[] args) {  
        int a = 3;  
        a = a+1;  
        // ahora a vale 4  
  
        a = 2*a;  
        // ahora a vale 8  
    }  
}
```

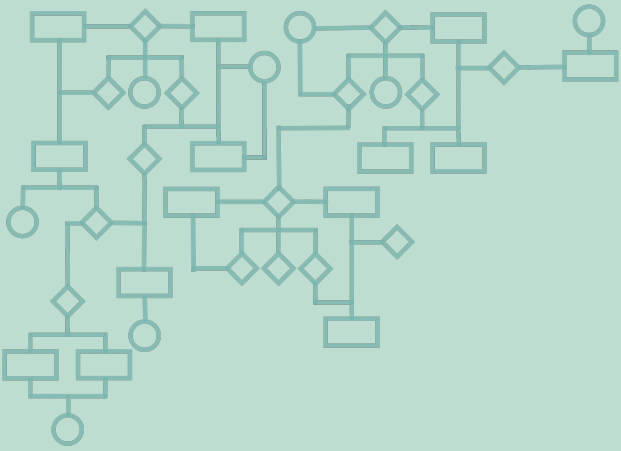
Comparaciones

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int a = 3;  
        boolean comparison = (a == 3);  
        // comparison es true  
  
        int b = 4;  
        comparison = (a == b);  
        // comparison ahora es false  
  
        comparison = (a >= 10);  
        // comparison es false, porque a no es mayor o igual a 10  
  
    }  
  
}
```


Comparaciones



Ojo! El resultado de una comparación es de tipo **boolean** (verdadero o falso)



Output



Output

Ahora haremos un programa que imprima “Hola mundo!” al usuario

```
public class Main {  
  
    public static void main(String[] args) {  
  
        System.out.println("Hola mundo!");  
  
    }  
  
}
```

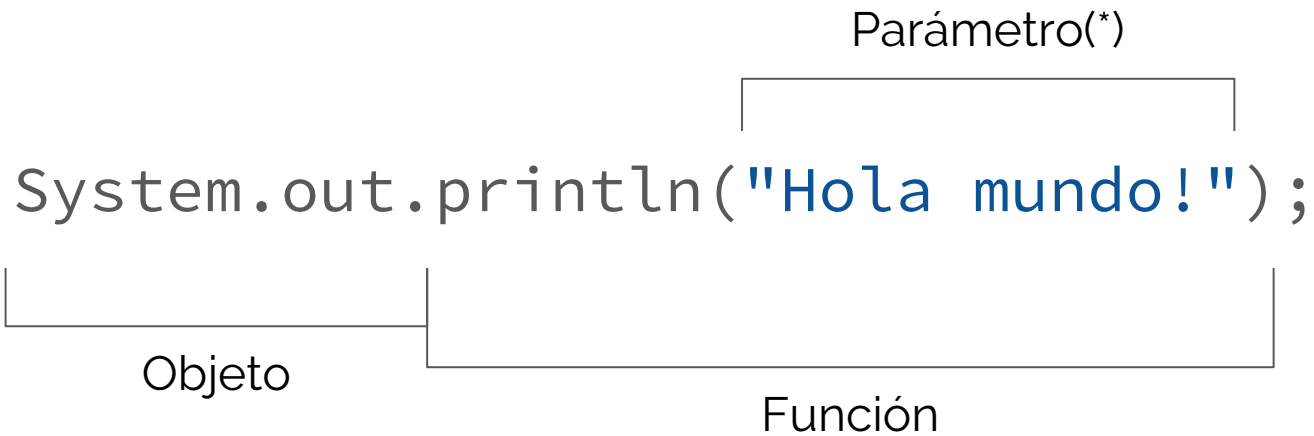
Output

Parámetro(*)

```
System.out.println("Hola mundo!");
```

Objeto

Función



(*) En este caso está recibiendo "Hola Mundo", un String.

