Anthony Fabian Ramirez Orellana

Carne: 9490-22-958

Sección: “A”

Catedrático: Jorge Pérez

Tarea 10 (Diferencias divididas).

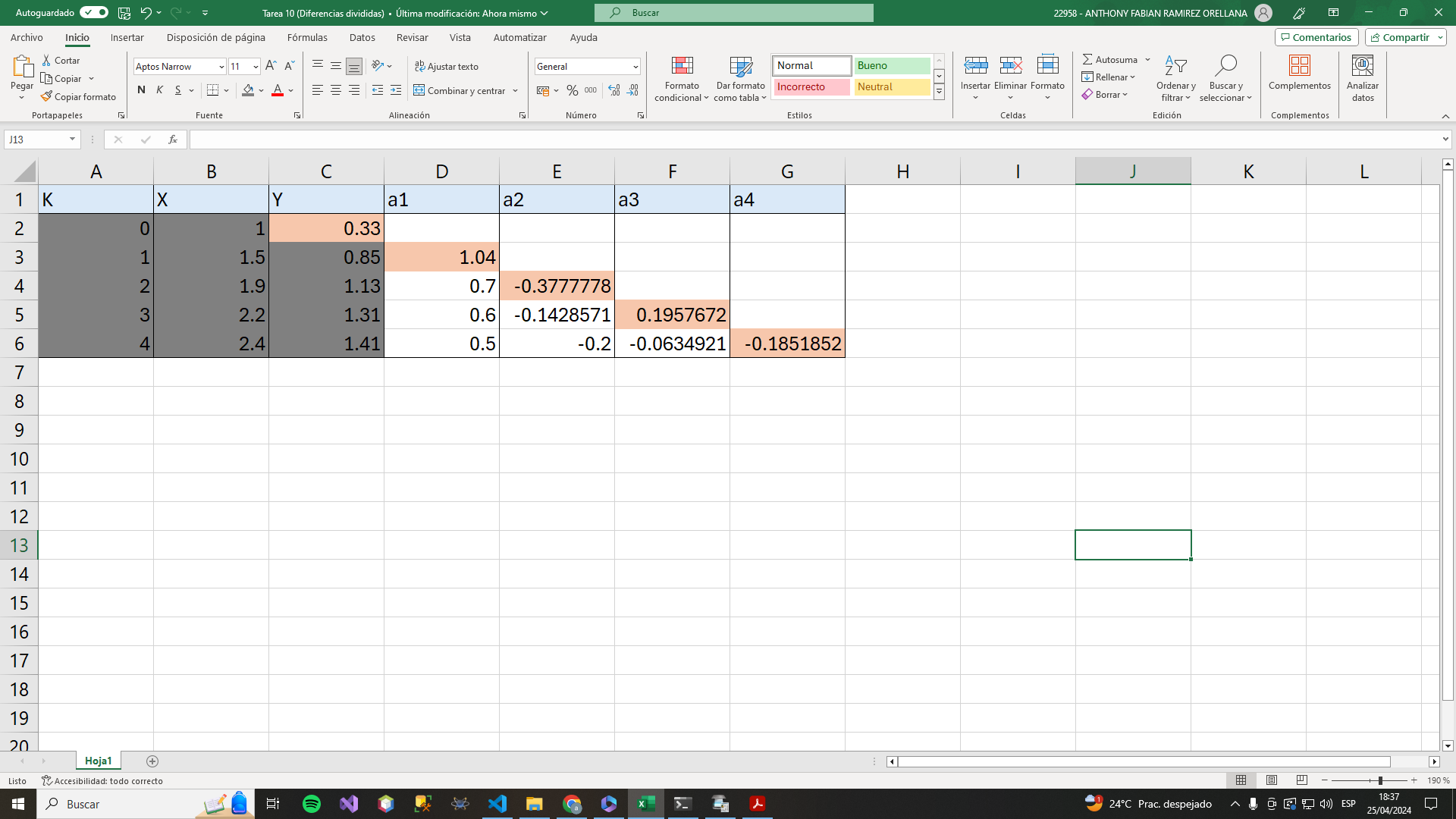
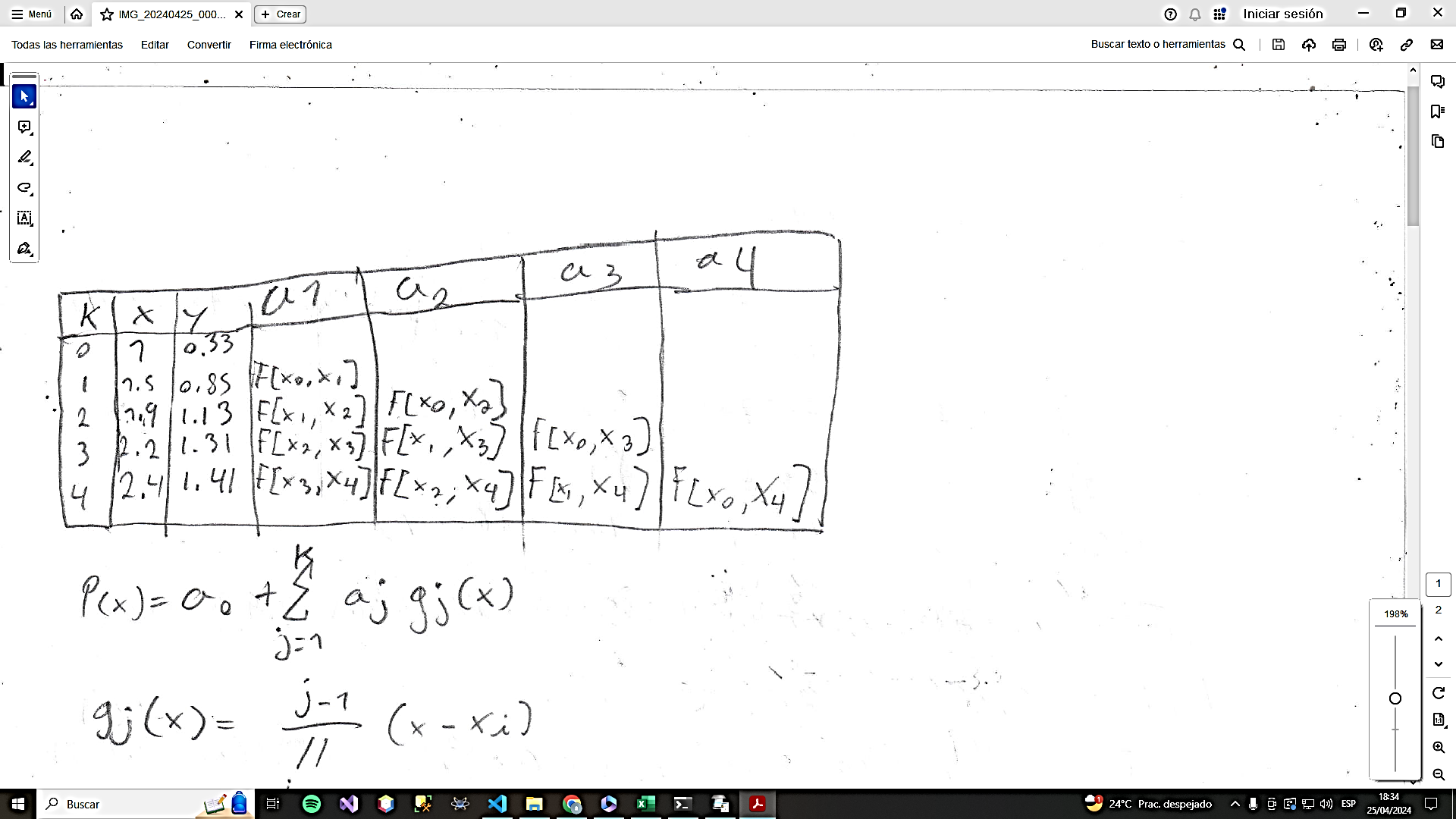
27/04/2024

