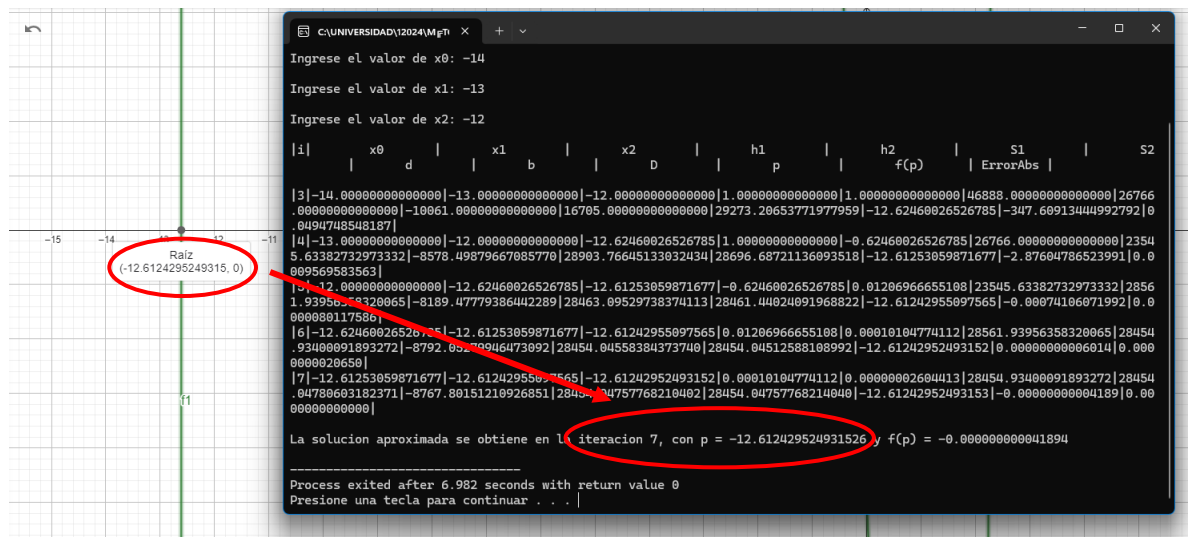


EDWIN ADONY MONTEJO MARTÍNEZ 9490-21-3898

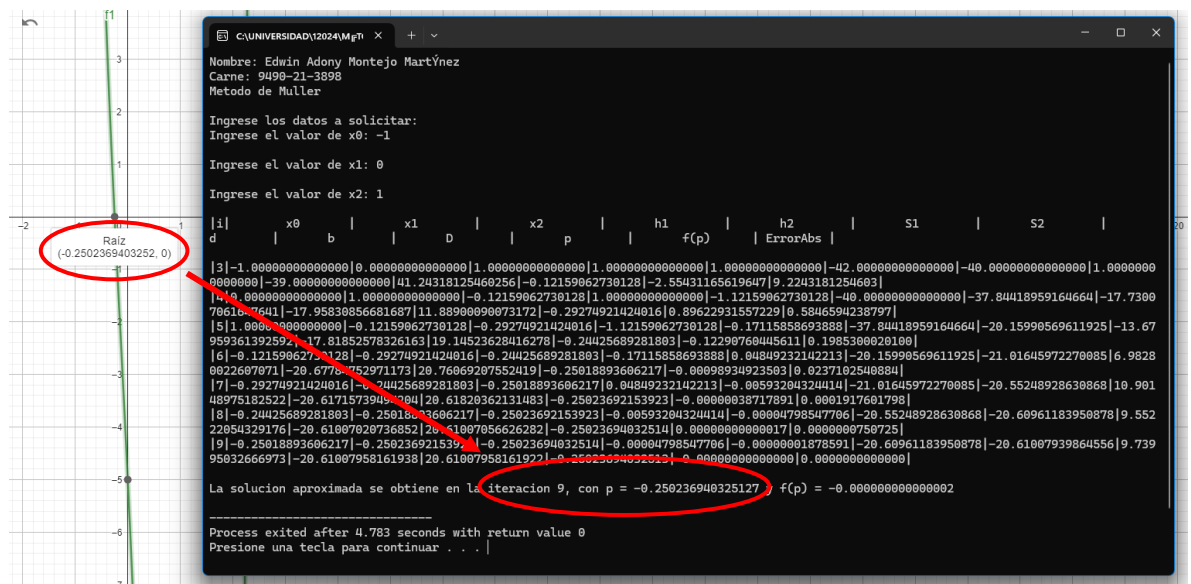
a) $p_1(x) = x^5 + 11x^4 - 21x^3 - 10x^2 - 21x - 5 = 0$