Output

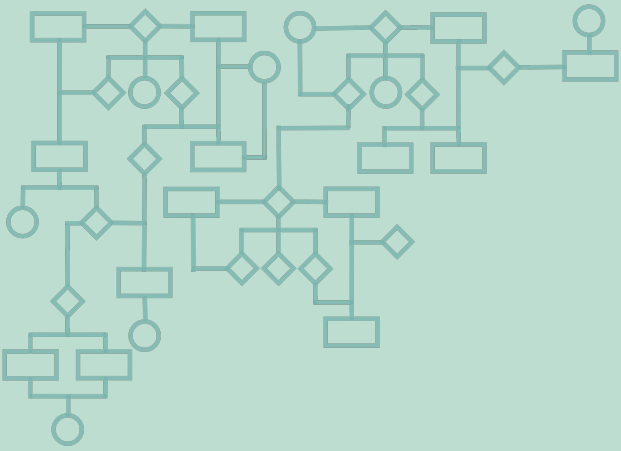
Para mostrar información en pantalla al usuario usamos la función `System.out.println`

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int a = 2;  
        int result = 2*a;  
  
        System.out.println(result);  
  
    }  
  
}
```

Output

Podemos juntar lo que queremos imprimir junto a texto

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int a = 2;  
        int result = 2*a;  
  
        System.out.println("El resultado es: " + result);  
  
    }  
}
```



Input



Importar herramientas (librerías)



Tenemos la siguiente situación:

- 1) Estás haciendo un trabajo en tu casa y te das cuenta que no tienes una herramienta en especial en tu caja de herramientas.
- 2) Vas al Homecenter y compras esa herramienta.

Esta nueva herramienta te da la capacidad de hacer tareas que antes no podías.

En programación, a esto se le llama "importar librerías"

Scanner

Para utilizarlo hay que importarlo

```
import java.util.Scanner;
```

Scanner

Para utilizarlo hay que importarlo

```
import java.util.Scanner;
```

Lo tenemos que crear y asignar a una variable

```
Scanner scanner = new Scanner(System.in);
```

Scanner

Para utilizarlo hay que importarlo

```
import java.util.Scanner;
```

Lo tenemos que crear y asignar a una variable

```
Scanner scanner = new Scanner(System.in);
```

Después, le podemos pedir que nos diga lo que se ingresa

```
String linea = scanner.nextLine();
```

```
int numero = scanner.nextInt();
```

Scanner

Para utilizarlo hay que importarlo

```
import java.util.Scanner;
```

Lo tenemos que crear y asignar a una variable

```
Scanner scanner = new Scanner(System.in);
```

Después, le podemos pedir que nos diga lo que se ingresa

```
String linea = scanner.nextLine();
```

```
int numero = scanner.nextInt();
```

Notemos que hay que importar el Scanner, y que el resultado de nextLine se guarda como un String. Lo siguiente que le pedimos es un entero.

Ejemplo

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

        System.out.println("Ingrese su nombre: ");

        Scanner scanner = new Scanner(System.in);
        String username = scanner.nextLine();

        System.out.println("Hola " + username);

    }
}
```

HackerRank

Nos permite hacer un set de ejercicios y probar nuestro código con varios ejemplos

Cada *challenge* es un problema

C02 - Elementos básicos [Details ▶](#)

Challenges



¡Hola mundo!



Success Rate: 100.00% Max Score: 10 Difficulty: Easy

Solve Challenge

Mensaje OCI



Success Rate: 100.00% Max Score: 10 Difficulty: Easy

Solve Challenge

✓ ¡Hola tú!



Success Rate: 100.00% Max Score: 10 Difficulty: Easy

Try Again

¡Hola tú! (con edad)



Success Rate: 100.00% Max Score: 10 Difficulty: Easy

Solve Challenge

Suma de input



Success Rate: 100.00% Max Score: 10 Difficulty: Easy

Solve Challenge

HackerRank - Challenge

¡Hola mundo!

by [compuctines](#)

Problem

Submissions

Leaderboard

Discussions

Tu programa debe imprimir el mensaje: ¡Hola mundo!

Input Format

No hay input.

Output Format

El retorno debe ser un string que diga: Hola mundo!

Sample Output 0

```
Hola mundo!
```

Explanation 0

Tu programa debe imprimir: "Hola mundo!"



Submissions: 1

Max Score: 10

Difficulty: Easy

Rate This Challenge:



[More](#)

Admin Options

[Edit Challenge](#)

[View Submissions](#)

Current Buffer (saved locally, editable)

Java 8



