



## Algoritmos y Estructura de Datos y Algoritmos\ Primera Práctica Calificada 1

**Profesor:** Laura Elizabeth Florian Cruz

Nombre: \_\_\_\_\_

Código: \_\_\_\_\_

Duración: 110 min

### Instrucciones:

- No se permite el uso de materiales. No se podrá utilizar libros, ni apuntes de clase, ni dispositivos electrónicos.
- Resuelva la pregunta haciendo uso del computador y envíe la solución al mail
- indicado por el profesor.
- El tiempo de los algoritmos influirá en el puntaje de cada pregunta.
- Todas las soluciones deben contar con menú o código de comprobación.
- Código que no compila no será considerado en su totalidad.
- Soluciones misteriosas o supuestas invalidan la solución.
- La solución debe ser entregada al profesor eliminando los archivos \*.sdf y carpetas Debug
- La solución debe ser empaquetada y enviada al correo: [pcsilflo@upc.edu.pe](mailto:pcsilflo@upc.edu.pe) con el apellido y código del alumno. Ejemplo: florianU30041985\_pc1

### Pregunta 1 (10 Puntos)

Implementar una aplicación (consola) que permita guardar, mostrar, modificar información de partidos políticos para las elecciones presidenciales. Cada organización política tiene un candidato presidencial (nombre, DNI, dirección, fechaNacimiento) y una lista de congresistas. El programa debe guardar los registros, leerlos y modificarlos.

A continuación se define las siguientes funcionalidades esperadas:

- **(1 Puntos)** Definición del Tipo de Dato Abstracto OrganizacionPolitica (usando clases y relaciones entre clases.
- **(1 Puntos)** Crear un menú de opciones
- **(2 Puntos)** Implementar los métodos para agregar organizaciones cada uno con su candidato y congresistas, modificarlos, eliminarlos y el menú de opciones.
- **(2 Puntos)** Tener como opción el ingreso de la información completa de una organización política y guardarlo en un archivo JNE\_elecciones2016.txt
- **(2 Puntos)** Adicionar la opción leer la información de organizacionesPoliticas desde archivo al
- **(2 Puntos)** La opción de modificar el archivo será para aclarar la información del candidato presidencial

### Pregunta 2 (4 Puntos)

Implemente una función con templates que permita ordenar los elementos de un arreglo de manera descendiente.

### Pregunta 3 (3 Puntos)

Implementar un algoritmo que lea una cadena y extraiga las vocales que contengan (colocándola en otra cadena) y las muestre a la inversa.

### Pregunta 4 (1.5 Puntos) (Creen Un Archivo En Notepad Con La Respuesta)

Calcular el tiempo detallado y la notación Big O de la siguiente función que recibe como parámetro un vector de enteros y el número de elementos del vector:

```
int fun(int n)
{
    int count = 0;
    for (int i = n; i > 0; i /= 2)
        for (int j = 0; j < i; j++)
            count += 1;
    return count;
}
```

Tiempo detallado:

Tiempo asintótico (BigO):

**Pregunta 5 (1.5 Puntos)** (Creen Un Archivo En Notepad Con La Respuesta)

Indique solamente su tiempo asintótico (big O)

*int funcion1 (int n)*

```
{  
    int i, j, k, p, q = 0;  
    for (i = 1; i < n; ++i)  
    {  
        p = 0;  
        for (j = n; j >= 1; j = j/2)  
            ++p;  
        for (k = 1; k <= p; k = k*2)  
            ++q;  
    }  
    return q;  
}
```

Tiempo asintótico (BigO):

## Rúbrica (FALTA AGREGAR LO DE ARREGLOS)

Crecimiento de funciones			
Tiempo constante o lineal	Evalúa e identifica el tiempo de una función que contiene estructuras repetitivas anidadas e identifica número de iteraciones constantes	No identifica repetitivas con número de iteraciones constantes	No identifica el tiempo en ningún caso.
	2	1	0
Tiempo lineal o exponencial	Evalúa e identifica el tiempo de una función que contiene estructuras repetitivas anidadas e identifica número de iteraciones constantes	No identifica correctamente algunos componentes del algoritmo	No identifica el tiempo en ningún caso.
	2	1	0
Tiempo logarítmico	Evalúa e identifica el número de iteraciones de las repetitivas y asocia el comportamiento con crecimiento logarítmico.	No identifica correctamente el tiempo logarítmico pero identifica los demás componentes correctamente.	No identifica el tiempo en ningún caso.
	2	1	0
Tipos de datos abstractos, archivos y programación en general			
Clases principales	Implementa la clase principal de la aplicación usando principios de Encapsulamiento	No usa encapsulamiento correctamente	No implementa clase principal
	1	0.5	0
Clases gestoras o controladoras	Identifica, diseña e implementa una clase gestora para manejar una colección de objetos de tipo clase Principal	No hace uso de colecciones	No implementa clase gestora
	2	1	0
Operaciones	Implementa operaciones de registrar, listar y eliminar como parte de la clase gestora, estas operaciones no necesariamente deben manejar los datos directamente de archivos.	No implementa dichas operaciones como parte de la clase gestora o controladora	No implementa las operaciones

	2	1	0
Lectura de archivos	La aplicación abre un archivo al iniciar la aplicación, lee todos los datos en una colección de objetos dentro de clase gestora y cierra el archivo.	Lee los archivos de manera parcial o con errores	No lee archivos
	2	1	0
Escritura de archivos	La aplicación abre un archivo antes de terminar y escribe todos los datos de la colección de objetos al mismo, luego cierra el archivo.	Escribe archivos de manera parcial o con errores	No escribe archivos
	2	1	0
Interfaz de usuario/casos de prueba	Implementa una interfaz de usuario en modo consola con menú de opciones o casos de prueba simple probando todas las operaciones principales del programa.	Implementa una interfaz incompleta o sus casos de prueba no cubren todas las operaciones	No implementa
	1	0.5	0