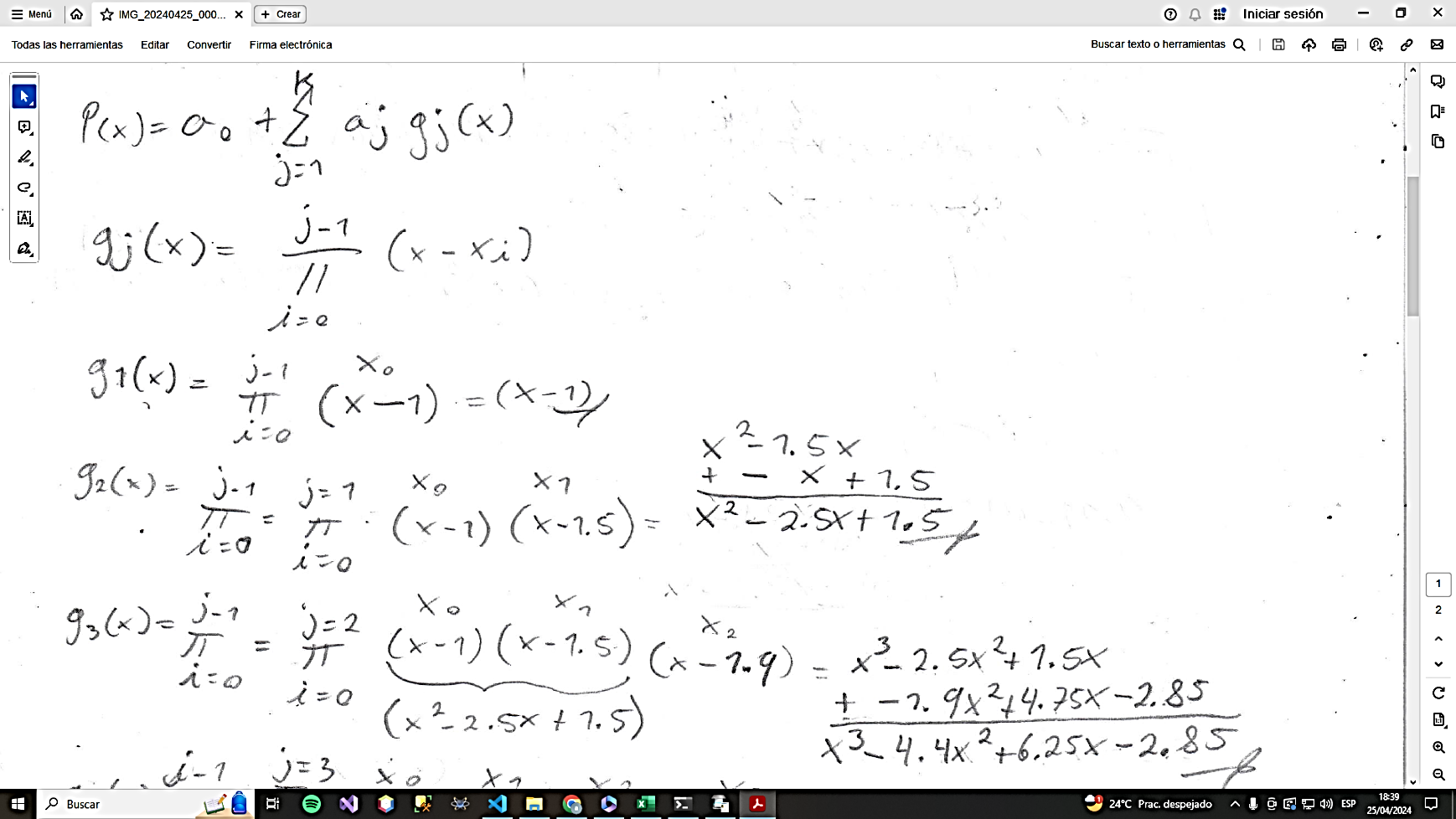
**TAREA No 10**

Dada la siguiente tabla de valores, encuentre el polinomio interpolador utilizando el método de Diferencias Divididas. Utilice el polinomio resultante para interpolar en los valores x=1.35; x=2.1; x=2.3.

