

Nota =  $0.7 + 0.7 + 0.5 + 0.5 + 0.4 = 2.8 / 5 = 5.6$

En próximas entregas, intenta subir también los .java además del .pdf. Consigues algunas cosas en los ejercicios pero otras no, de todas formas, ánimo que con la práctica se mejora. Intenta los ejercicios de autoevaluación, te ayudarán.

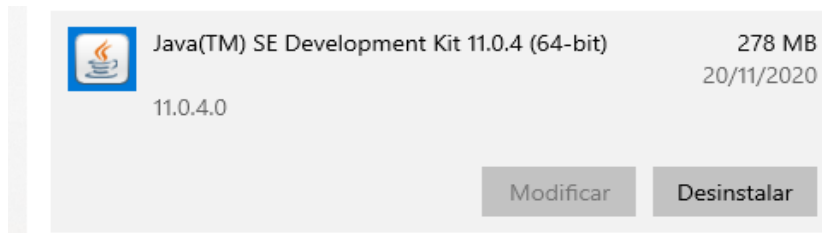
## PRÁCTICA UNIDAD 2: PRIMEROS PASOS EN JAVA.

### EJERCICIOS PADAWAN (OBLIGATORIOS)

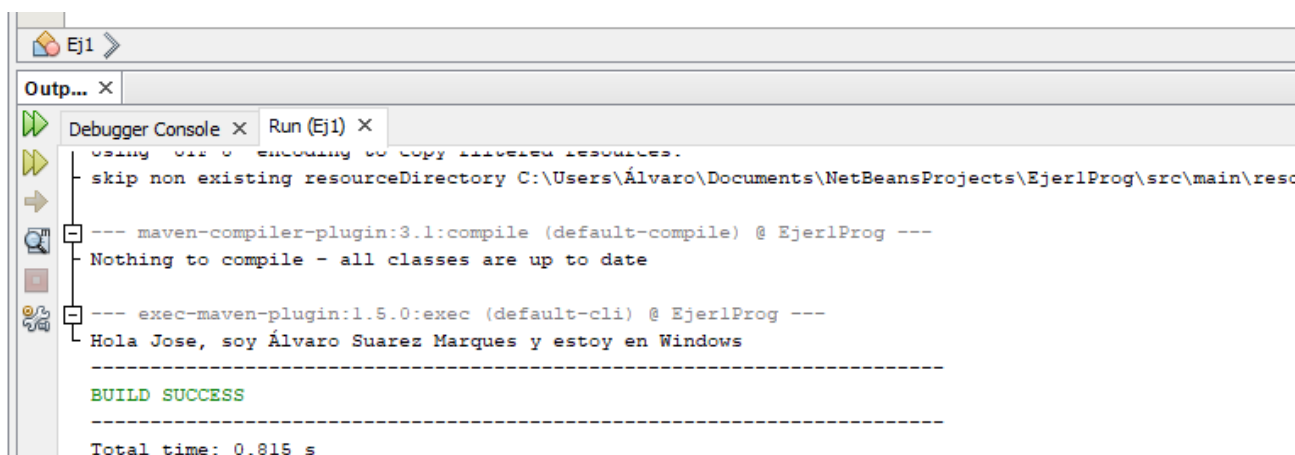
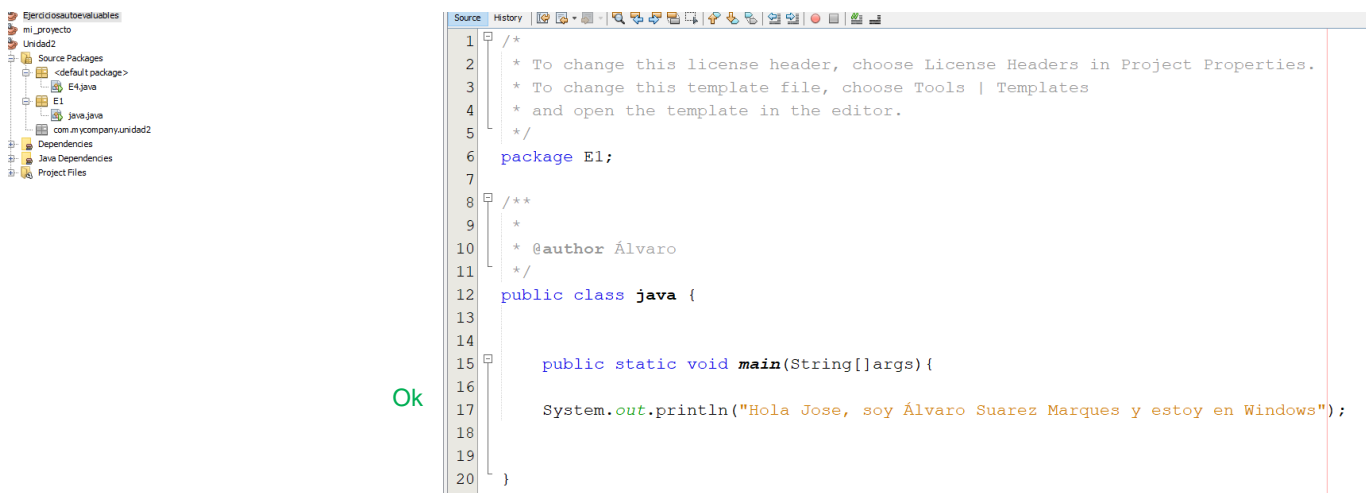
Álvaro Suárez Marqués

