

Final INFORMATICA I	Total Hojas	Duración	FINAL	Julio 2025
		19 a 20.30 hs	# 2	
Nombre y Apellido del alumno	Nº Legajo	Calificación	Docente a cargo / Firma	

Nota: Recuerde que leer y comprender el enunciado es parte del proceso de evaluación

Parte teórica

- 1) ¿Qué diferencias hay entre una biblioteca y un encabezado? ¿Qué información contiene cada uno?
- 2) Dibujar un diagrama de bloques que permita comprender la interacción entre un proceso cliente y un servidor. El diagrama debe poder describir el funcionamiento del sistema. Detalle la función o el comportamiento de cada bloque. **No es obligatorio explicitar el prototipo de cada función que se invoca. Si utilizó la biblioteca de la cátedra en su cursada, puede nombrar los bloques y explicar su funcionamiento.**

Parte práctica

IMPORTANTE: El enunciado está pensado para que lo puedan resolver utilizando estructuras autoreferenciadas. De forma que si no conoce en profundidad el tema o no se siente seguro al momento de la resolución, puede optar por utilizar un arreglo de memoria dinámica (memoria consecutiva), entiendo que al reducir la dificultad del examen no podrá aspirar a una nota mayor a 8 puntos. **Si no puede resolver alguna de las funciones del punto 1 o 2, en el punto 3 donde realiza el main, puede invocarlas como si un colega las hubiese desarrollado.**

Dado los siguientes tipos de datos:

```
typedef struct frecPalabras
{
    char palabra[30];
    int cantidad;
} PALABRAS;

typedef struct nodo
{
    PALABRAS miPalabra;
    struct nodo *sig;
} NODO;
```

- 1) Desarrollar una función que reciba una palabra, verifique si la misma se encuentra dentro de una pila. Si no se encuentra, **la debe agregar al principio**. Si se encuentra deberá incrementar la cantidad de apariciones de la palabra en una unidad. La función deberá contemplar que las palabras escritas en mayúscula o minúscula o cualquier variación de las mismas contabilizará en forma equivalente.

Debe cumplir con el siguiente prototipo:

int AgregarEnPila(NODO **pila, char * palabra);

- 2) Desarrollar una función que recorra una **pila** y busque la palabra que mayor repetición tiene. Si hay dos o más palabras que contengan la misma puntuación, deberá informar la primera palabra. La función **devuelve 0 si no hay palabras o distinto de cero si encontró al menos una palabra**. La función obtendrá por referencia la palabra en cuestión y su cantidad de repeticiones.

Debe cumplir con el siguiente prototipo:

int BuscarPalabraMax(NODO **pila, char * palabraObtenida, int *cantidadRepeticiones);

- 3) Desarrollar un programa que reciba por argumentos del main una serie de palabras donde la separación válida de las mismas es el espacio. A partir de ese conjunto de strings deberá cargar en una pila las palabras (**Función del punto (1)**), luego deberá mostrar la palabra que más se repitió (**Función del punto (2)**). Por último, deberá liberar todos los recursos utilizados en forma adecuada. Si lo requiere, puede desarrollar otras funciones complementarias quedando su definición a su cargo.

Ejemplo de uso: ./app auto perro casa Auto