



## **ESTRUCTURAS DE DATOS Y ALGORITMOS**

**2º de Grado en Ingeniería Informática**

**(Curso 2020-2021)**

### **PRÁCTICA 2**

#### **1. NORMAS GENERALES**

1. El código deberá mostrar modularidad, calidad, legibilidad y el uso apropiado de estructuras y la inclusión de comentarios. En esta práctica deberá codificarse tanto en C como en C#.
2. El alumno deberá subir a la plataforma virtual la práctica realizada. La subida al espacio habilitado podrá realizarse hasta el 02/12/2020 en convocatoria ordinaria y hasta el 13/06/2021 en convocatoria extraordinaria, ambas fechas inclusive.
3. En las Partes 1 y 3 podrá utilizarse, si el alumno lo desea, la plantilla proporcionada. Para las partes 2, 4 y 5 no se proporciona plantilla.
4. Se deberá subir únicamente un fichero \*.rar, nombrado del siguiente modo: PR2\_NombreApellido.rar (Ejemplo: PR2\_AntonioPerez.rar). El nombre y el apellido del alumno no deberá contener acentos.
5. El fichero \*.rar contendrá:
  - En el caso de C únicamente los ficheros fuente, nombrados siguiente modo: PR2\_P1\_NombreApellido.c, PR2\_P2\_NombreApellido.c (Ejemplo: PR2\_P1\_AntonioPerez.c, PR2\_P2\_AntonioPerez.c). El nombre y apellido del alumno, no deberán contener acentos.
  - En el caso C# todo el proyecto de Visual Studio.
6. Aquellos ejercicios cuyos ficheros de entrega no cumplan la normativa de nombrado se calificarán con 0 puntos.
7. El ejercicio deberá realizarse individualmente.

## 2. OBJETIVOS

- El objetivo es familiarizarse con algunas estructuras de datos: conjuntos, pilas, listas y colas.

## 3. ENUNCIADO

### 3.1. Primera parte: Conjuntos (en C)

Debe desarrollarse un programa en C en el que dados dos archivos Archivo1, Archivo2 que contienen palabras separadas por un blanco o fin de línea, se creen dos conjuntos con las palabras de Archivo1 y Archivo2, y se realicen determinadas operaciones con ellos.

Diseñar y codificar en C un programa que muestre por la salida estándar un menú con las siguientes opciones:

#### **MENU**

1. Mostrar conjuntos 1 y 2.
2. Unir conjuntos 1 y 2.
3. Intersección de conjuntos 1 y 2.
4. Diferencia de conjuntos 1 y 2.
5. Salir del programa.

### 3.2. Segunda parte: Conjuntos (en C#)

Debe desarrollarse un programa en C# que: introduzca en el conjunto 1 los números pares comprendidos entre 1 y 10, y en el conjunto 2 los números entre 1 y 10 múltiplos de cuatro. Debe utilizarse obligatoriamente la librería System.Collections así como la clase HashSet. El programa presentará el siguiente menú de opciones:

#### **MENU**

1. Mostrar conjuntos 1 y 2.
2. Unir conjuntos 1 y 2.
3. Intersección de conjuntos 1 y 2.
4. Diferencia de conjuntos 1 y 2.
5. Salir del programa.

Los resultados de las opciones 1, 2, 3 y 4 se mostrarán por pantalla. La opción 5 mostrará la palabra “Adios” y el programa terminará.

### 3.3. Tercera parte: Pilas y Colas (en C)

Diseñar y codificar en C un programa que muestre por la salida estándar un menú con las siguientes opciones:

#### MENU

1. Gestionar pilas.
2. Mostrar pilas.
3. Gestionar colas.
4. Mostrar colas.
5. Salir del programa.

Se describe a continuación la funcionalidad de cada una de estas opciones:

#### 1. Gestionar pilas

Escribir un programa con el que se trabaje con un total de 4 pilas ( $n=4$ ). Se solicitará al usuario dos enteros  $i, j$ , tal que  $1 \leq \text{abs}(i) \leq n$ . De tal forma que el criterio de selección y llenado de las pilas es el siguiente:

- Si  $i > 0$  se colocará el elemento  $j$  en la pila  $p_i$ .
- Si  $i < 0$  se eliminará un elemento de la pila  $p_i$ .
- Si  $i = 0$  no se hace nada.

Debe respetarse el mecanismo de gestión de pilas (accediendo por la cima).

#### 2. Mostrar pilas

Esta opción muestra en pantalla el contenido de las pilas.

#### 3. Gestionar colas

Escribir un programa con el que se trabaje con un total de 4 colas ( $n=4$ ). Se solicitará al usuario dos enteros  $i, j$ , tal que  $1 \leq \text{abs}(i) \leq n$ . De tal forma que el criterio de selección y llenado de las colas es el siguiente:

- Si  $i > 0$  se colocará el elemento  $j$  en la cola  $c_i$ .
- Si  $i < 0$  se eliminará un elemento de la cola  $c_i$ .