#### EJERCICIO 1. Crea/Usa una máquina virtual que tenga un SO Windows (Windows 7, 8, 10) de 64 bits.

Versión Java instalada: Te pedía capturas de definir las variables PATH, JAVA\_HOME y CLASSPATH



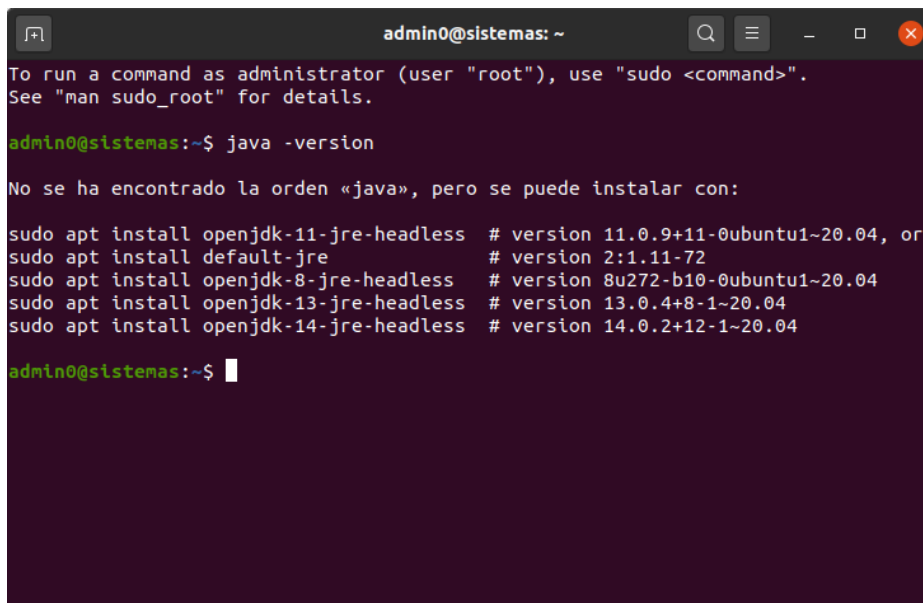
Programa creado:



## EJERCICIO 2. Crea/Usa una máquina virtual que tenga un SO Windows GNU/Linux de 64 bits (Ubuntu, Debian...).

El primer paso sería ir a la terminal y escribir el siguiente comando para ver la versión de java:

Java -version

A terminal window titled 'admin0@sistemas: ~' with search, menu, and window control icons. It displays the command 'java -version' and its output. The output indicates that the 'java' command is not found and lists available OpenJDK packages with their versions. The terminal text is as follows:

```
admin0@sistemas:~$ java -version

No se ha encontrado la orden «java», pero se puede instalar con:

sudo apt install openjdk-11-jre-headless # version 11.0.9+11-0ubuntu1~20.04, or
sudo apt install default-jre             # version 2:1.11-72
sudo apt install openjdk-8-jre-headless  # version 8u272-b10-0ubuntu1~20.04
sudo apt install openjdk-13-jre-headless # version 13.0.4+8-1~20.04
sudo apt install openjdk-14-jre-headless # version 14.0.2+12-1~20.04

admin0@sistemas:~$
```

Una vez vista la versión, instalaremos la que queramos con el comando:

