|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Ing. Guadalupe Lizeth Parrales Romay |
| *Asignatura:* | Programación Orientada a Objetos |
| *Grupo:* | 04 |
| *No de Práctica(s):* | 01 |
| *Integrante(s):* | Becerril Olivar Axel Daniel. |
| *No. de Equipo de cómputo empleado:* | 317113888 (Entrega en línea) |
| *No. de Lista o Brigada:* | 317113888 (Entrega en línea) |
| *Semestre:* | 2021-2 |
| *Fecha de entrega:* | 10/03/21 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Introducción**

En Java existen 57 tipos de palabras reservadas, mismas que no pueden ser utilizadas como nombre de una variable, a continuación, se mostrara cada una de las palabras con su debido significado…

|  |  |  |  |
| --- | --- | --- | --- |
| **PALABRA** | **SIGNIFICADO** | **PALABRA** | **SIGNIFICADO** |
| abstract | Implementa abstracciones en Java | else | Se utiliza junto con la palabra if para crear una sentencia |
| assert | Describe si una declaración es verdadera o falsa | enum | Se utiliza para crear enumeraciones |
| boolean | Define una variable booleana, ya sea “true” o ”false” | extends | Especifica una superclase |
| break | Finaliza un ciclo | final | Se usa en una variable que no puede ser anulada |
| byte | Entero de 8 bits | finally | Específica que cierta parte del código siempre se ejecutara |
| case | Se utiliza parae valuar alguna expresión | float | Declara un punto flotante |
| catch | Especifican que hacer en un caso determinado | for | Se ocupara para crear un ciclo |
| char | Declara una variable tipo carácter | goto | Instrucción de salto |
| class | Define la implementación de un objeto | if | Se usa para indicar una estructura de control |
| const | No se utiliza | implements | Define la interfaz de una clase |
| continue | Rompe un ciclo permitiéndolo realizar iteraciones | import | Se utiliza para importar paquetes |
| default | Se usa para declarar valores predeterminados | instanceof | Evalúa si un objeto es compatible con la interfaz |
| do | Se utiliza junto con While para crear un ciclo | int | Declara una variable tipo entero |
| double | Declara una variable punto flotante | interface | Declara una clase especial |
| long | Declara una variable tipo entero | public | Modificador de acceso publico |
| native | Indica que un método no esta en el mismo archivo | return | Finaliza un método regresando un valor |
| new | Crea un objeto nuevo | short | Declara una variable tipo entero |
| package | Agrupa las clases | static | Indica que un método o campo es único en la clase |
| private | Modificador de acceso privado | strictfp | Permite restringir el redondeo de cálculos |
| protected | Modificador de acceso protegido | super | Manda a llamar una superclase |
| switch | Es una instrucción de control | transcient | Indica que un atributo no sea serializado |
| synchronized | Se utiliza para obtener un bloqueo mutex | try | Define un bloque de código para que tenga excepciones |
| this | Hace referencia a un objeto | void | Es un tipo de dato que no regresa valores |
| throw | Permite realizar una excepción en un método | volatile | Indica que una variable se modifica a través de subprocesos o hilos |
| throws | Permite realizar excepciones en un método | while | Sirve para crear un ciclo |

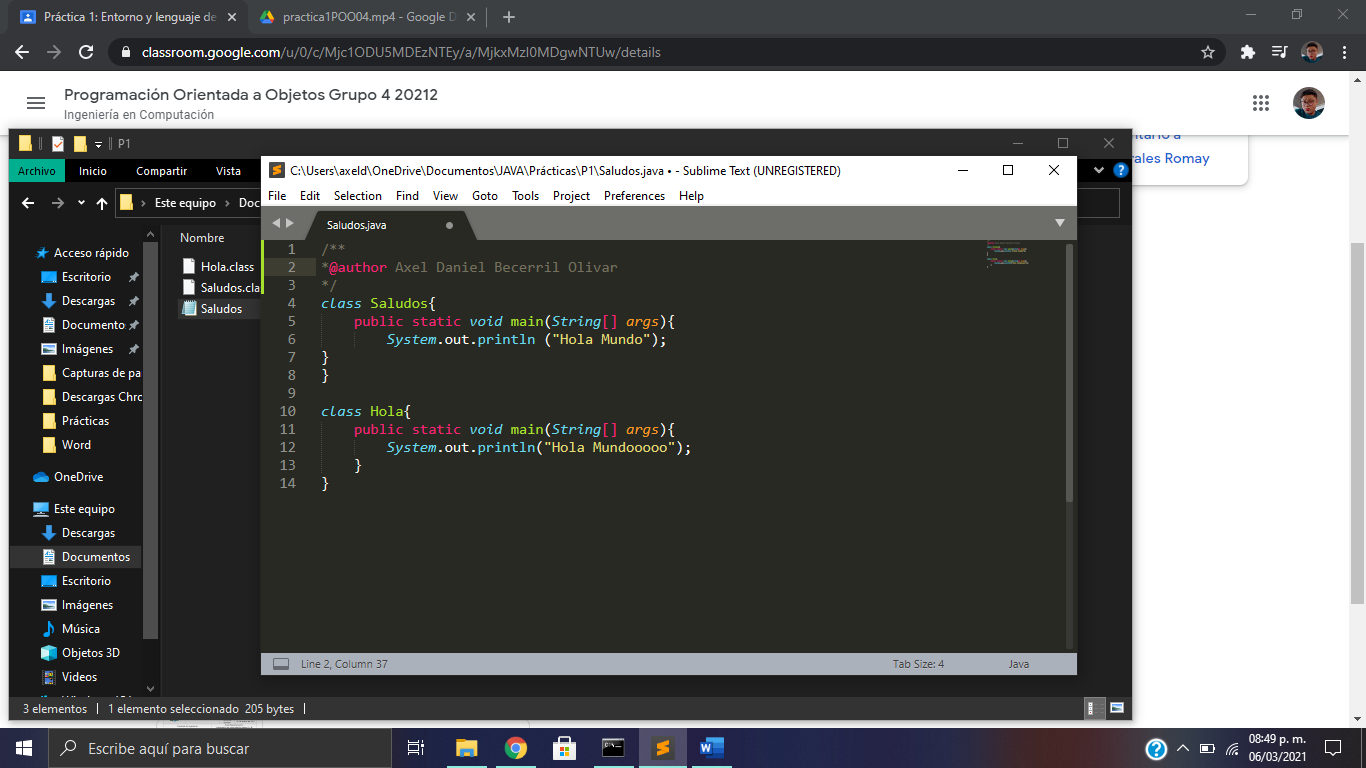
También existen las llamadas etiquetas de documentación en Java, sirven para que en el futuro otros desarrolladores o incluso nosotros podamos saber para que sirve cada clase, interfaz o método.

|  |  |
| --- | --- |
| Etiqueta | Descripción |
| @author | Muestra el nombre del programador |
| @see | Hace referencia de un método con otro |
| @param | Se utiliza en todos los parámetros que utiliza un método, ya que indica su parámetro |
| @return | Indica que es lo que va a regresar un método |
| @deprecated | Sirve para decir que métodos o clases son obsoletos |

**Desarrollo**

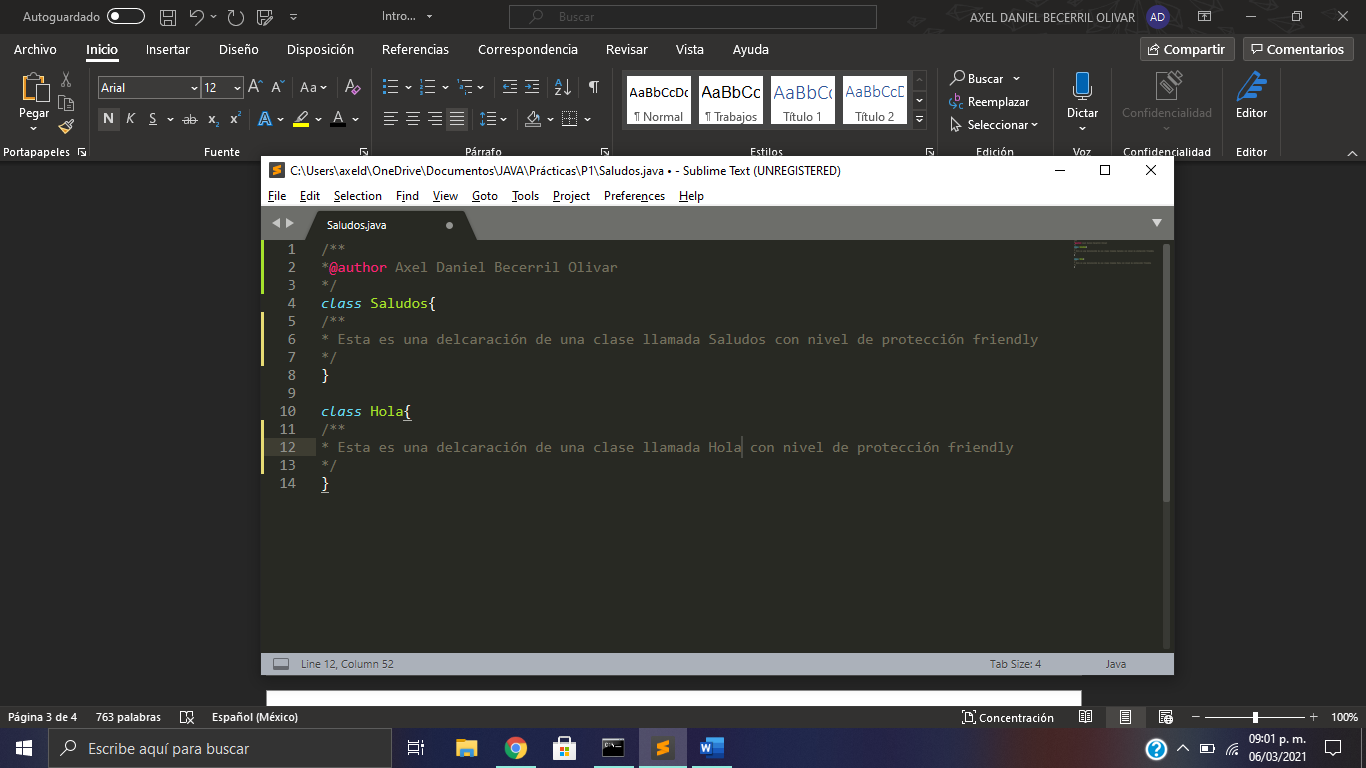
1. **Cree un archivo fuente de Java con el nombre Saludos.java en el editor de texto plano de su elección.**

En la captura se puede observar el archivo fuente creado con su respetivo nombre.