```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
8     }
9 }
```

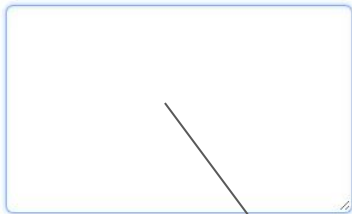
Line: 1 Col: 1

Input y output en HackerRank

Input:

 [Upload Code as File](#)

☒ Test against custom input



→
Escribe aquí tu input
para probar tu código

Run Code

Submit Code

↓
Haz click y se mostrará
lo que escribe tu código

Input y output en HackerRank

Output:

Input (stdin)

Your Output (stdout)

Hola mundo!

Expected Output

Hola mundo!

Errores



Errores

! error: incompatible types: possible lossy conversion from double to int

```
int r = ((double) a) / ((double) b);
```

! error: ';' expected

```
result = ((double) a) / ((double) b)
```

Programar es frustrante, pero muy enriquecedor!

Errores

Traducción error: tipos incompatibles: posible pérdida al convertir desde double a int

! error: incompatible types: possible lossy conversion from double to int

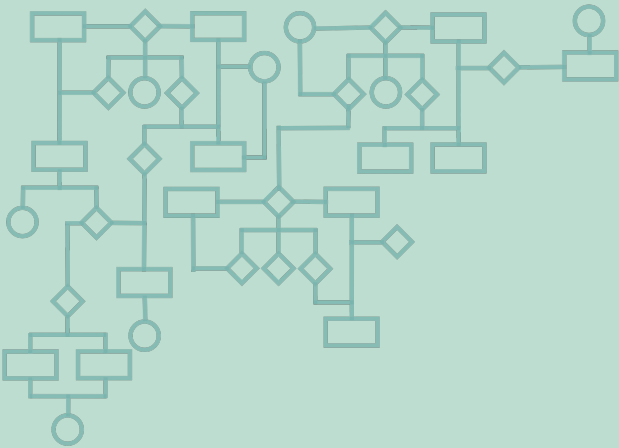
```
int r = ((double) a) / ((double) b);
```

Traducción error: se esperaba un ';'

! error: ';' expected