Sudo apt install openjdk-12-jre-headless

```
admin0@sistemas: ~  
  
sudo apt install openjdk-11-jre-headless # version 11.0.9.1+1-0ubuntu1~20.04, o  
r  
sudo apt install default-jre # version 2:1.11-72  
sudo apt install openjdk-8-jre-headless # version 8u275-b01-0ubuntu1~20.04  
sudo apt install openjdk-13-jre-headless # version 13.0.4+8-1~20.04  
sudo apt install openjdk-14-jre-headless # version 14.0.2+12-1~20.04  
  
admin0@sistemas:~$ sudo apt install openjdk-11-jre-headless  
[sudo] contraseña para admin0:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
  ca-certificates-java java-common  
Paquetes sugeridos:  
  default-jre fonts-dejavu-extra fonts-ipafont-gothic fonts-ipafont-mincho  
  fonts-wqy-microhei | fonts-wqy-zenhei  
Se instalarán los siguientes paquetes NUEVOS:  
  ca-certificates-java java-common openjdk-11-jre-headless  
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 327 no actualizados.  
Se necesita descargar 37,8 MB de archivos.  
Se utilizarán 172 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n]
```

Una vez hecho esto ya tendremos instalado el jdk con la versión que hayamos elegido.

Programa realizado:

```
Ej1.java  
Abrir Guardar  
1 public class Ej1 {  
2     public static void main(String[] args) {  
3  
4         System.out.println("Hola Jose soy Alvaro Suarez Marques y estoy en linux");  
5  
6     }  
7  
8  
Java Anchura del tabulador: 8 Ln 8, Col 2 INS
```

Compilación del programa y ejecución de este en la terminal:

```
admin0@sistemas:~$ javac Ej1.java
admin0@sistemas:~$ java Ej1
Hola Jose soy Alvaro Suarez Marques y estoy en linux
admin0@sistemas:~$
```

Te ha vuelto a faltar pasarme capturas de pantalla de definir las variables PATH, JAVA\_HOME y CLASSPATH

## EJERCICIO 3. USO DE LA CLASE String Y OPERADOR TERNARIO. 0.5

```
import java.util.Scanner;
public class E3 {
    public static void main(String[] args) {
        String username, password, validaclave = "";

        Scanner teclado = new Scanner(System.in);

        System.out.println("Introduce tu nombre de usuario");
        username = teclado.nextLine();

        validaclave = (username.equalsIgnoreCase("admin")) ? "El usuario \"admin\" no se puede utilizar":"";
        System.out.println(validaclave);

        char primerChar = username.charAt(0);
        boolean numero;
        numero = (primerChar >= '0' && primerChar <= '9') ? true : false;
        /*if (primerChar >= '0' && primerChar <= '9')
            numero = true;
        */

        // Ok, también puedes escribir directamente numero, en vez de numero == true
        validaclave = (numero == true) ? "El nombre de usuario debe comenzar por letra":"";

        System.out.println("Introduce tu contraseña");
        password = teclado.nextLine();
        validaclave = (password.matches("[0-9]{8}")) ? "" : "Password debe tener un mínimo de 8 caracteres";
        // Pero esta expresión valida que sean 8 números, no 8 letras cualesquiera. Para la longitud podrías usar password.length()
        // validaclave = (password.matches()) ? "El usuario \"admin\" no se puede utilizar":""; ch >= 'a' && ch <= 'z'
        // para saber si hay minúsculas: "Hola" ---- paso a mayús. --> "HOLA" ¿iguales? no -> hay minúsculas
        // Para saber si hay mayúsculas: parecido...

        System.out.println(validaclave);

        // Para saber si tiene '@'/:... password.indexOf('@') >= 0 || ....
        // Otra posibilidad: passwords.contains(...)
```

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

```
public class E3 {
```

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

```
        String username, password, validaclave = "";
```

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

```
        System.out.println("Introduce tu nombre de usuario");
```

```
        username = teclado.nextLine();
```

```
validaclave = (username.equalsIgnoreCase("admin"))? "El usuario  
\"admin\" no se puede utilizar":"";
```

```
System.out.println(validaclave);
```

```
char primerChar = username.charAt(0);
```

```
boolean numero;
```

```
numero = (primerChar>= '0' && primerChar<='9') ? true : false;
```

```
/*if (primerChar>= '0' && primerChar<='9')
```

```
    numero = true;
```

```
*/
```

```
validaclave = (numero == true)? "El nombre de usuario debe  
comenzar por letra":"";
```

```
System.out.println("Introduce tu contraseña");
```

```
password= teclado.nextLine();
```

```
validaclave = (password.matches("[0-9]{8}"))? "":"Password debe  
tener un mínimo de 8 caracteres";
```

```
//validaclave = (password.matches())? "El usuario \"admin\" no se  
puede utilizar":"";
```

```
System.out.println(validaclave);
```

```
}
```

```
}
```