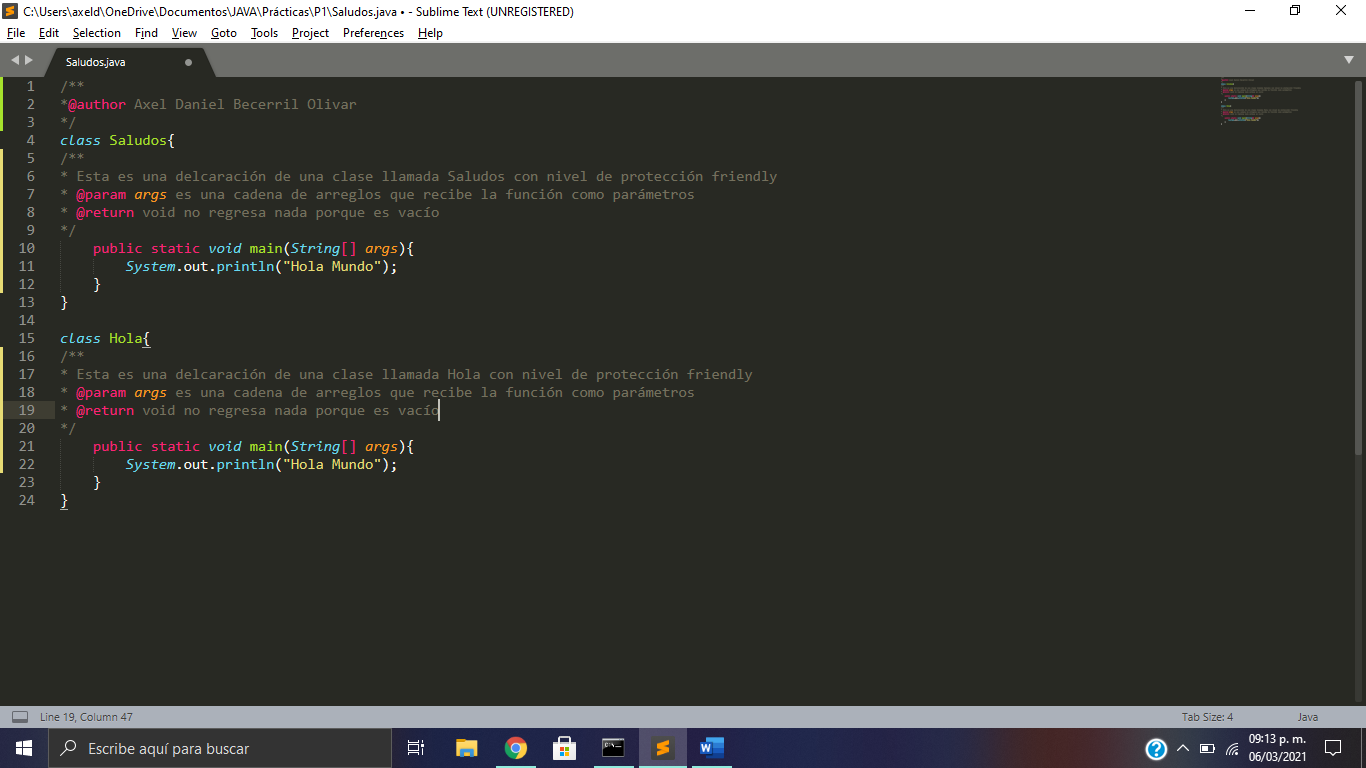
1. **Dentro de este archivo cree la clase Hola y la clase Saludos, ambas sin un modificador de acceso (nivel de protección por defecto).**

Se puede observar como se crearon las clases con su nivel de protección friendly.

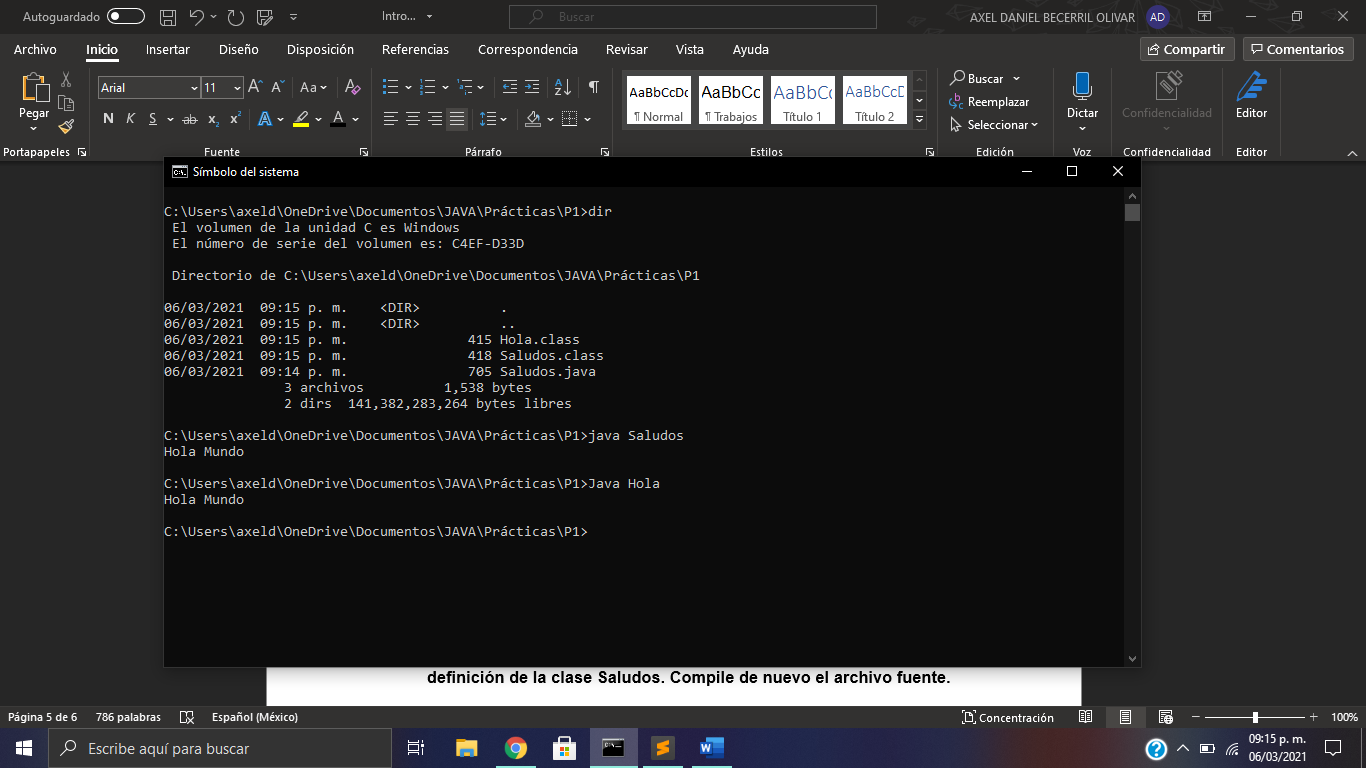


1. **Dentro de la definición de cada clase cree el método main y dentro de este codifique la impresión del mensaje "Hola Mundo".**

Se creó en ambas clases la impresión del mensaje.

