

Práctica Entregable Primera Evaluación

El siguiente supuesto práctico entregable pondrá en la primera evaluación (bloque hasta Noviembre) en la parte práctica.

Para que se considere entregada tendrá que cumplir total o parcialmente con los requisitos indicados, siendo de forma parcial como mínimo, tener uno de los apartados correctos.

La detección de copia total o parcial de las prácticas puede llevar a la anulación de la entrega y, siendo una práctica obligatoria, el no alcanzar automáticamente el aprobado en la primera evaluación.

Requisitos:

La aplicación a entregar deberá funcionar como un solo programa que tenga un menú de opciones para la ejecución de diferentes subprogramas. Al ejecutarse deberá mostrarse un menú de opciones:

- Convertir numeración
- Jugar al ahorcado
- Calculadora
- Cifrado Cesar
- Salir

La opción de Salir se indicará para cerrar el programa (Sin System.exit).

Funcional:

En la práctica deberéis usar funciones y métodos así como paquetes y clases para ordenar el código.

Estructura:

La estructura del programa será la indicada a continuación:

```
✓ wirtz.practicas.practicaentregable
  ✓ ahorcado
    ✓ JuegoAhorcado.java
  ✓ calculadora
    ✓ Calculadora.java
  ✓ cesar
    ✓ CifradoCesar.java
  ✓ numeracion
    ✓ CambioNumeracion.java
  ✓ principal
    ✓ MenuPrincipal.java
  ✓ utilidades
    ✓ UtilCambioNumeracion.java
    ✓ UtilCifrado.java
    ✓ UtilObtencionDatos.java
  PracticaEntregable.java
```

UtilObtencionDatos tiene los métodos de E/S de información con el usuario. Puede ser útil crear un método para los menús que reciba el texto del menú y las opciones mínima y máxima dentro del menú.

UtilCambioNumeracion y UtilCifrado tendrán métodos de utilidad para dichos subprogramas.

1. Convertir numeración:

Al seleccionar esta opción se mostrará un menú:

- Convertir a binario
- Convertir a decimal
- Atrás
- Salir

La opción ‘Atrás’ volverá al menú principal.

La opción ‘Salir’ saldrá del programa (sin usar System.Exit)

La conversión a binario se realizará mediante divisiones.

La conversión a decimal se realizará mediante el teorema fundamental de la numeración.

2. Jugar al ahorcado:

Al seleccionar esta opción se mostrará un menú:

- Indicar intentos (defecto 3)
- Indicar palabras
 - Número de palabras.
 - ◆ Se obtendrán en bucle cada una de las palabras.
- Jugar
- Atrás
- Salir

La opción ‘Atrás’ volverá al menú principal.

La opción ‘Salir’ saldrá del programa (sin usar System.Exit)

En **indicar intentos** se solicita al usuario que introduzca cuantas oportunidades tiene antes de perder la partida.

En **indicar palabras** se solicita al usuario el número de palabras posibles que pueden salir al usuario. Tras indicarlo se le pedirán tantas palabras como indicara y se almacenarán para partidas.

En el caso de no indicar intentos ni palabras, el programa tendrá indicados valores por defecto en variables en código (3 intentos y tantas palabras como queráis).

En **Jugar** se iniciará la partida del ahorcado:

- Se mostrarán con _ los caracteres disponibles a introducir. (si, por ejemplo, la palabra es “alumna” se mostrará “_____”).
- Se pedirá al usuario que introduzca o bien un carácter o la palabra completa.
 - Si se introduce un carácter, se buscará en la palabra y se mostrarán las letras coincidentes (si, con la palabra anterior, se introduce la ‘a’ se mostrará “a _____ a” y se indicarán los intentos restantes).
 - Si se introduce una palabra, se comparará con la palabra a descifrar, pudiendo ganar la partida o que se reste un intento.
- La partida termina si bien se ha encontrado la palabra (no quedan caracteres por revelar o se ha indicado la palabra completa) o si bien no quedan más intentos restantes. En ambos casos deberá mostrarse el resultado.

En la configuración hay una colección de palabras posibles de las que se obtendrá la palabra a revelar. Para hacer esto, definid un array de cadenas que tenga un número de palabras que vosotros consideréis.

Para obtener una palabra de forma aleatoria sobre esta colección, podéis usar el método Math.Random de esta forma:

String palabraBuscar = palabras[(int)(Math.random() * palabras.length)];

Math.random() devuelve un valor entre 0 y 1 (incluye el 0 pero excluye el 1).

