

**Examen Final**  
**P.O.O.**  
**Ahorcado en Java**

**Fecha: 29/07/2021**

**Nombre: Christian Buestan**

## **CONSIDERACIONES**

1. La evaluación se realizará individual y se subirá al AVAC el día 2 de agosto hasta las 09h00.
2. En caso de copia, tendrán la calificación de cero.
3. Sólo se calificará si están implementando el patrón de diseño MVC

## **ENUNCIADO:**

Desarrollar un proyecto de Java en Netbeans que permita implementar un SISTEMA DE ADIVINANZA usando el patrón MVC.

### Clase vista:

1. Caja de texto donde el usuario ingresará el número de intentos, además del gráfico del ahorcado (1 pts).
2. Botón Visualizar. Mostrará en la tabla las palabras que se presentan en el fichero palabras.txt (10 palabras)
3. Botón Ocultar/Jugar. Mostrar en las celdas de la tabla sólo "x" y escogerá una palabra a jugar (1 pts)
4. Etiqueta Puntaje. Mostrará el número de aciertos. (1 pts)
5. Etiqueta Intentos. Mostrará el número de intentos. Tendrá un máximo de 4 intentos (1 pts)
6. Presentará una tabla y el nombre del jugador.

### Clase controlador de Adivinanza:

1. Permitirá mostrarPalabras.
2. Permitirá validarCoincidencias.
3. DibujarAhorcado
4. Etc.

### Clase lógica:

1. Leerá del fichero palabras.txt las palabras y lo mostrará en la tabla, al pulsar el botón Visualizar. (1 pts)
2. Validará que en la caja de texto no ingrese números, sino palabras. Mostrará un JShowMessageDialog con la excepción (1 pts)
3. En caso de no acertar con la palabra aumentar el ahorcado del gráfico.
4. Se procederá con el juego ingresando palabra a palabra al finalizar se guarda un archivo con el usuario, intentos, aciertos y las palabras adivinadas separadas por puntos y comas dentro del documento resultado.csv. (1 pts)
5. Cuando el número de intentos realizados sea igual al número de intentos propuestos, se mostrará un mensaje que ha finalizado, lo mismo pasaría si ya adivino todas. (1 pts)

### Clase Archivos:

6. Tiene el fichero palabras.txt que contiene un listado con las palabras a colocar. (1 pts)
7. Genera el fichero resultado.csv donde guardará. El formato será una palabra debajo de la otra palabra, el usuario y los datos de aciertos, intentos. (1 pts)

Subir el informe con demostraciones y capturas de pantallas de la ejecución y el código a su GIT personal.

## **Capturas de pantalla:**

[illegible]