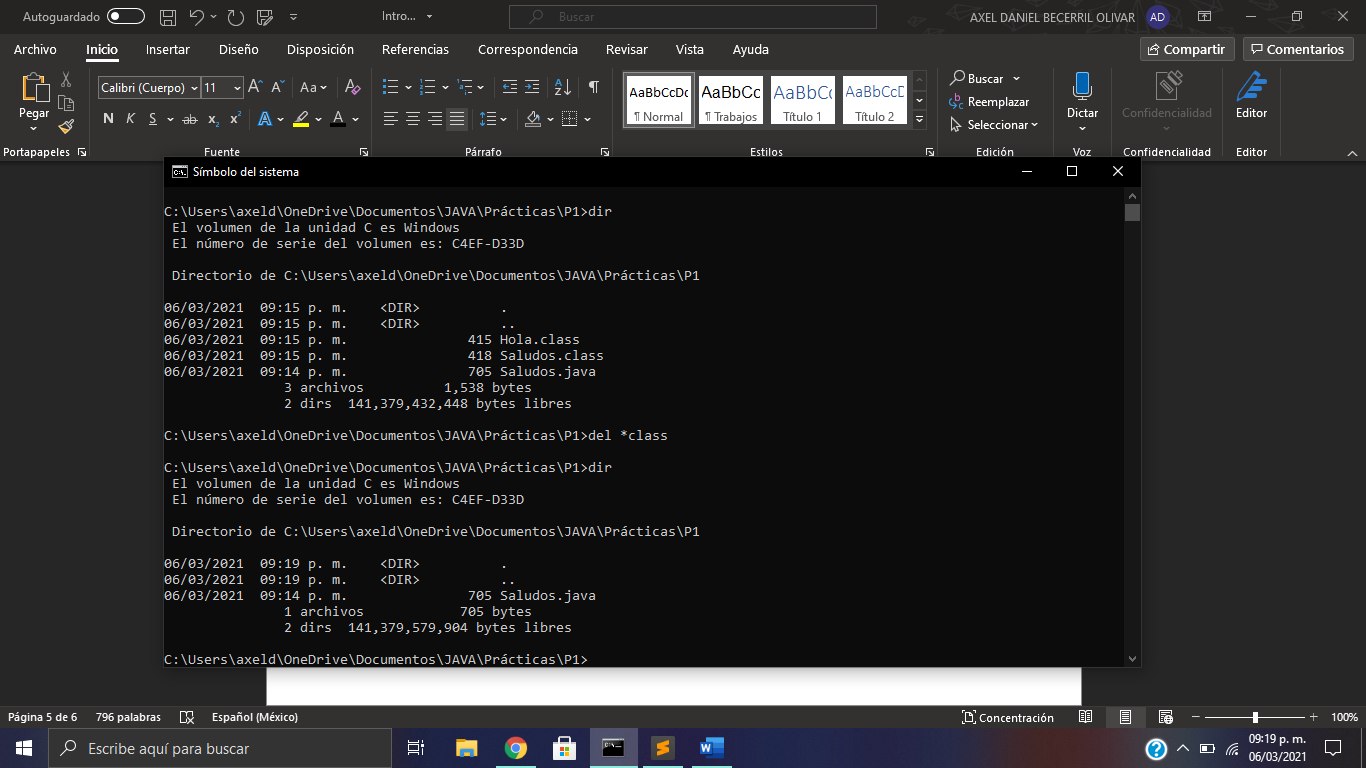


1. **Compile el archivo fuente y ejecute las clases generadas.**

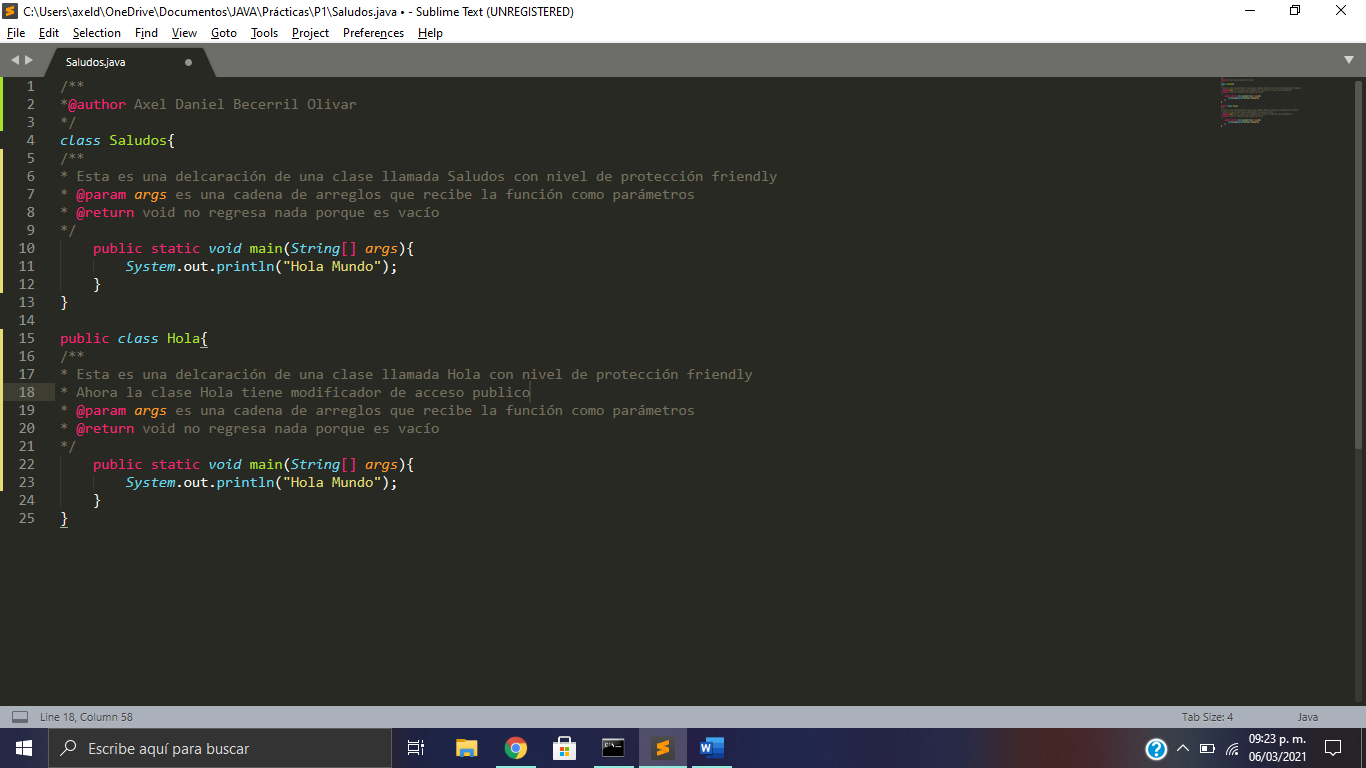
Se ejecutaron ambas clases y se imprimió correctamente el mensaje.

1. **Elimine de su carpeta de trabajo los archivos .class, modifique su archivo fuente agregando el modificador de acceso público a la definición de la clase Hola. Compile de nuevo el archivo fuente.**

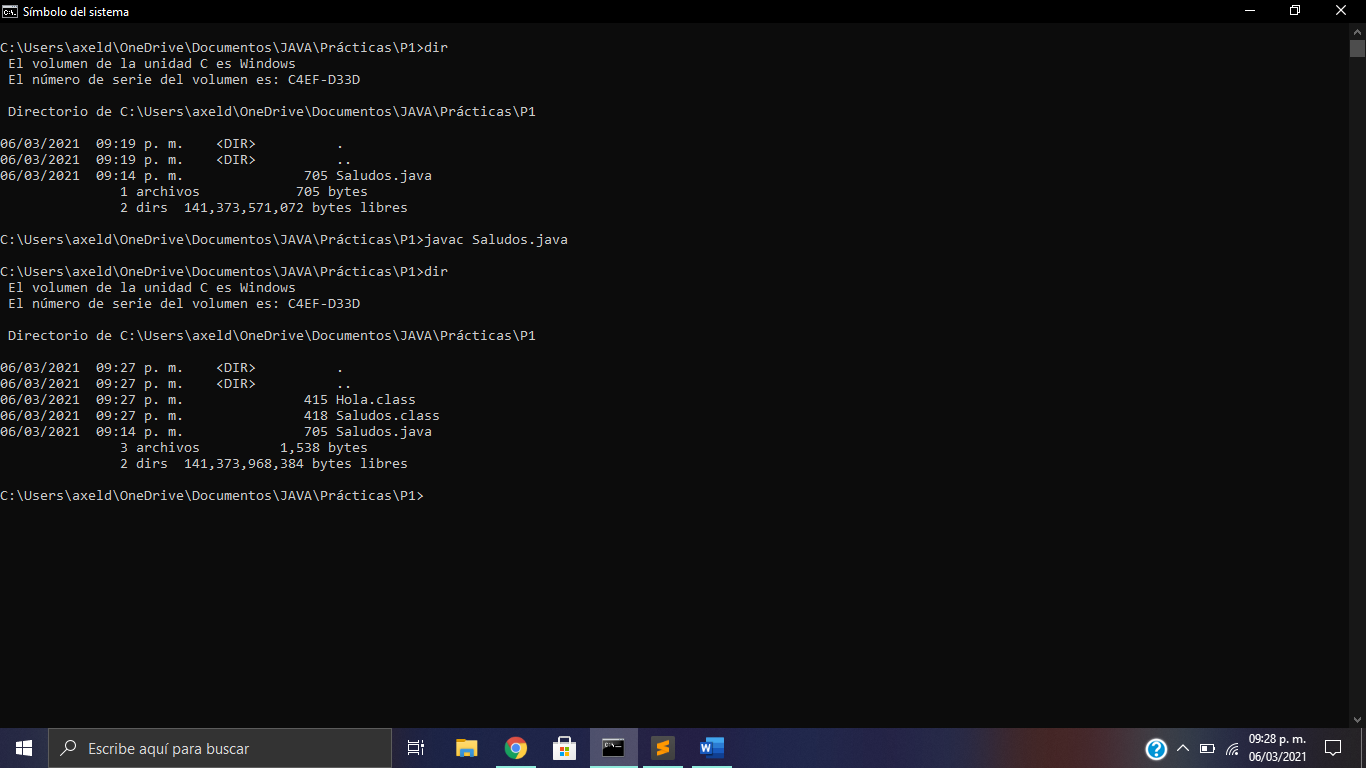
Se eliminaron correctamente los archivos .class



Ahora se modificó la clase Hola para que dejará de tener nivel de protección friendly y ser publico.