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente





Diagrama

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

**PROBLEMA:**

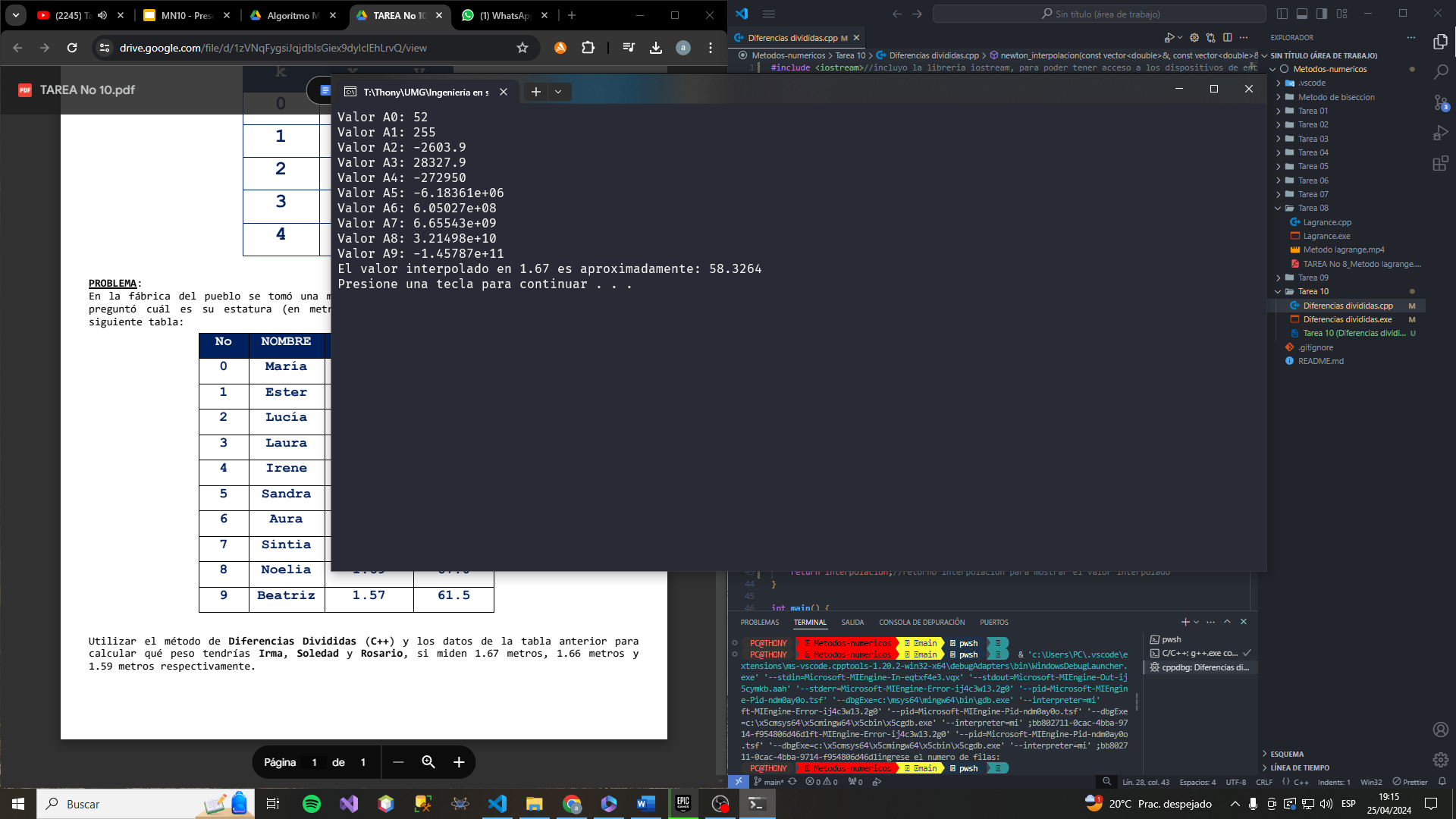
En la fábrica del pueblo se tomó una muestra de 10 personas de sexo femenino, y se les preguntó cuál es su estatura (en metros) y su peso (en kilogramos), y se obtuvo la siguiente tabla:

Tabla

Descripción generada automáticamente

Utilizar el método de Diferencias Divididas (C++) y los datos de la tabla anterior para calcular qué peso tendrías Irma, Soledad y Rosario, si miden 1.67 metros, 1.66 metros y 1.59 metros respectivamente.