Usuario

Intentos: 4

X

X

X

X

Palabras

Intentos: 4

X

X

X

X

Palabras

cazo

loba

malo

cado

saco

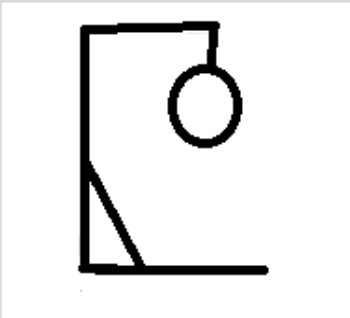
mina

finá

bola

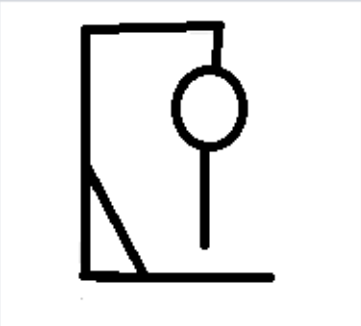
tapo

coca



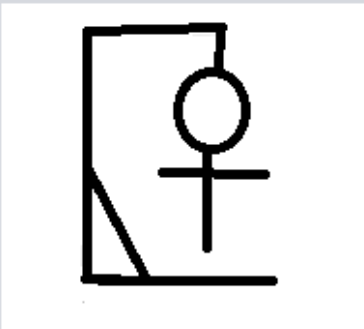
Intentos:3

Palabras
cazo
loba
malo
cado
saco
mina
finá
bola
tapo
coca



Intentos:2

Palabras
cazo
loba
malo
cado
saco
mina
finá
bola
tapo
coca



Intentos:2

Adivinar

X

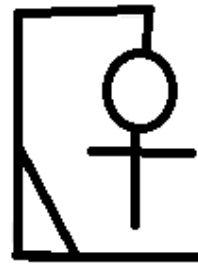
X

l

a

Palabras

cazo  
loba  
malo  
cado  
saco  
mina  
fina  
bola  
tapo  
coca



Intentos:1

Adivinar

b

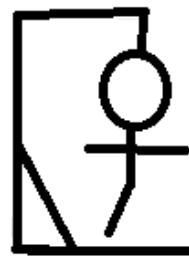
X

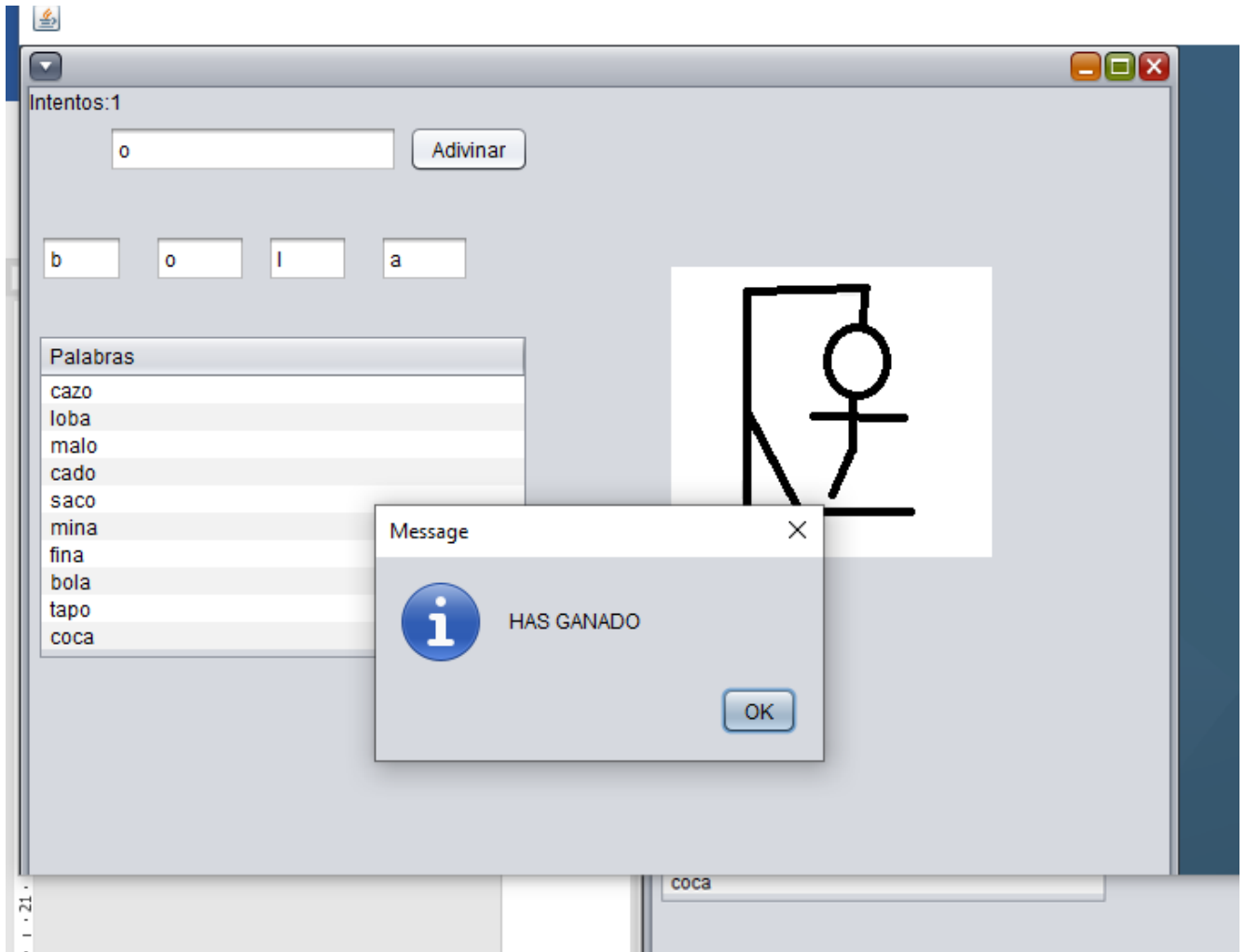
l

a

Palabras

cazo  
loba  
malo  
cado  
saco  
mina  
fina  
bola  
tapo  
coca





▼

Intentos:1

Jugar denuevo

o

Adivinar

b

o

l

a

Palabras

cazo

loba

malo

cado

saco

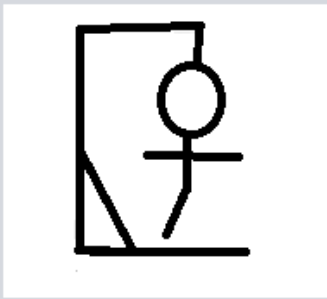
mina

finá

bola

tapo

coca



▼

Usuario

CD

Crear Usuario

Intentos:1

Jugar denuevo

o

Adivinar

b

o

l

a

Palabras

cazo

loba

malo

cado

saco

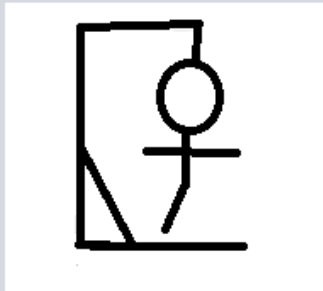
mina

finá

bola

tapo

coca



**Conclusiones:**

El presente proyecto tiene como objetivo ser interactivo con el usuario para que el mismo sea capaz de jugar cómodamente el clásico juego del ahorcado. También se usa archivos y excepciones para que el programa sea lo autosuficiente para identificar un problema con el ingreso de datos o los procesos de lógica interna.

Se espera que el presente trabajo sea reconocido como optimo y destacable para el docente encargado de la materia, muchas gracias.