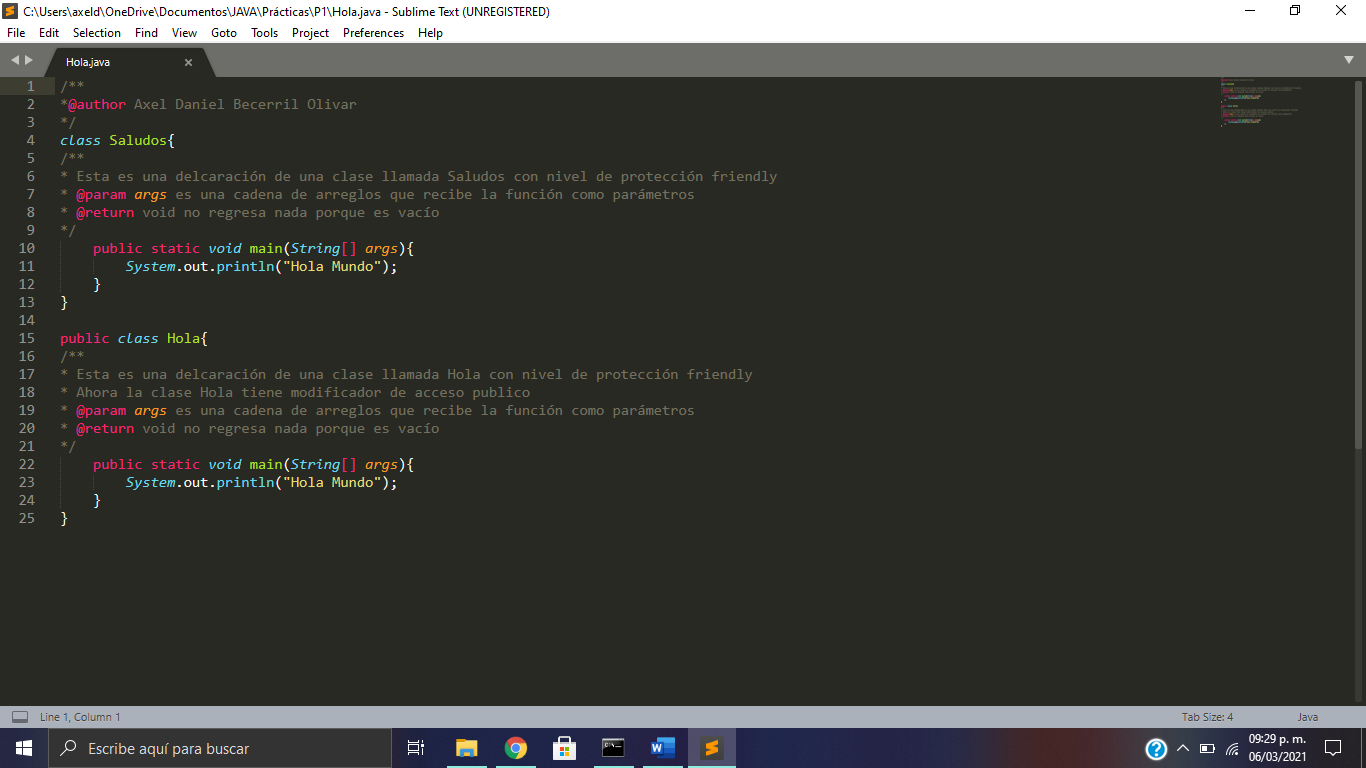


A continuación, se compiló nuevamente el código fuente de manera correcta.

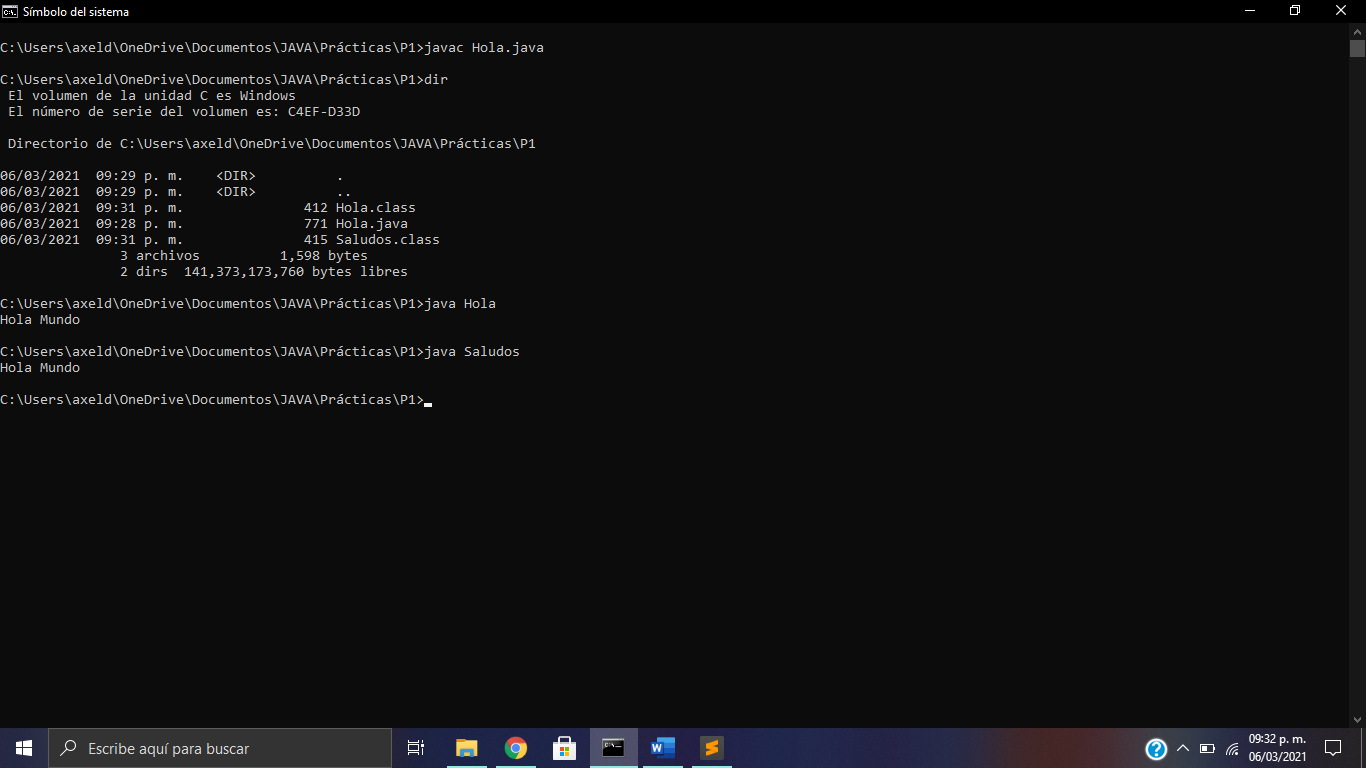


1. **Cambie el nombre del archivo fuente a Hola.java, compile de nuevo el archivo fuente y ejecute las clases generadas.**

Se cambió el nombre del archivo.

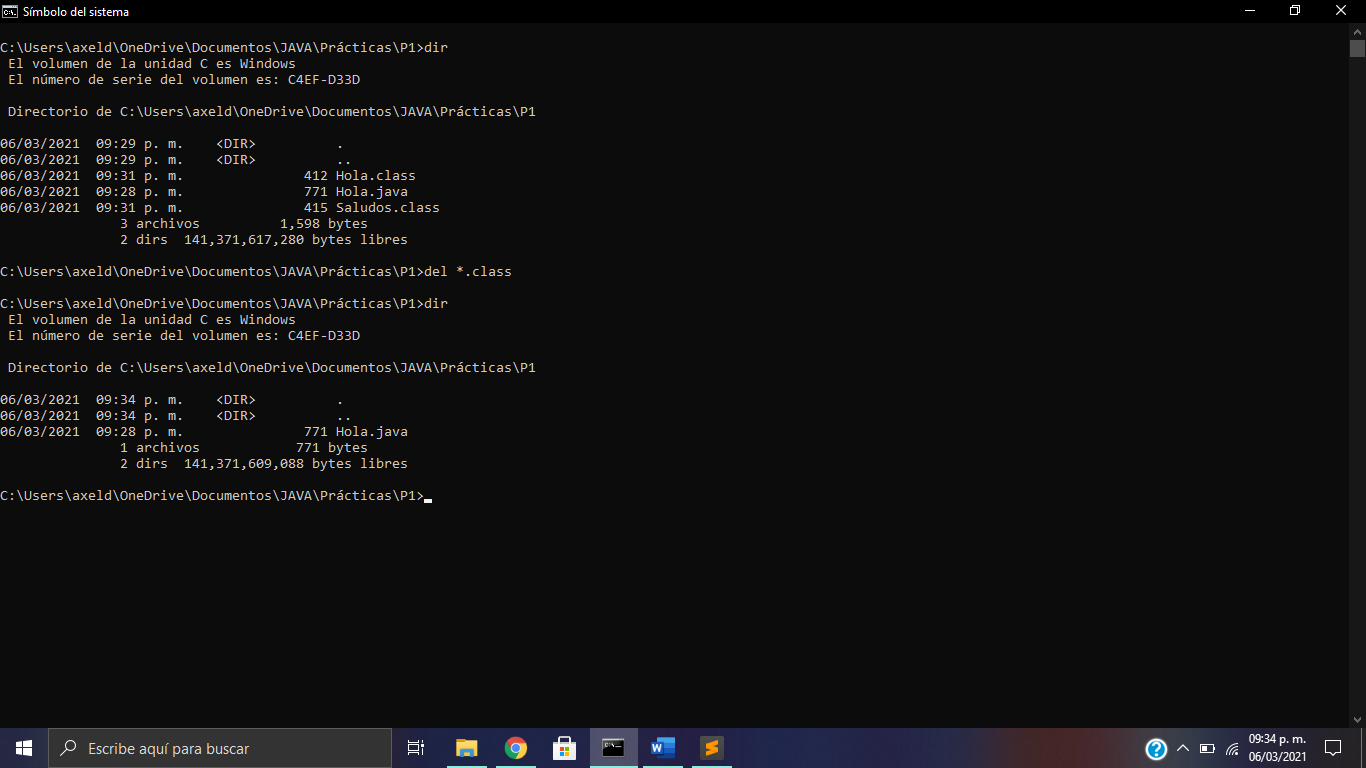


Se compiló el archivo y pudo ser ejecutado correctamente.

