Parte A

**4.- Resuelva la siguiente tarea en al menos dos paradigmas de programación (SOLO diseño). No implemente la solución. Dado un ordenamiento de piezas de dominó, determine si es un ordenamiento válido de manera que retorne verdadero o falso, según corresponda.**

**Basado en el paradigma “Orientado a Objetos”:**

Entrada: Listado de piezas de dominó en un cierto orden.

Salida: “True” o “False” dependiendo del resultado final.

1° Crear el objeto “Pieza de Dominó” el cual tiene por característica 2 números “N° Izquierdo” y “N° derecho” (Se asume una vista horizontal de las piezas), cada número izquierdo y derecho son asignados a cada pieza dependiendo de los números de las piezas que vienen por entrada. Este objeto posee un método llamado “Girar” el cual intercambia la posición del número izquierdo con el número derecho (Es un comportamiento no requerido pero que podría ayudar a conseguir un orden deseado en el “Estuche”).

2° Crear el objeto “Estuche” el cual contiene cada una de las piezas de dominó en un orden específico considerando los números izquierdo y derecho (El límite de piezas que puede contener el estuche no es especificado). Este objeto posee un estado “Validación” de tipo binario: “Aceptable” o “Inaceptable”, los cuales dependen del orden de las piezas que están inertes en el estuche. Si al introducir todas las piezas en el estuche, los números derechos eh izquierdos de cada pieza coinciden con los números izquierdos y derechos de sus piezas adyacentes entonces el estado del estuche se mantiene “Aceptable”, de no cumplirse lo anterior el estado pasa a ser “Inaceptable”.

3° Introducir los parámetros de entrada a cada pieza de dominó.

4° Introducir cada pieza de dominó en su estuche.

5° Si al introducir todas las piezas en el estuche el estado “Validación” es “Aceptable”, el programa arroja “True” y en el caso que el estado sea “Inaceptable”, el programa arroja “False”.

**Basado en el paradigma “Imperativo”:**

Entrada: Listado de piezas de dominó en un cierto orden.

Salida: “True” o “False” dependiendo del resultado final.

1° Crear una Validador que asigna valor true mientras no se muestre un orden de piezas incorrecto.

2° Hacer un ciclo que recorra toda la lista de las piezas (Pueden ser representados por arreglos o alguna clase de lista) comparando los números adyacentes entre piezas.

3° Si al recorrer la lista ordenada completa los números adyacentes entre piezas coinciden el programa retornará true.

4° Si al recorrer la lista se encuentra con al menos un par de números de piezas adyacentes que no coinciden el programa retornará false.

**5.- Implemente su diseño en pseudocódigo ad hoc al paradigma.**

Basado en el paradigma “Orientado a Objetos”:

Basado en el paradigma “Imperativo”:

Entrada: int arreglo de arreglos de tamaño 2 “Ordenamiento”: Representa los valores de cada pieza, int “CantidadDePiezas”: n° de piezas de dominó (\*Se asume que debe haber al menos una pieza de dominó).

Salida: Validador True or False dependiendo del caso

Bolean Validador= true

Para i=0 hasta (CantidadDePiezas-1) {

Si Ordenamiento [i][1] == Ordenamiento [i+1][0] o Ordenamiento [i+1][0]== NULL

Validador= true

Sino

Validador= false

Return Validador

Return Validador

}

**6.- Por ejemplo, no está permitido el arreglo de piezas (2 3) (4 2) (2 2) (3 5) y si está permitido (2 3) (3 5) (5 4) (4 5).**

Parte B

**5.- A través de que etapas tuvo que pasar mientras trabajaba en la solución en cada paradigma de programación?**

Antes de comenzar con el diseño del programa me fue necesario investigar las reglas o condiciones que tiene cada paradigma para poder visualizar el enfoque que debía darle al programa. Luego tuve que asimilar estas reglas para realizar el programa que resolvería el problema dado. Finalmente para la implementación en pseudo-código se me fue imposible no pensar en algún lenguaje de programación utilizado con anterioridad para darle forma al pseudo-código, según creo es la parte más difícil ya que es justo en la parte donde más se nota el paradigma utilizado.

**6.- ¿Hubo etapas similares y/o diferentes?**

Las etapas fueron completamente distintas ya que el enfoque para llegar a la solución es distinta, como si se utilizaran otras herramientas u otra forma de visualizar el problema.

**7.- Para cada paradigma de programación, describa cuales de sus características son expresadas en su solución?**

Las características expresadas en mi algoritmo basado en paradigma imperativo radica en el conjunto de instrucciones que dan con el resultado final utilizando asignaciones, ciclos y ramificaciones.

Por otra parte las características expresadas en mi algoritmo basado en paradigma orientado a objetos se ven en la abstracción de objetos que poseen interacción entre ellos para dar con el resultado pedido.

**8.- Si tuviera que escoger un paradigma de programación para resolver el problema ¿Cuál preferiría? Explique su selección.**

Creo que el más indicado para resolver este problema es el paradigma Orientado a Objetos ya que al trabajar con un juego de mesa como lo son las piezas de Dominó, la abstracción de objetos se visualiza con mucha facilidad y da espacio a resolver más problemas si se agregaran los diversos usos o métodos de las piezas en sí.