

# Organización del Computador

## Trabajo Práctico

En este trabajo práctico vamos a implementar un juego de palabras donde un usuario tiene que ingresar una palabra y otro tiene que adivinarla. Al ingresar la palabra oculta se muestra un asterisco por cada letra. Luego el que adivina debe escribir una palabra con la misma cantidad de letras y dependiendo de las coincidencias con la palabra oculta se mostrará cuántas letras coinciden con la palabra original en la posición correcta (Bien), y cuántas letras coinciden pero están en una posición incorrecta (Regular). El que adivina tiene 10 intentos para encontrar la palabra oculta.

La siguiente es una muestra de la ejecución del programa, donde la palabra oculta es "PATO":

```
****
CASA Bien: 001 Regular: 000
COSA Bien: 000 Regular: 002
CASO Bien: 002 Regular: 000
MANO Bien: 002 Regular: 000
DADO Bien: 002 Regular: 000
SAPO Bien: 002 Regular: 001
PALO Bien: 003 Regular: 000
PATO Bien: 004 Regular: 000
Felicitaciones, ha ganado!
```

## Implementación

Una descripción en pseudocódigo del juego podría ser la siguiente

```
ingresar la palabra oculta
mientras BIEN es distinto de LONGITUD e INTENTOS > 0
    ingresar palabra
    marcar y contar letras BIEN y REGULAR
    escribir cantidades BIEN y REGULAR
    restar uno a INTENTOS
fin mientras
si BIEN es igual a LONGITUD
    mostrar mensaje Gano
si no
    mostrar mensaje Perdio
fin si
```

Se recomienda implementar los siguientes procedimientos:

INGRESAR_OCULTA	Recibe en DI el offset del espacio en memoria donde almacenar el texto ingresado. Permite ingresar caracteres hasta que el usuario presiona Enter (ASCII 13) o hasta que se llega a la longitud máxima (MAXLONG). Por cada caracter ingresado se debe mostrar un asterisco (*). Coloca un caracter \$ al final del texto.
-----------------	---

INGRESAR	Recibe en DI el offset del espacio en memoria donde almacenar la palabra adivinada, y en CL la cantidad de letras que tiene que tener. Permite ingresar caracteres (que se deben mostrar) hasta completar la cantidad de caracteres.
COPIAR	Recibe en SI el offset del texto a copiar y en DI el offset del espacio en memoria donde hay que copiarlo. Se copian todos los caracteres hasta encontrar el caracter \$ (que también se copia).
MARCAR	Hace una copia de la palabra oculta en AUX, llama a MARCAR_BIEN y a MARCAR_REGULAR y define las variables BIEN y REGULAR con los valores devueltos.
MARCAR_BIEN	Recibe en SI un puntero a una copia de la palabra oculta y en DI el puntero a la palabra adivinada. Recorre ambos textos contando las letras que coinciden exactamente y <b>reemplazándolas</b> en ambos textos por asteriscos (para no volver a contarlas). Devuelve en CL la cantidad de coincidencias encontradas.
MARCAR_REGULAR	Recibe en SI un puntero a la copia de la palabra oculta y en DI el puntero a la palabra adivinada, que ya fueron procesadas por MARCAR_BIEN. Para cada caracter de la palabra oculta que no es un asterisco, recorre la palabra adivinada hasta encontrar <b>una</b> coincidencia o hasta encontrar '\$'. Si encuentra una coincidencia reemplaza el caracter en por un asterisco para no volver a contarlo. Devuelve en CL la cantidad de coincidencias encontradas.
MOSTRAR_BYTE	Recibe en AL un número. Escribe en la pantalla los dígitos correspondientes a la representación decimal del número contenido en AL. Dividir el número por 100 para obtener las centenas, y dividir el resto por 10 para obtener las decenas y unidades. Luego escribir el caracter ASCII correspondiente a cada dígito decimal.

Se pueden usar las siguientes definiciones de datos

```

MAXLONG equ 20
OCULTA db MAXLONG+1 dup (0)
INTENTO db MAXLONG+1 dup (0)
AUX db MAXLONG+1 dup (0)
LONGITUD db 0
BIEN db 0
REGULAR db 0
INTENTOS db 10

```

## Entrega

La entrega debe incluir archivos con el **código fuente** del programa y un breve **informe impreso** con:

- Descripción de la implementación, problemas encontrados y decisiones tomadas.
- Código fuente.
- Ejecuciones del programa con distintas entradas mostrando las salidas obtenidas.
- Conclusiones.

El día de la entrega los integrantes de cada grupo deben presentar el informe impreso y mostrar el programa en funcionamiento. Los fuentes y el informe se deben enviar por mail a los docentes en un archivo adjunto.