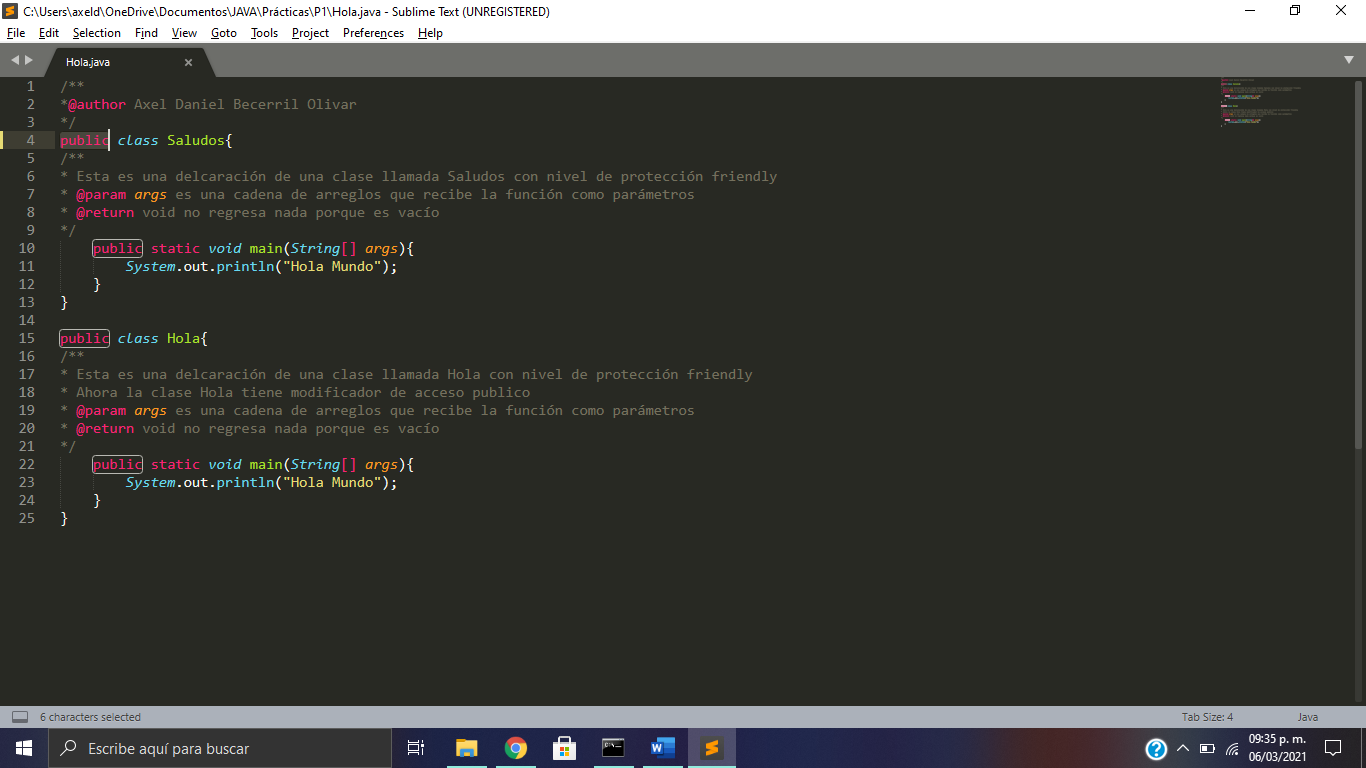


1. **Elimine de su carpeta de trabajo los archivos .class, modifique su archivo fuente agregando el modificador de acceso público a la definición de la clase Saludos. Compile de nuevo el archivo fuente.**

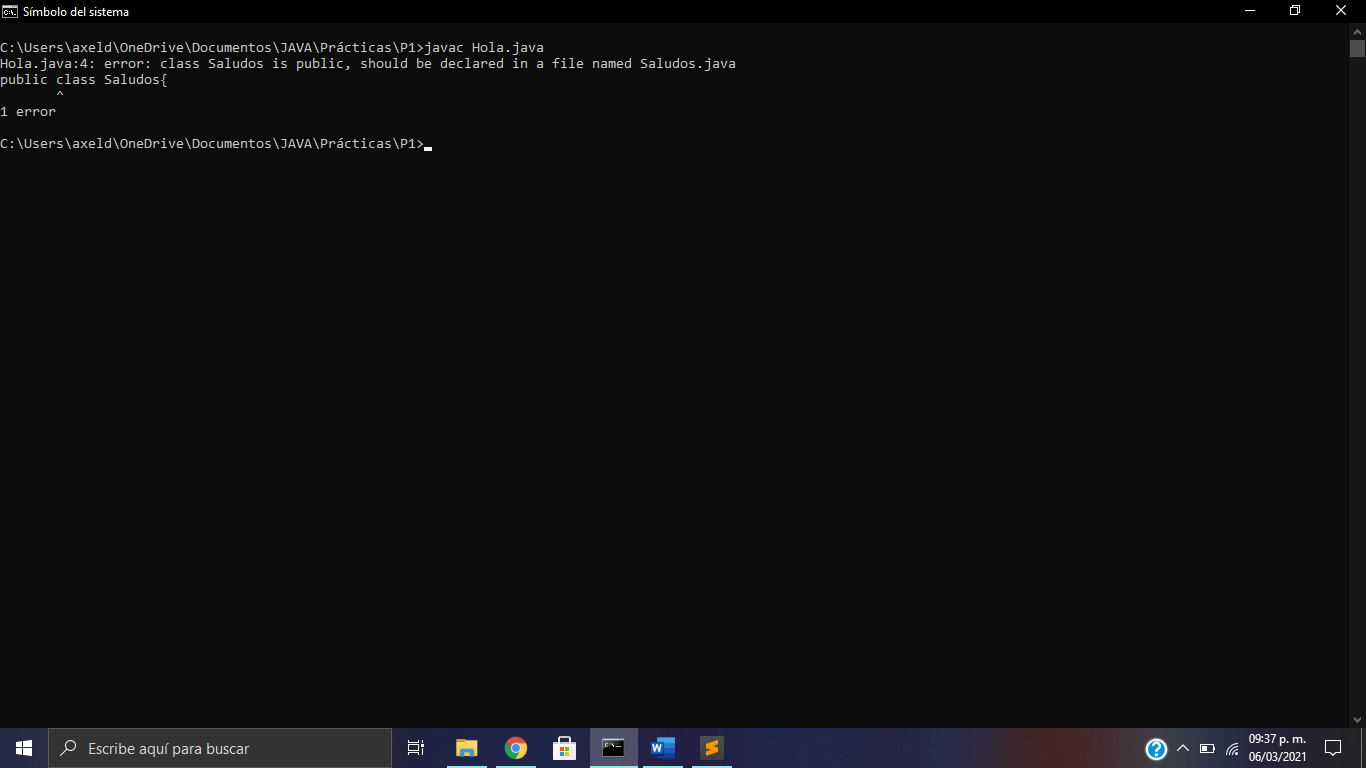
Se eliminaron los archivos .class



Ahora se modificó el nivel de protección a publico de la clase Saludos.