Debe respetarse el mecanismo de gestión de colas (accediendo por la cola).

#### 4. Mostrar colas

Esta opción muestra en pantalla el contenido de las colas.

## 5. Salir del programa

Escribe “Adios” y el programa termina.

Ejemplo de ejecución:

```
*****
* Opciones:                                *
* 1. Gestionar pilas                       *
* 2. Visualizar pilas                     *
* 3. Gestionar colas                      *
* 4. Mostrar colas                        *
* 5. Salir                                *
*****
Elija opcion:
```

Elegimos la opción 1:

```
*****
* Opciones:                                *
* 1. Gestionar pilas                       *
* 2. Visualizar pilas                     *
* 3. Gestionar colas                      *
* 4. Mostrar colas                        *
* 5. Salir                                *
*****
Elija opcion:1
Número entero i: 1
Número entero j: 2
```

Introducimos el número 2 en la pila 1.

Elegimos la opción 2:

```
*****
* Opciones:                                *
* 1. Gestionar pilas                       *
* 2. Visualizar pilas                     *
* 3. Gestionar colas                      *
* 4. Mostrar colas                        *
* 5. Salir                                *
*****
Elija opcion:2
Pila 1
    Numero: 2
Pila 2
Pila 3
Pila 4
```

Elegimos la opción 1 e introducimos el número 5 en la pila 1.

```

Elija opcion:1
Número entero i: 1
Número entero j: 5
*****
* Opciones:                                *
* 1. Gestionar pilas                       *
* 2. Visualizar pilas                     *
* 3. Gestionar colas                      *
* 4. Mostrar colas                       *
* 5. Salir                               *
*****
Elija opcion:

```

Elegimos la opción 2:

```

*****
* Opciones:                                *
* 1. Gestionar pilas                       *
* 2. Visualizar pilas                     *
* 3. Gestionar colas                      *
* 4. Mostrar colas                       *
* 5. Salir                               *
*****
Elija opcion:2
Pila 1
    Numero: 5
    Numero: 2
Pila 2
Pila 3
Pila 4

```

Ahora elegimos la opción 1, quitamos el número 2 de la pila 1:

```

*****
* Opciones:                                *
* 1. Gestionar pilas                       *
* 2. Visualizar pilas                     *
* 3. Gestionar colas                      *
* 4. Mostrar colas                       *
* 5. Salir                               *
*****
Elija opcion:1
Número entero i: -1
Número entero j: 2

```

Elegimos la opción 2:

```
*****
* Opciones:                                *
* 1. Gestionar pilas                       *
* 2. Visualizar pilas                     *
* 3. Gestionar colas                      *
* 4. Mostrar colas                       *
* 5. Salir                                *
*****
Elija opcion:2
Pila 1
    Numero: 5
Pila 2
Pila 3
Pila 4
```

### 3.4. Cuarta parte: Pilas y Colas (en C#)

Mismo enunciado que el apartado 3.3 , pero en C#. Debe utilizarse obligatoriamente la librería System.Collections así como las clases Stack y Queue.

### 3.5. Quinta parte: listas enlazadas (en C)

Un polinomio puede ser descrito como una lista enlazada. El primer nodo de la lista simboliza el primer término, el segundo el segundo término y así de modo sucesivo. Cada nodo de la lista enlazada contiene el coeficiente y el exponente del término.

Diseñar y codificar en C un programa que muestre por la salida estándar un menú con las siguientes opciones:

#### **MENU**

1. Evaluar polinomio
2. Sumar dos polinomios
3. Obtener polinomio derivada.
4. Salir del programa.

Se describe a continuación la funcionalidad de cada una de estas opciones:

#### **1. Evaluar polinomio**

Esta opción, para un polinomio p1, solicitará el coeficiente y el exponente del término al usuario y lo introducirá en una lista enlazada. El coeficiente y el exponente serán números enteros. Se pedirá al usuario un valor entero x, y calculará para él el valor del polinomio. Se mostrarán en pantalla: los términos del polinomio y el valor obtenido para x.

#### **2. Sumar dos polinomios**

Esta opción, para cada polinomio  $p_1$  y  $p_2$ , solicitará el coeficiente y el exponente del término al usuario y lo introducirá en una lista enlazada. El coeficiente y el exponente serán números enteros. Se pedirá al usuario un valor de  $x$ , y calculará para él el valor del polinomio suma. Se mostrarán en pantalla: los términos del polinomio suma y el valor obtenido para  $x$ .

### **3. Obtener polinomio derivada**

Esta opción, para un polinomio  $p_1$ , solicitará el coeficiente y el exponente del término al usuario y lo introducirá en una lista enlazada. Se pedirá al usuario un valor de  $x$ , y calculará para él el valor del polinomio derivada. Se mostrarán en pantalla: los términos del polinomio derivada y el valor obtenido para  $x$ .

### **4. Salir del programa**

Escribe “Adios” y el programa termina.