```
result = ((double) a) / ((double) b)
```

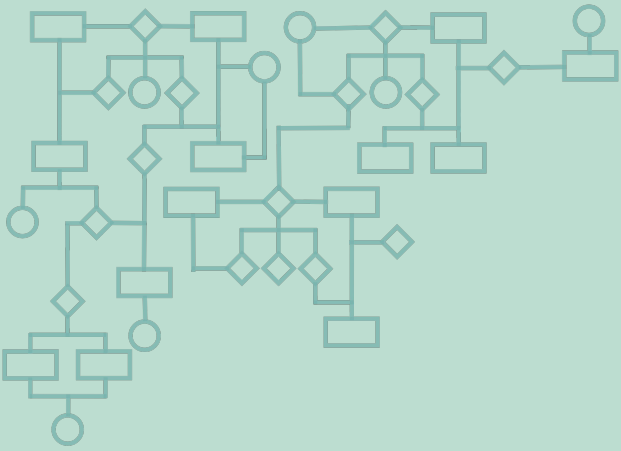
Programar es frustrante, pero muy enriquecedor!



Elementos básicos

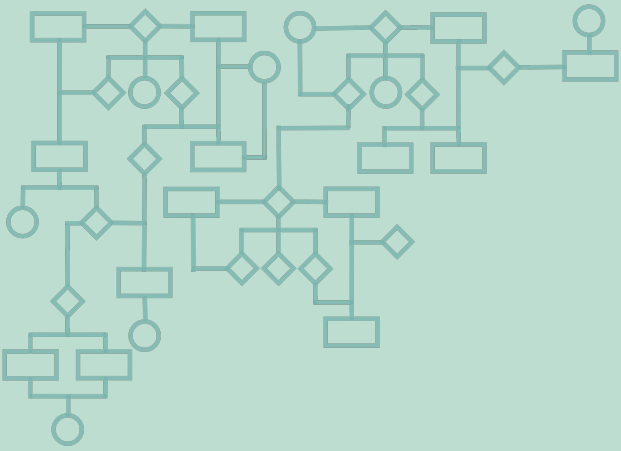
Taller de programación Computines





Repaso: iMentimeter!

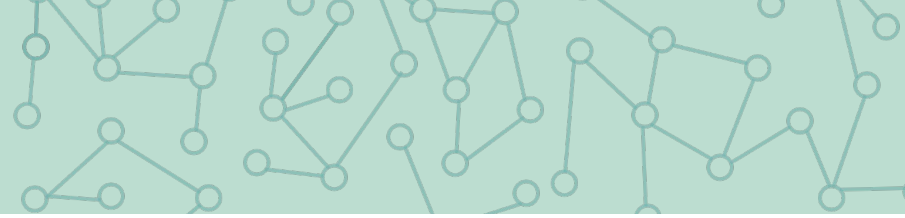




Eclipse

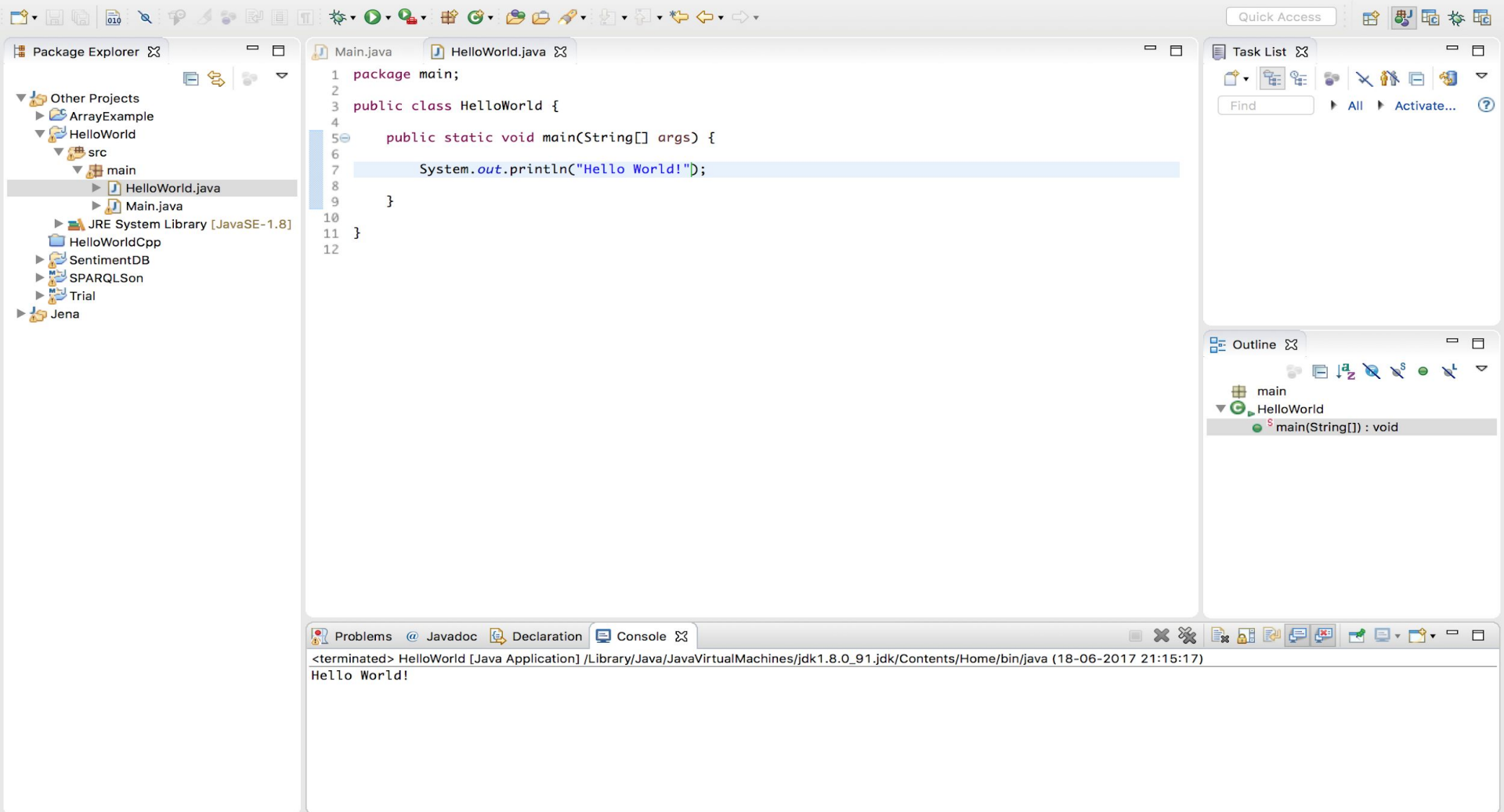


Eclipse

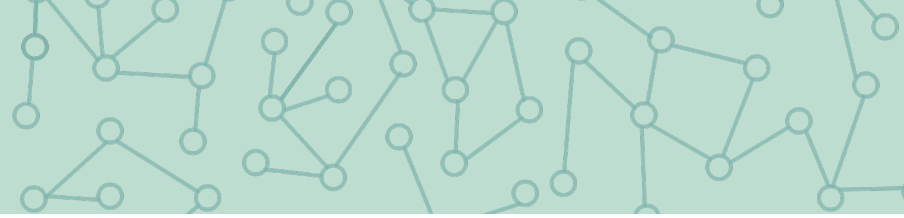


Eclipse es una herramienta que nos permite escribir nuestros programas en Java



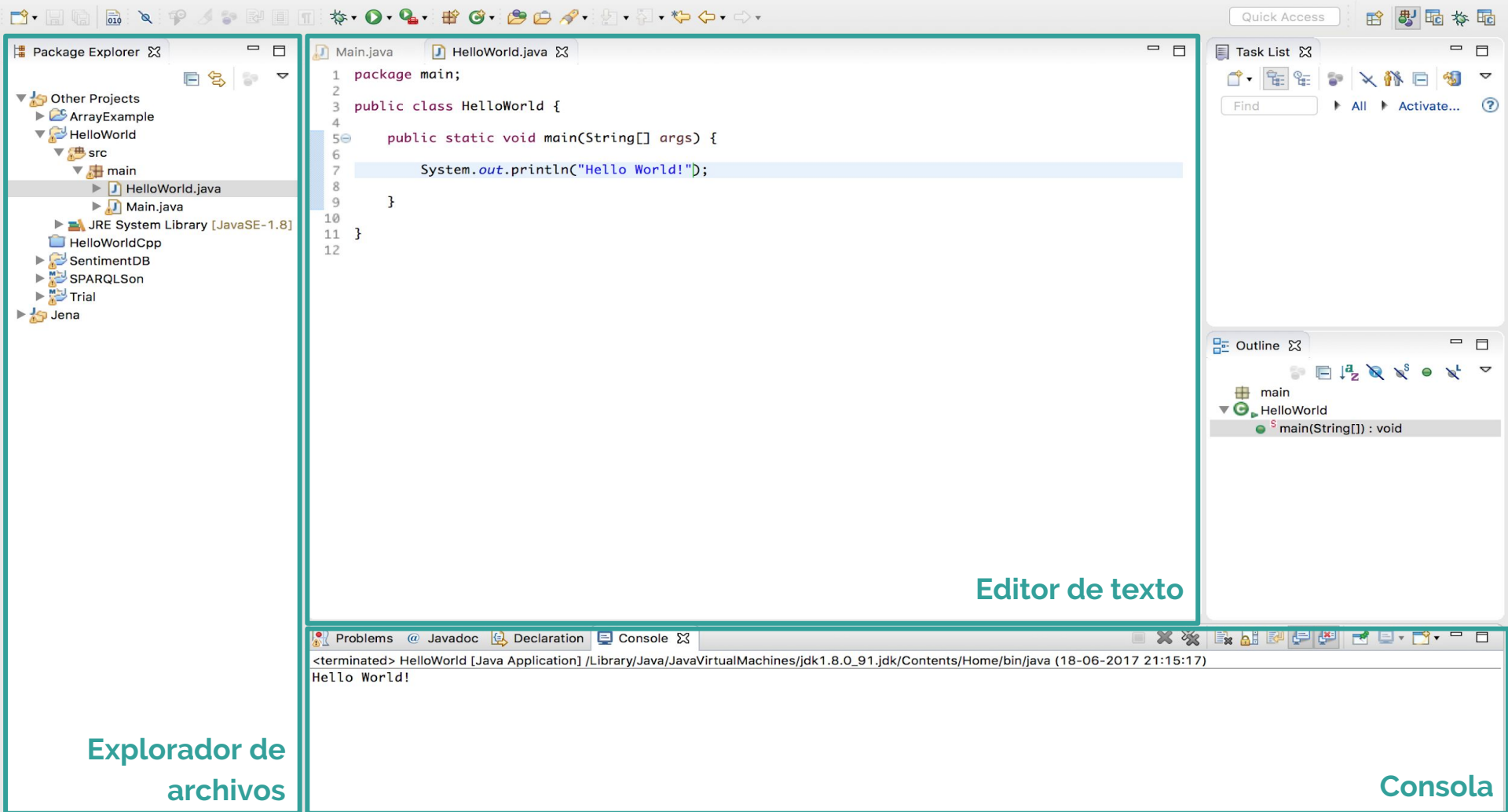


Eclipse



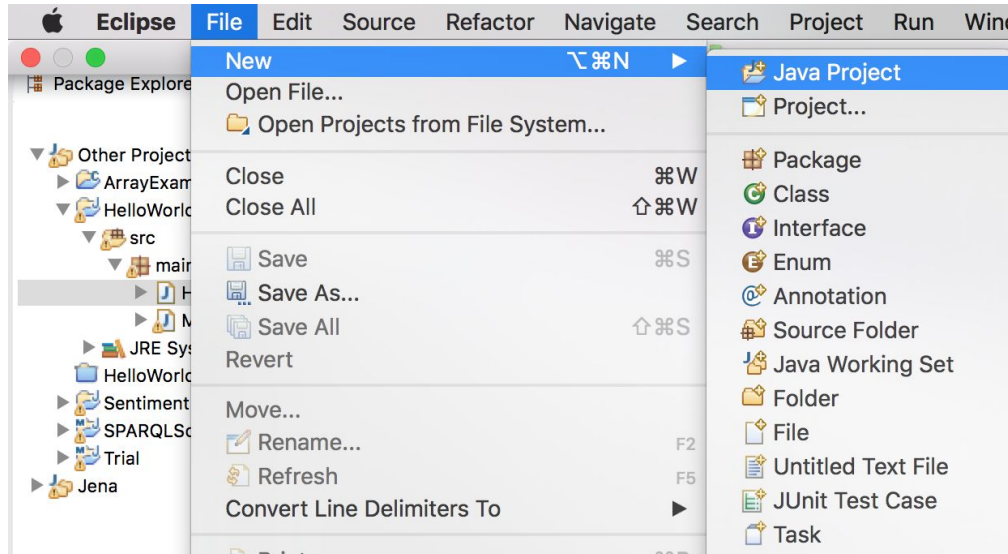
Podemos destacar:

- 1) **Package Explorer:** nos permite navegar a través de nuestros distintos proyectos
- 2) **Editor de texto:** es la ventana central, donde escribimos nuestros programas