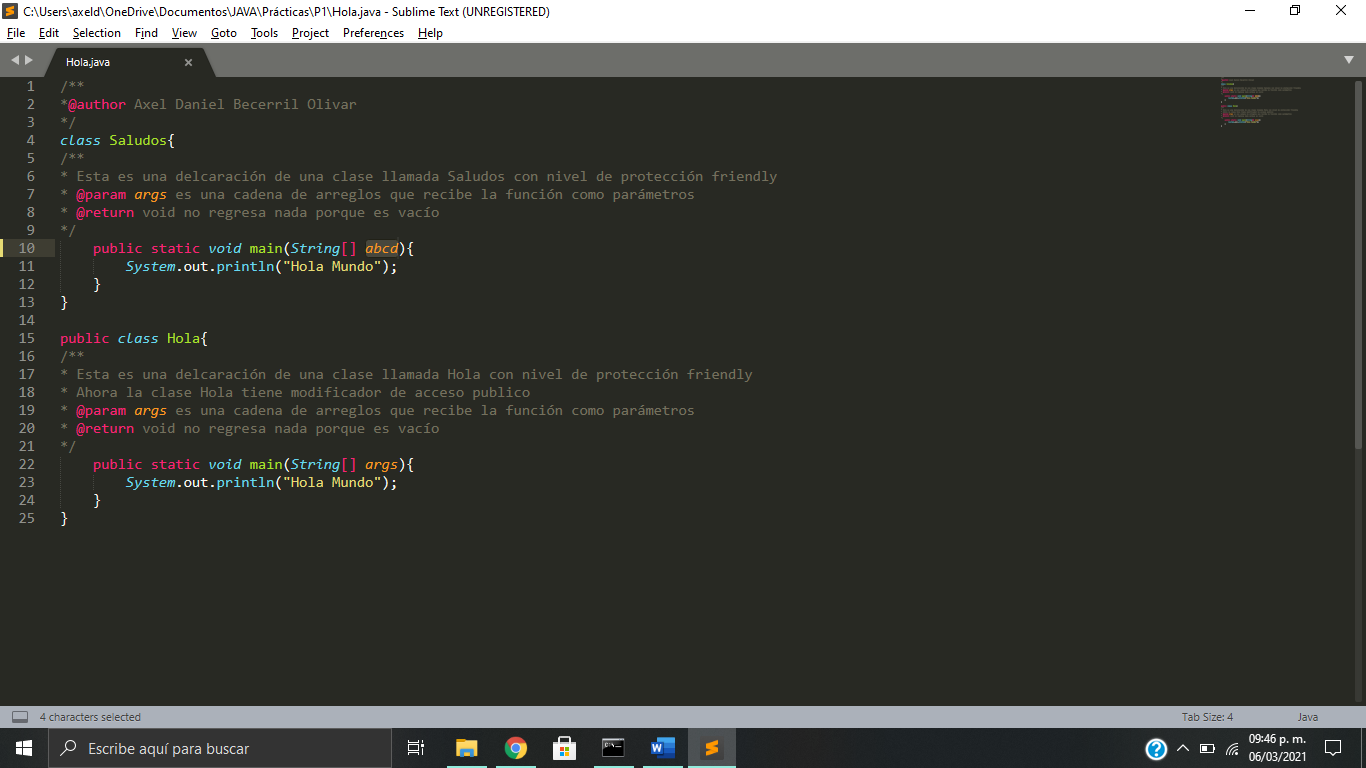


Al momento de compilar el archivo fuente aparece el siguiente error.

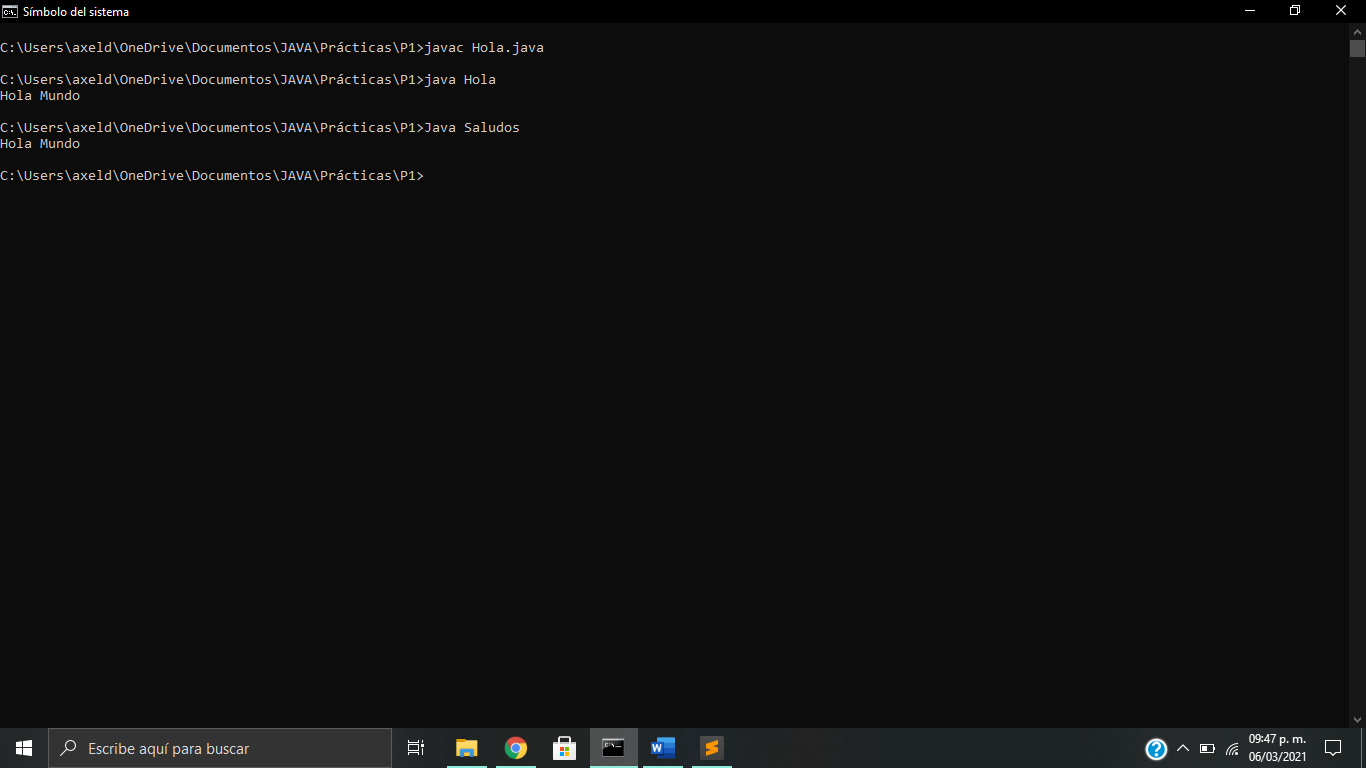


1. **Modifique su archivo fuente quitando el modificador de acceso público de la definición de la clase Saludos, cambie el nombre del parámetro args que se encuentra definido en el método main de cualquiera de las clases. Compile y ejecute las clases generadas.**

Se quitó el modificador de acceso público de la clase Saludos y el parámetro args.