3. Calculadora:

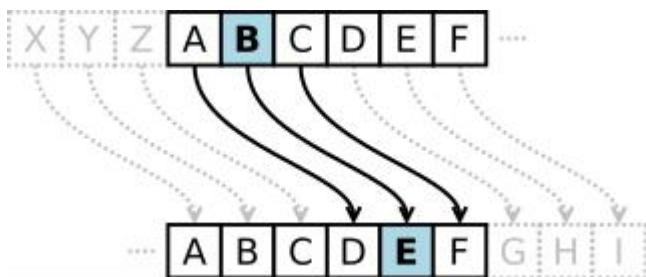
Al seleccionar esta opción se mostrará un menú:

- Sumar
- Restar
- Multiplicar
- Dividir Entero
- Dividir Real
- Módulo
- Atrás
- Salir

La obtención de datos podéis hacerla únicamente con valores enteros.

4. Cifrado César

El [cifrado César](#) aplica un desplazamiento dado N a una cadena de texto y utiliza esa nueva cadena.



Por ejemplo, con un desplazamiento de 3 espacios, si el texto fuese Hola, el cifrado sería : Krñd

En este subprograma deberéis mostrar el menú:

- Indicar desplazamiento (por defecto 1)
- Cifrar
- Descifrar
- Atrás
- Salir

La opción de **indicar desplazamiento** se usa para obtener del usuario cuantos caracteres debe desplazarse al realizar el cifrado (si no se indica se usará desplazamiento 1).

Las opciones de **cifrar** o **descifrar** preguntarán una cadena y aplicarán el desplazamiento para cifrar o descifrar la cadena dada, mostrando cual sería el mensaje cifrado o descifrado correspondientemente.

Para realizar el cifrado/descifrado, usad una cadena o array con el abecedario y usad el desplazamiento para moverlos sobre ella.

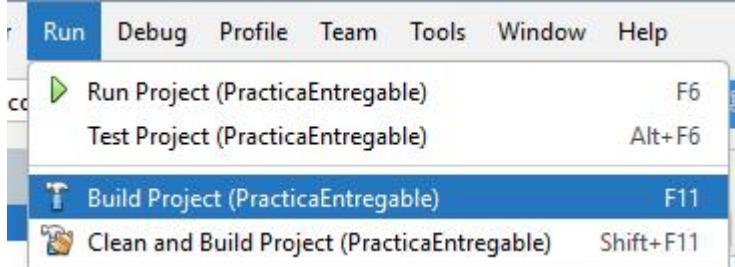
5. Creación de ficheros JAR

En Java, el ejecutable que podemos usar es un fichero comprimido con extensión .jar.
Para crear este archivo con Maven:

En pom.xml, dentro del tag <project>, después de </properties> definid este código (tendéis que cambiar los valores del mainClass por los adecuados a vuestro proyecto):

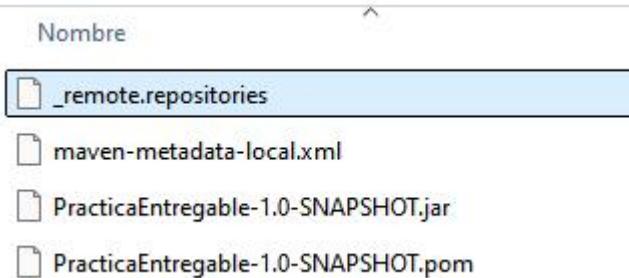
```
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-jar-plugin</artifactId>
            <version>2.4</version>
            <configuration>
                <archive>
                    <manifest>
                        <addClasspath>true</addClasspath>
                        <mainClass>wirtz.practicas.practicaentregable.PracticaEntregable</mainClass>
                    </manifest>
                </archive>
            </configuration>
        </plugin>
    </plugins>
</build>
```

Para generar un .jar:



Esto os generará un .jar en el directorio del proyecto (tenéis la ruta en la ventana Output después de la generación)

Entre los ficheros generados habrá un .jar



Podemos ejecutarlo mediante la consola:

```
> java -jar .\PracticaEntregable-1.0-SNAPSHOT.jar
```

```
Bienvenido a la práctica entregable 1  
Elige la opción
```

- 1. Convertir numeración
- 2. Jugar al ahorcado
- 3. Calculadora
- 4. Salir.

Al ejecutarlo desde la consola, si queremos que nos quede más la estructura de programa, podemos 'limpiar la consola' cuando queramos con este código:

```
public static void limpiarConsola() {  
    System.out.println("\033[H\033[2J");  
    System.out.flush();  
}
```

6. Entrega

La entrega de la práctica será de todo el código del proyecto (Podéis exportarlo a un zip mediante el menú File) y opcionalmente, en una carpeta aparte, el .jar generado.

Deberá crearse un comprimido de todo lo indicado con el siguiente nombre, excluyendo acentos:
Apellidos_Nombre_Ev1.rar (por ejemplo VazquezSanchez_Fernando_Ev1.rar).