Interpolación 1.67:



Interpolación 1.66:

Texto

Descripción generada automáticamente

Interpolación 1.59:

Captura de pantalla de computadora

Descripción generada automáticamente

Código:

#include <iostream>//incluyo la libreria iostream, para poder tener acceso a los dispositivos de entrada y salida

#include <vector>//Es una platilla de clase que proporciona un contenedor de memoria dinamica, puede contener un numero variable de elementos del mismo tipo

using namespace std;//importa todo el espacio de nombres std al codigo actual, se utiliza mayormente para utilizar el prefijo std::

// Función para calcular las diferencias divididas

vector<vector<double>> **diferencias\_divididas**(const vector<double>& x, const vector<double>& y) {//creo una funcion de el tipo vector variable tipo dobule con nombre diferencias\_divididas y le paso como parametros el vector "x" y el "y" constantes para no modificarlos la calcular

    int n = x.**size**();//creo la variable n igualada a la cantidad de valores en x, el tamaño de x

    vector<vector<double>> **tabla**(n, **vector**<double>(n, 0.0));//cre un vector variable tipo double llamado tabla pasandole el valor de n y el de vector double en posicion n, 0.0

    // Copiar "y" a la primera columna de la tabla

    for (int i = 0; i < n; ++i) {

        tabla**[**i**][**0**]** = y**[**i**]**;

    }

    // Calcular las diferencias divididas

    for (int j = 1; j < n; ++j) {

        for (int i = 0; i < n - j; ++i) {

            tabla**[**i**][**j**]** = (tabla**[**i+1**][**j-1**]** - tabla**[**i**][**j-1**]**) / (x**[**i+j**]** - x**[**i**]**);

        }

    }

    return tabla;//para trabajr con la siguiente funcion

}

// Función para realizar la interpolación de Newton

double **newton\_interpolacion**(const vector<double>& x, const vector<double>& y, double valor) {//esta funcion es de tipo double y paso los valros constantes del vector variable tipo dobule "x" y el "y", tambien el valor a interpolar

    vector<vector<double>> tabla = **diferencias\_divididas**(x, y);//creo mi vector variable tipo double de nombre tabla y lo igualo a la funcion anterior para calcular los valores

    int n = x.**size**();//creo n de nuevo del tamaño de x para la manipulacion de los for

    double interpolacion = tabla**[**0**][**0**]**;//creo la variable double interpolacon para guardar los resultados

    cout **<<** "Valor A0"**<<**": "**<<**tabla**[**0**][**0**]<<endl**;//despliego el primero que es fijo de la tabla entregada por el usuario

    for (int i = 1; i < n; ++i) {//inicio con los for para calcular los valores, desde 1 ya que el anterior era el 0

        double termino = tabla**[**0**][**i**]**;//creo mi variable termino para almacenar el valor cada vez que se regrese en el for

        cout **<<** "Valor A"**<<**i**<<**": "**<<**termino**<<endl**;//despliego el valor en consola para mostrar los distintos A

        for (int j = 0; j < i; ++j) {//comienzo con el siguiente for, esta vez con j en cero, sienfo j menor que i

            termino \*= (valor - x**[**j**]**);// calculo termino = tremino \* (valor - x[j]), solamente acortando un poco la linea con =\*

        }

        interpolacion += termino;// sumo interpolacion = interpolacion + termino, para ir calculando el valor final

    }

    return interpolacion;//retorno interpolacion para mostrar el valor interpolado

}

int **main**() {

    int n = 0;//creo una variable entera n para pedir el numero total de valores en la tabla

    // Puntos conocidos para la interpolación

    cout**<<**"ingrese el numero de filas: "**<<endl**;//despliego un mensaje para pedir el valor de filas, utilizando n

    try//abro un try and catch para prevenir ingreso de datos erroneos

    {

        cin**>>**n;//ingreso el valor del usuairio en la variable n

    }

    catch(const invalid\_argument& e)//de aber un error la aplicacion no calculara nada y se cerrara