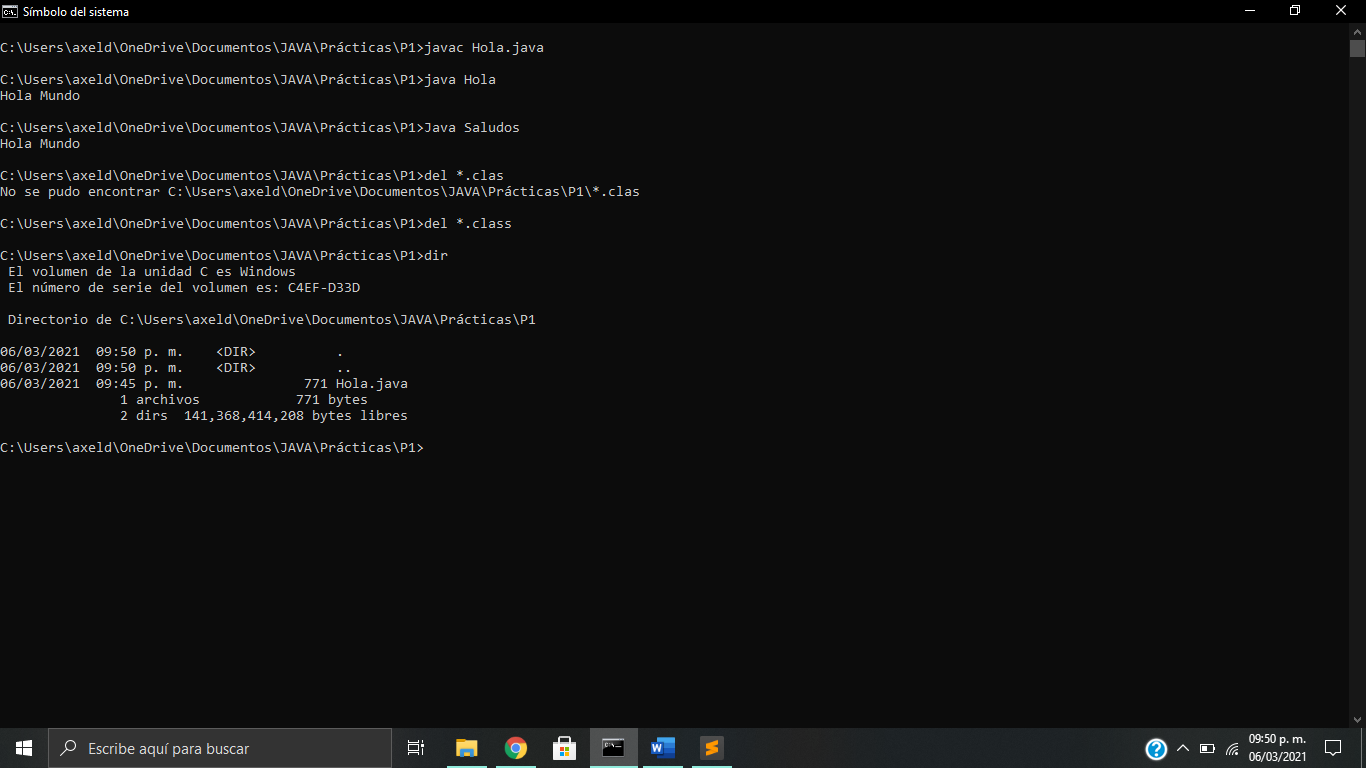


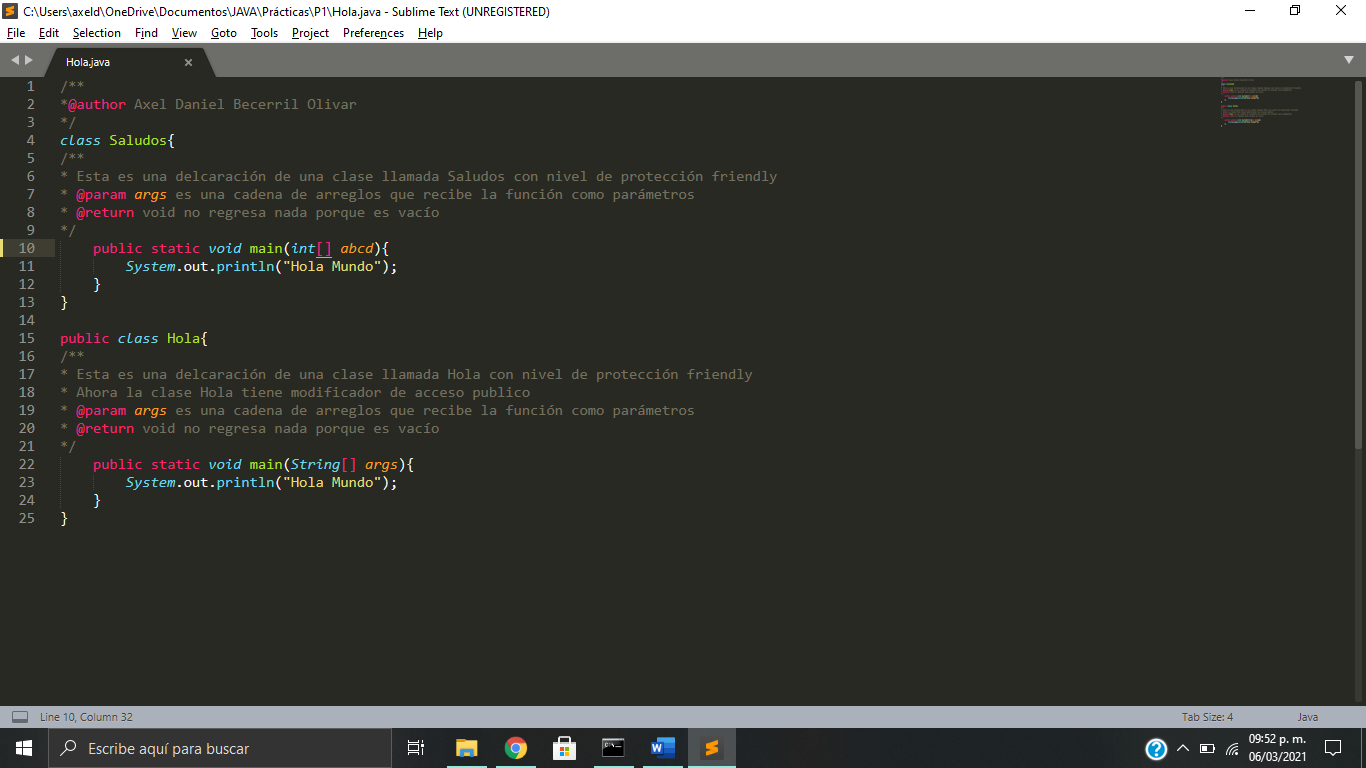
Se compiló y se ejecutaron las clases.

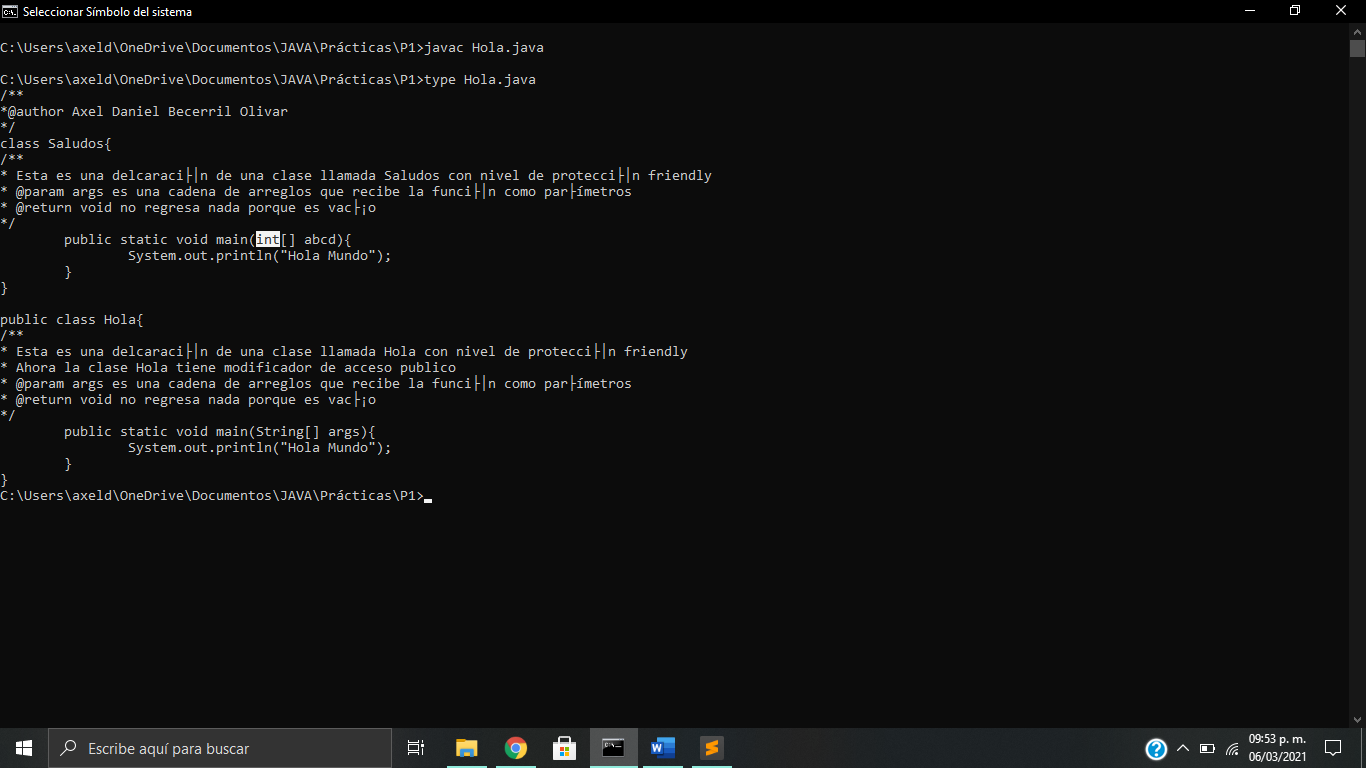


1. **Elimine de su carpeta de trabajo los archivos .class, modifique su archivo fuente cambiando el tipo de dato del parámetro args del método main de cualquiera de las clases definidas en su código fuente. Compile de nuevo su archivo fuente.**

Se eliminaron los archivos .class correctamente y se cambió el tipo de dato en el parámetro args.