- 3) **Consola:** lugar donde se interactúa con el usuario, mostrando resultados o pidiendo valores



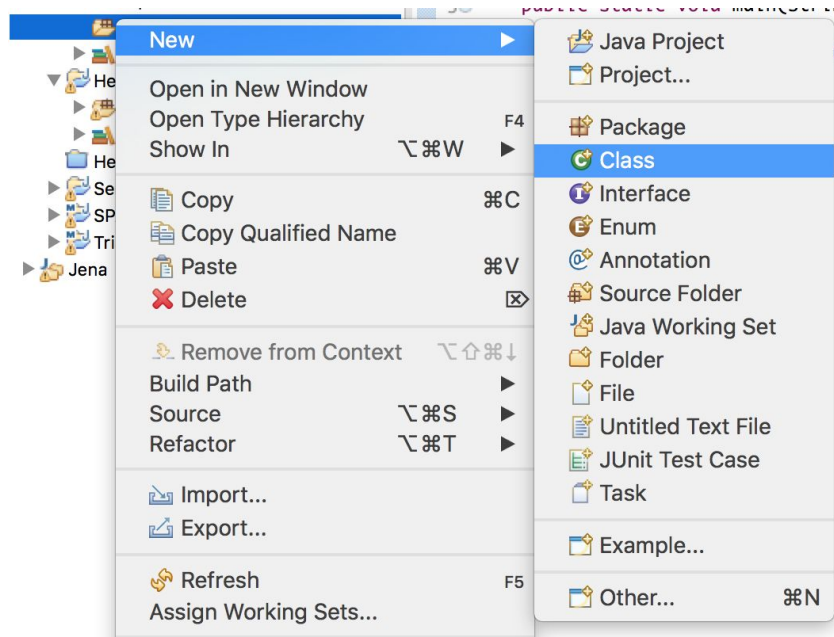
Crear un proyecto en Eclipse

Para crear un nuevo proyecto nos vamos a Archivo (File), después a New y luego Java Project

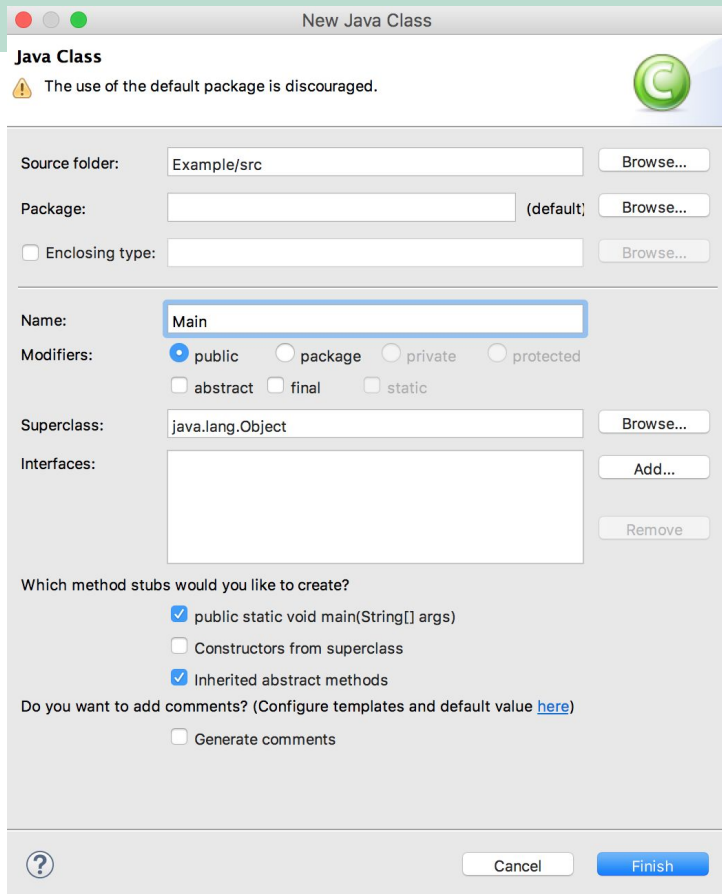


Crear una nueva clase en Eclipse

Una vez creado un proyecto, le hacemos click con el botón derecho y hacemos click a New y luego Class



Crear una nueva clase en Eclipse



The screenshot shows the 'New Java Class' dialog box in the Eclipse IDE. The title bar says 'New Java Class'. Below the title bar, there's a warning icon and text: 'The use of the default package is discouraged.' To the right is the Eclipse logo. The dialog is