    {

        cout **<<** e.**what**() **<<** '\n';//despliego en consola que hubo un error en la ejecucion

    }

    vector<double> **x**(n);//creo 2 vectores con tamaño variable, les paso de una vez su valor de numero de datos

    vector<double> **y**(n);

    cout**<<**"Ingrese los valores Xs, Ys"**<<endl**;//doy contexto de los valores que pedire a continuacion.

    for(int js = 0; js < 2; js++){//comienzo a pedir los valores para rellenar al tabla de valores

        string col;//creo un string col, para desplegar la columna que estoy pidiendo, si es la de x o la de y

        if(js+1 == 1){//con un condicional valido si estamos en la columna 1 o en la columna 2

            col **=** "X";//si estamos en la columna 1 entonces col es igual a X

            cout**<<**"Columna " **<<** col **<<** ": "**<<endl**;//ahora estructuro el mensaje para que se le haga mas falcil al usuiario ingresar los datos

            for(int iS = 0; iS < n; iS++){//comienzo a pedir los valores con for anidados

                cout**<<**"valor "**<<**iS+1**<<**": "**<<endl**;//despliego un mensaje para el valor de la fila

                try//abro un try and catch para prevenir ingreso de datos erroneos

                {

                    cin**>>**x**[**iS**]**;//pido el valor y lo inserto en la posicion de la matriz correspondiente

                }

                catch(const std::invalid\_argument& e)//de aber un error la aplicacion no calculara nada y se cerrara

                {

                    cout **<<** e.**what**() **<<** '\n';//despliego en consola que hubo un error en la ejecucion

                }

            }

        }else {

            col **=** "Y";//sino col es igual a Y

            cout**<<**"Columna " **<<** col **<<** ": "**<<endl**;//ahora estructuro el mensaje para que se le haga mas falcil al usuiario ingresar los datos

            for(int iS = 0; iS < n; iS++){//comienzo a pedir los valores con for anidados

                cout**<<**"valor "**<<**iS+1**<<**": "**<<endl**;//despliego un mensaje para el valor de la fila

                try//abro un try and catch para prevenir ingreso de datos erroneos

                {

                    cin**>>**y**[**iS**]**;//pido el valor y lo inserto en la posicion de la matriz correspondiente

                }

                catch(const std::invalid\_argument& e)//de aber un error la aplicacion no calculara nada y se cerrara

                {

                    cout **<<** e.**what**() **<<** '\n';//despliego en consola que hubo un error en la ejecucion

                }

            }

        }

    }

    // Valor a interpolar

    double valor\_a\_interpolar;

    cout**<<**"ingrese el valor de a interpolar Z: "**<<endl**;//pido el valor de z

    try///abro un try and catch para prevenir ingreso de datos erroneos

    {

        cin**>>**valor\_a\_interpolar;//ingreso el valor del usuario en mi variable puntoZ

    }

    catch(const invalid\_argument& e)//de aber un error la aplicacion no calculara nada y se cerrara

    {

        cout **<<** e.**what**() **<<** '\n';//despliego en consola que hubo un error en la ejecucion

    }

**system**("cls");//utilizo system para limpiar la consola de la recopilacion de datos

    // Realizar la interpolación y mostrar el resultado

    double resultado = **newton\_interpolacion**(x, y, valor\_a\_interpolar);//creo una variable resultado y la igualo a mi funcion para iniciar con todos los calculos, mandando los vectores "x", "y" y el valor a interpolar

    cout **<<** "El valor interpolado en " **<<** valor\_a\_interpolar **<<** " es aproximadamente: " **<<** resultado **<<** std::**endl**;//despliego el resultado en consola

**system**("pause");//utilizo sistem para pausar el programa

    return 0;

}

Video del funcionamiento:

<https://drive.google.com/file/d/1QDdsj3jBEzrxwPPgCTM4_OuzQ78TWYf-/view?usp=sharing>