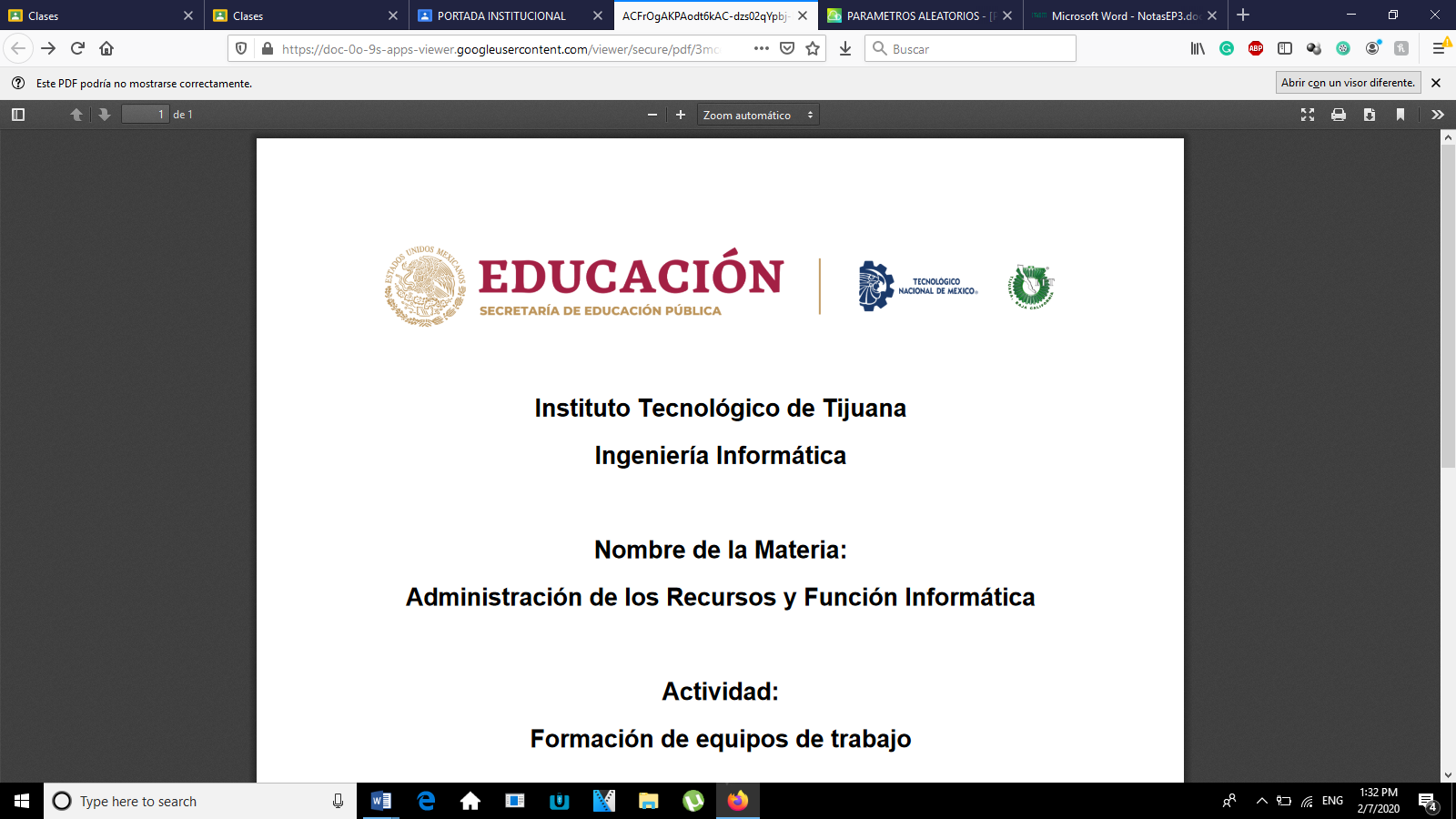
Tecnológico Nacional de México Instituto Tecnológico de Tijuana

SUBDIRECCIÓN ACADÉMICA

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

SEMESTRE: FEBRERO – JUNIO 2022

Ingeniería en sistemas computacionales

Estructura de datos serie 2SC3D

Tarea 2.1

Unidad 1- introducción a las estructuras de datos

Sabino Andres Manuel Alejandro 20211841

NOMBRE DEL MAESTRO

**M.C.C. LUZ ELENA CORTEZ GALVAN**

FECHA DE ENTREGA 01/03/2022

**CONTENIDO DEL TRABAJO:**

Contenido

[¿Qué es la recursividad? 3](#_Toc97027122)