## **EJERCICIO 4. LA CLASE PRECONSTRUIDA Math Y FORMATEO DE TEXTO CON printf.**

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

No consigues cuadrar la tabla porque cuando el formato es importante, por ejemplo para cuadrar salida por pantalla, debes usar printf() u otros formateadores de datos, no println(). Además era precisamente lo que el ejercicio te pedía, que para formatear, utilizases printf() y así te acostumbres a usarlo.

```

public class E4 {
    public static void main(String[]args){
        double xc = 0, yc = 0, xp, yp;
        final double longitud_canion = 1; //1px=5m
        double alfa, velocidadInicial;

        //CORDENADAS PANTALLA
        double x0 = 0, y0 = 24, xf = 80, yf = 24; //valor en pixels 1 = 5

        Scanner teclado = new Scanner(System.in);
        System.out.print("Teclee angulo(grados): ");
        alfa= teclado.nextDouble();
        Te falta pasar el ángulo a radianes. Ten en cuenta que todas las operaciones trigonométricas en los ordenadores
        se hacen en radianes:  alfa *= 3.14 / 180;  // 2 PI radianes = 360 grados
        System.out.print("Teclee velocidad(m/s): ");
        velocidadInicial= teclado.nextDouble();

        xp= ((xc*5)+(longitud_canion*5))*Math.cos(alfa);
        yp= ((yc*5)+(longitud_canion*5))*Math.sin(alfa);
        System.out.println("Posición inicial del proyectil: "+(float)xp+",
        "+(float)yp );

        final double gravedad = 9.81;
        double ts = (velocidadInicial*Math.sin(alfa))/gravedad;
        double tv = 2*ts;

        System.out.println("Tiempo de subida: "+(float)ts+" y de vuelo:
        "+(float)tv);

        //final double velHorizontal = velocidadInicial*Math.cos(alfa);
        //double velVertical = velocidadInicial*Math.sin(alfa);
        double hmax = ((velocidadInicial*2)*(Math.sin(alfa))*2)/2*gravedad;
    }
}

```

```
System.out.println("Altura máxima: "+ (float)hmax);  
double dmax = ((velocidadInicial*2)*(Math.sin(alfa))*2)/gravedad;  
System.out.println("Distancia máxima: "+(float)dmax);
```

```
double t1 = 0.00;  
double x1 = 4.33;  
double y1 = 0.24;  
  
//System.out.println("    "+t1+" | (    "+x1*5+" , "+y1*5+" ) |    ("+x1+" ,  
"+y1+" )    |");  
double t25 = tv*0.25;  
double x25 = (dmax/2)/2;  
double y25 = hmax/2;  
  
//System.out.println("    "+t25+" | (    "+x25*5+" , "+y25*5+" ) |  
("+x25+" , "+y25+" )    |");  
  
//calcular la posicion cuando el tiempo esta en 25% del total  
double t50 = tv*0.50;  
double x50 = dmax/2;  
double y50 = hmax;  
  
//System.out.println("    "+t50+" | (    "+x50*5+" , "+y50*5+" ) |  
("+x50+" , "+y50+" )    |");  
double t75 = (float)tv*0.75;  
double x75 = (float)x50*25;  
double y75 = (float)hmax;  
  
//System.out.println("    "+t75+" | (    "+x75*5+" , "+y75*5+" ) |  
("+x75+" , "+y75+" )    |");  
double t100 = tv;  
double x100 = x50*2;  
double y100= 0.00;  
  
System.out.println("    "+t100+" | (    "+x100*5+" , "+y100*5+" ) |  
("+x100+" , "+y100+" )    |");
```

```

//TABLA

System.out.println("+-----+-----+-----+
-----+");

System.out.println("| tiempo | Posicion en el mundo | Posicion
en Pantalla |");

System.out.println("+-----+-----+-----+
-----+");

System.out.println("| (" + t1 + ") | (" + x1*5 + "), (" + y1*5 + ") | (" +
x1 + ", " + y1 + ")");

System.out.println("+-----+-----+-----+
-----+");

System.out.printf("| (" + " %.2d " + ") | (" + x25*5 + ", " + y25*5 + ")
| (" + x25 + ", " + y25 + ") |", t25 );

System.out.println("+-----+-----+-----+
-----+");

System.out.printf("| (" + " %.2d " + ") | (" + x50*5 + ", " + y50*5 + ") |
(" + x50 + ", " + y50 + ") |", t50);

System.out.println("+-----+-----+-----+
-----+");

System.out.println("| " + t75 + " | (" + x75*5 + ", " + y75*5 + ")
| (" + x75 + ", " + y75 + ") |");

System.out.println("+-----+-----+-----+
-----+");

System.out.println("| " + t100 + " | (" + x100*5 + ", " + y100*5 + ") |
(" + x100 + ", " + y100 + ") |");

System.out.println("+-----+-----+-----+
-----+");

}

}