divided into several sections. The 'Source folder' section has a text field with 'Example/src' and a 'Browse...' button. The 'Package' section has a text field with '(default)' and a 'Browse...' button. There's a checkbox for 'Enclosing type' with a 'Browse...' button. The 'Name' section has a text field with 'Main'. The 'Modifiers' section has radio buttons for 'public' (selected), 'package', 'private', and 'protected', and checkboxes for 'abstract', 'final', and 'static'. The 'Superclass' section has a text field with 'java.lang.Object' and a 'Browse...' button. The 'Interfaces' section has a text area and 'Add...' and 'Remove' buttons. The 'Which method stubs would you like to create?' section has checkboxes for 'public static void main(String[] args)' (checked), 'Constructors from superclass', and 'Inherited abstract methods' (checked). The 'Do you want to add comments? (Configure templates and default value [here](#))' section has a checkbox for 'Generate comments'. At the bottom, there's a question mark icon, a 'Cancel' button, and a 'Finish' button.

Java Class

⚠ The use of the default package is discouraged.

Source folder: Example/src Browse...

Package: (default) Browse...

☐ Enclosing type: Browse...

Name: Main

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☒ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

? Cancel Finish

Luego le agregamos un nombre (generalmente **Main** o **Principal**), y hacemos click al casillero:

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