[¿Cuáles son los tipos de recursividad que existen? 3](#_Toc97027123)

[Ventajas y Desventajas de la recursividad. 4](#_Toc97027124)

[Programa ejemplo en C# consola que utilice recursividad. 5](#_Toc97027125)

[BIBLIOGRAFÍAS 6](#_Toc97027126)

# ¿Qué es la recursividad?

Recursión o recursividad es la forma en la cual se especifica un proceso basado en su propia definición.1​ La recursión tiene esta característica discernible en términos de autorreferencialidad, autopoiesis, fractalidad, o, en otras palabras, construcción a partir de un mismo tipo. Con ánimo de una mayor precisión, y para evitar la aparente circularidad en esta definición.

Un problema que pueda ser definido en función de su tamaño, sea este N, pueda ser dividido en instancias más pequeñas (< N) del mismo problema y se conozca la solución explícita a las instancias más simples, lo que se conoce como casos base, se puede aplicar inducción sobre las llamadas más pequeñas y suponer que estas quedan resueltas.

# ¿Cuáles son los tipos de recursividad que existen?

**Recursividad simple o directa.**

La función incluye una referencia explícita a sí misma.

**Recursividad mutua o indirecta.**

El módulo llama a otros módulos de forma anidada y en la última llamada se llama al primero.

**De cabeza.**

La llamada se hace al principio del subprograma, de forma que el resto de instrucciones se realizan después de todas las llamadas recursivas.

**De cola.**

La llamada se hace al final del subprograma, de forma que el resto de forma que el resto de instrucciones se realizan antes de hacer la llamada.

**Intermedia.**

Las instrucciones aparecen tanto antes como después de las llamadas.

**Múltiple.**

Se producen varias llamadas recursivas en distintos puntos del subprograma.

**Anidada.**

La recursión se produce en un parámetro de la propia llamada recursiva.

La llamada recursiva utiliza un parámetro que es resultado de una llamada

recursiva.

# Ventajas y Desventajas de la recursividad.

**Ventajas:**

**mayor simplicidad el condigo en problemas recursivos**

Si un problema se puede definir fácilmente de forma recursiva (por ejemplo, la factorial o la potencia) es código resultante puede ser más simple que el equivalente iterativo.

o También es muy útil para trabajar con estructuras de datos que se pueden definir de forma recursiva, como los árboles.

**Posibilidad de marcha atrás. back tracking**

Las características de la pila de llamadas hacen posible recuperar los datos en orden inverso a como salen, posibilitando cualquier tipo de algoritmo que precise volver hacia atrás.

**Desventajas:**

**Mayor uso de la pila de memoria.**

Cada llamada recursiva implica una nueva entrada en la pila de llamadas dónde se cargará tanto la dirección de retorno como todos los datos locales y argumentos pasados por valor.

El tamaño que reserva el compilador a la pila de llamadas es limitado y puede agotarse, generándose un error en tiempo de compilación.

**Mayor tiempo en las llamadas.**

Cada llamada a un subprograma implica: Cargar en memoria el código del procedimiento.

o Cargar en memoria el código del procedimiento.

o Meter en la pila la dirección de retorno y una copia de los parámetros pasados por valor.

o Reservar espacio para los datos locales.

o Desviar el flujo del programa al subprograma, ejecutarlo y retornar al programa llamador.

Esto implica un mayor tiempo de ejecución, sobre todo si hay muchas llamadas anidadas, algo normal en programas recursivos.

# Programa ejemplo en C# consola que utilice recursividad.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

class Program

{

public static int CalculateSumRecursively(int n, int m)

{

int sum = n;

if (n < m)

{

n++;

return sum += CalculateSumRecursively(n, m);

}

return sum;

}

static void Main(string[] args)

{

Console.WriteLine("introduce el numero n: ");

int n = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("introduce el numero m: ");

int m = Convert.ToInt32(Console.ReadLine());

int sum = CalculateSumRecursively(n, m);

Console.WriteLine(sum);

Console.ReadKey();

}

}

# BIBLIOGRAFÍAS

colaboradores de Wikipedia. (2022a, enero 31). *Recursión*. Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/Recursi%C3%B3n>