```



# EJERCICIO JEDI

## EJERCICIO 7. CLASES PARA FECHAS.

Si miras las soluciones de los ejercicios de autoevaluación, verás varios ejercicios que trabajan con fechas y de los que podrías aprovechar código.

```
import java.util.Scanner;
import java.util.GregorianCalendar;
import java.text.SimpleDateFormat;

public class E7 {
    public static void main(String[] args) {
        int dia, mes, ano, diasNaturales;

        Scanner teclado= new Scanner(System.in);
        GregorianCalendar fecha=new java.util.GregorianCalendar();

        System.out.print("Día del mes: ");
        dia=teclado.nextInt();
        System.out.print("Mes: ");
        mes=teclado.nextInt();
        System.out.print("Año: ");
        ano=teclado.nextInt();
        System.out.print("Plazo(días naturales): ");
        diasNaturales=teclado.nextInt();

        dia= fecha.get(java.util.GregorianCalendar.DAY_OF_MONTH);
        mes= fecha.get(java.util.GregorianCalendar.MONTH)+1;
        ano= fecha.get(java.util.GregorianCalendar.YEAR);

        //int fechaEntrega = dia.add(fecha.DAY_OF_MONTH, diasNaturales);
        //fechaEntrega= fecha.get(java.util.GregorianCalendar.DAY_OF_MONTH);

        SimpleDateFormat sdf = new SimpleDateFormat("dd/MMMM/yyyy hh:mm:ss");
        //System.out.print("Tienes que entregarlo el: " + fechaEntrega);
```

Lees día, mes y año  
Lees plazo.  
Construyes la fecha a partir de (día, mes, año)  
Le sumas los días del plazo a la fecha.  
Miras si la fecha resultante es domingo (le restas 2 días)  
Si es sábado, le restas 1 día.  
Muestras la fecha.

```
// Calcula fecha de entrega
Date fechaEntrega = new Date( fecha.getTime() + diasNaturales * 24 * 60 * 60 * 1000 );
```

import java.util.Scanner; getTime() te devuelve un long (milisegundos desde 1/1/1900), si le sumas días en milisegundos (por eso \* 24 \* 60 \* ...) puedes hacer otra fecha....

import java.util.GregorianCalendar;

import java.text.SimpleDateFormat;

public class E7 {

public static void main(String[] args){

int dia, mes, ano, diasNaturales;

Scanner teclado= new Scanner(System.in);

```

GregorianCalendar fecha=new java.util.GregorianCalendar();

System.out.print("Día del mes: ");
dia=teclado.nextInt();
System.out.print("Mes: ");
mes=teclado.nextInt();
System.out.print("Año: ");
ano=teclado.nextInt();
System.out.print("Plazo(dias naturales): ");
diasNaturales=teclado.nextInt();

dia= fecha.get(java.util.GregorianCalendar.DAY_OF_MONTH);
mes= fecha.get(java.util.GregorianCalendar.MONTH)+1;
ano= fecha.get(java.util.GregorianCalendar.YEAR);

//int fechaEntrega = dia.add(fecha.DAY_OF_MONTH,
diasNaturales);

//fechaEntrega=
fecha.get(java.util.GregorianCalendar.DAY_OF_MONTH);

SimpleDateFormat sdf = new SimpleDateFormat("dd/MMMMM/yyyy
hh:mm:ss");

//System.out.print("Tienes que entregarlo el: " + fechaEntrega);

}

}

```