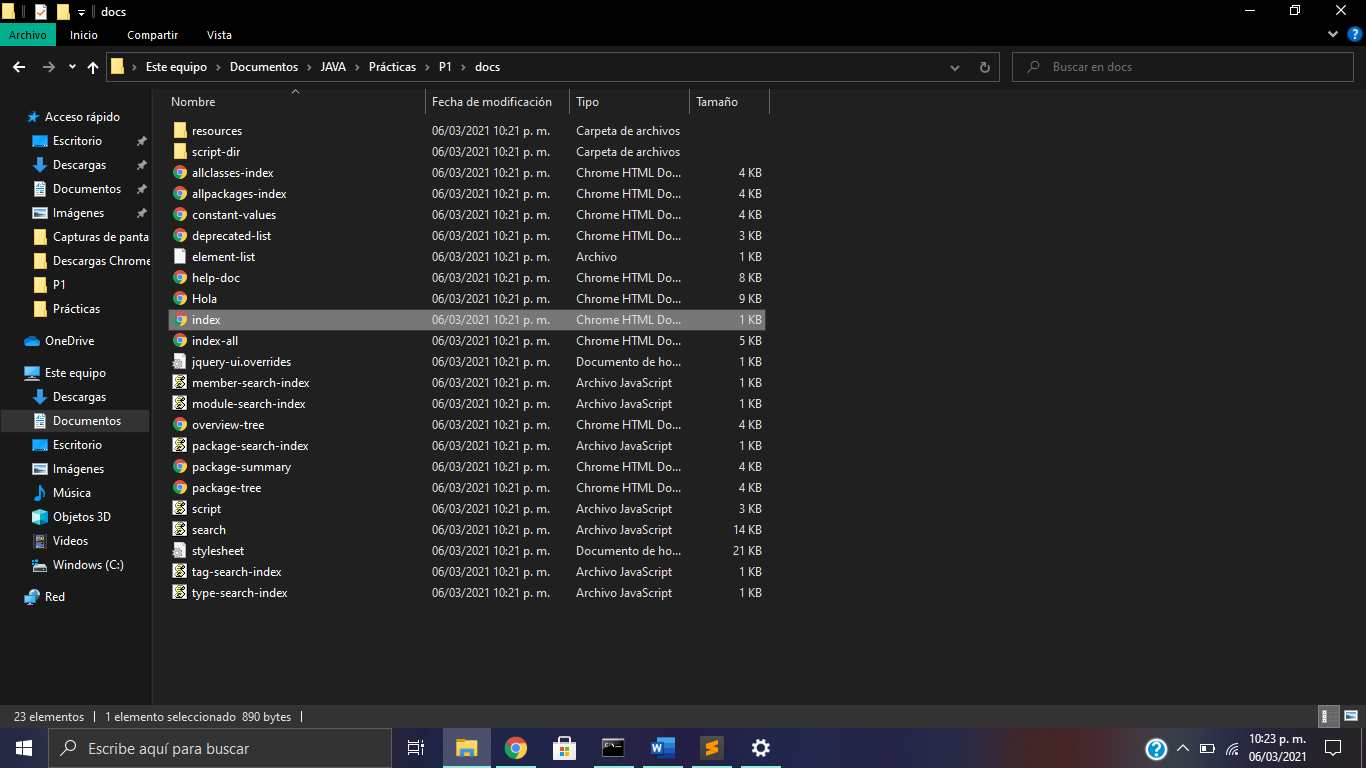
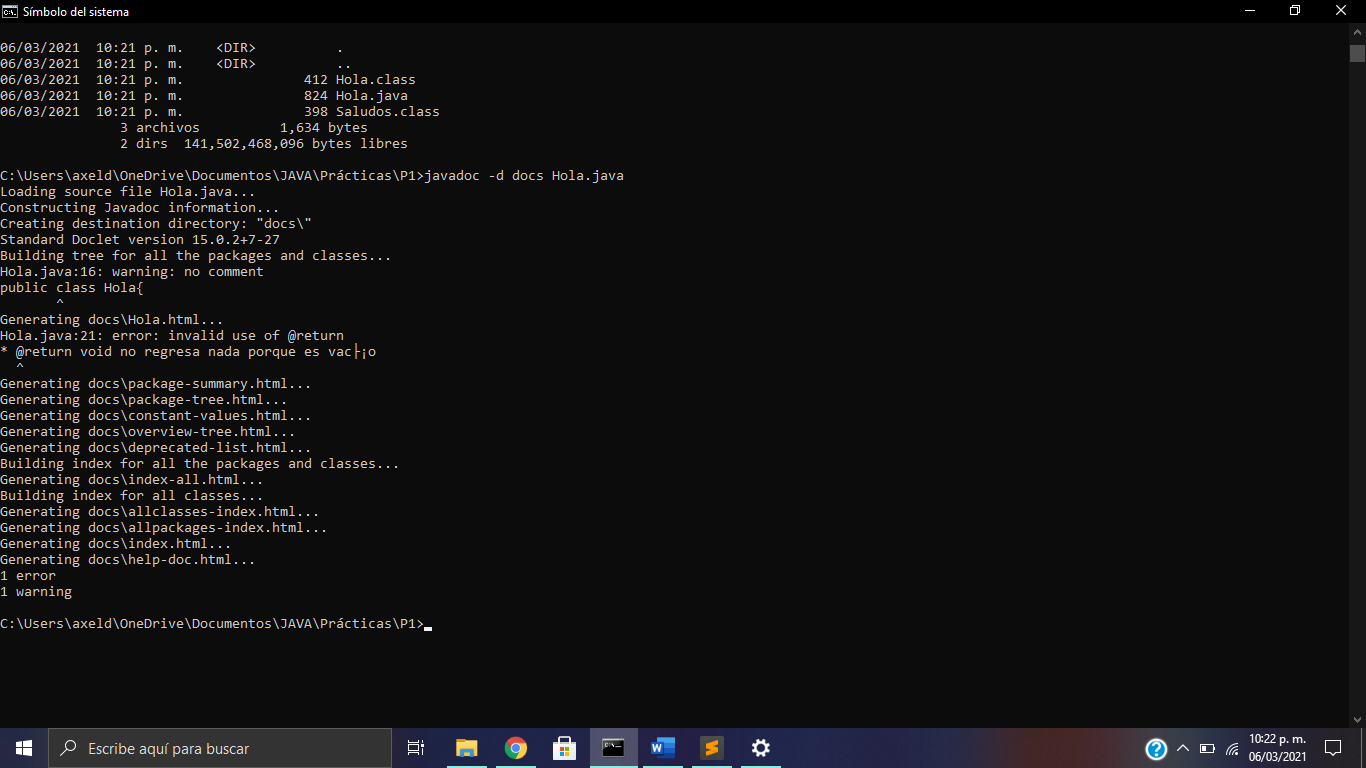


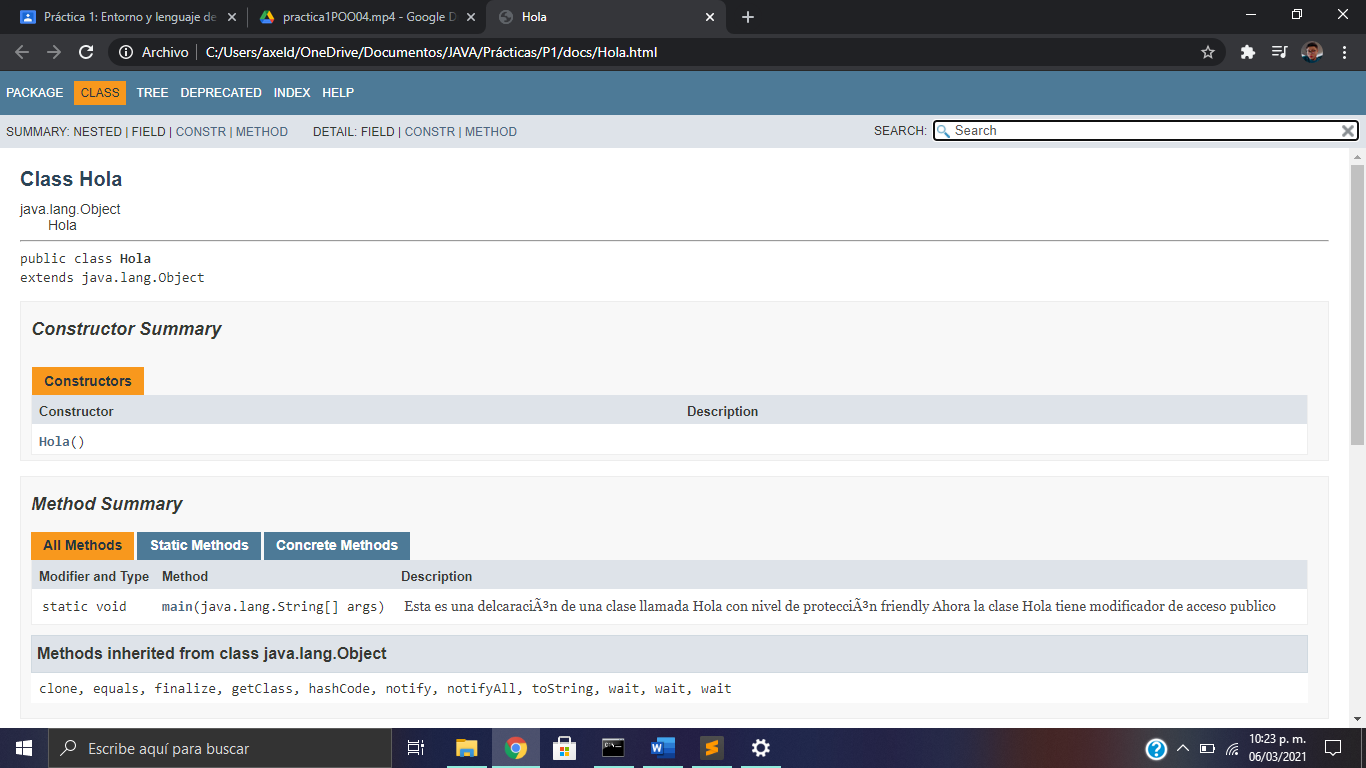




1. **Agregue comentarios de documentación a su código fuente y genere la documentación del mismo con el comando javadoc.**

Se creó la documentación.





1. **Diagramas UML**

**Icono

Descripción generada automáticamente**

**Conclusión**

Pude interactuar de manera exitosa con el lenguaje Java, aprendí los distintos errores que existen al momento de codificar e interactuar con el CMD, aprendí como crear una clase, de igual manera como hacer comentarios de documentación y su importancia.

**Referencias**

Martín, Antonio

**Programador Certificado Java 2.**

Segunda Edición.

México

Alfaomega Grupo Editor, 2008

Sierra Katy, Bates Bert

**SCJP Sun Certified Programmer for Java 6 Study Guide**

Mc Graw Hill

Dean John, Dean Raymond.

**Introducción a la programación con Java**

Primera Edición.

México

Mc Graw Hill, 2009

(s.f.). (C. Pes, Productor) Recuperado el 06 de 03 de 2021, de Abrirllave.com: https://www.abrirllave.com/java/palabras-clave.php

(s.f.).

*Ciberaula*. (s.f.). (Á. Roldán, Productor) Recuperado el 06 de 03 de 2021, de https://www.ciberaula.com/cursos/java/palabras\_reservadas\_java.php