Raíces:

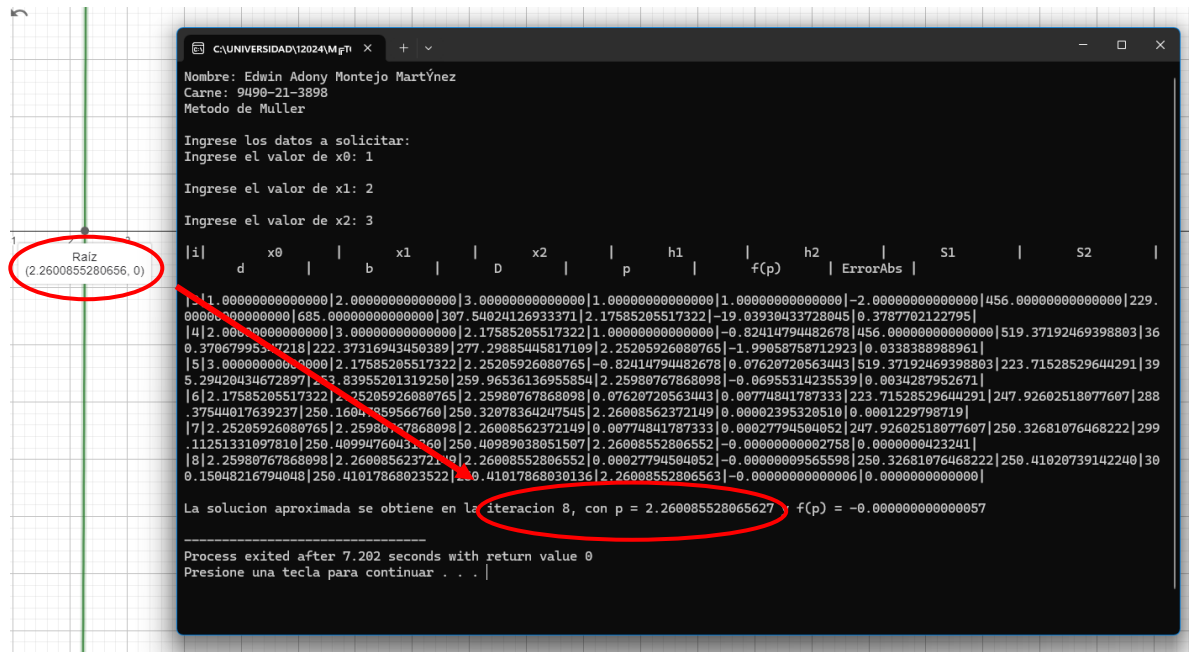
1. -12.612429524931526



2. -0.250236940325127

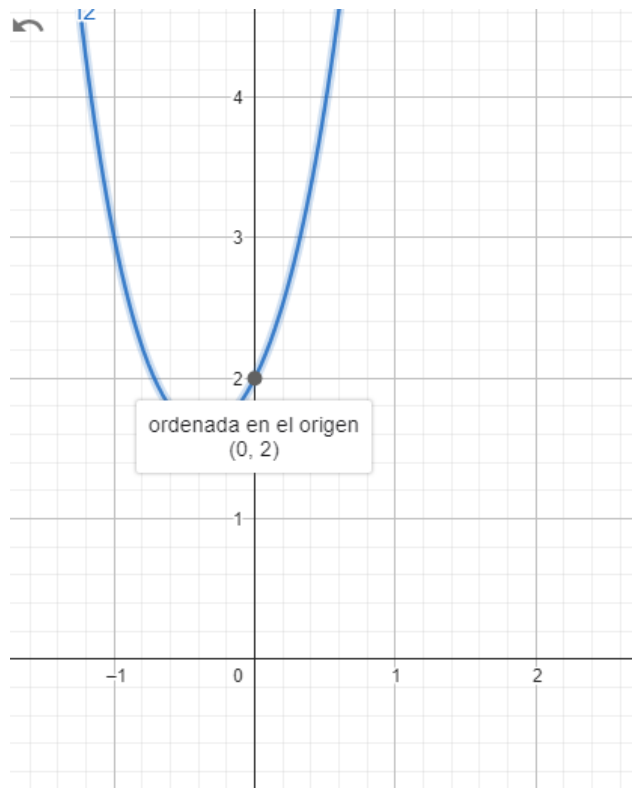


3. 2.260085528065627



b) $p_2(x) = x^4 + x^3 + 3x^2 + 2x + 2 = 0$

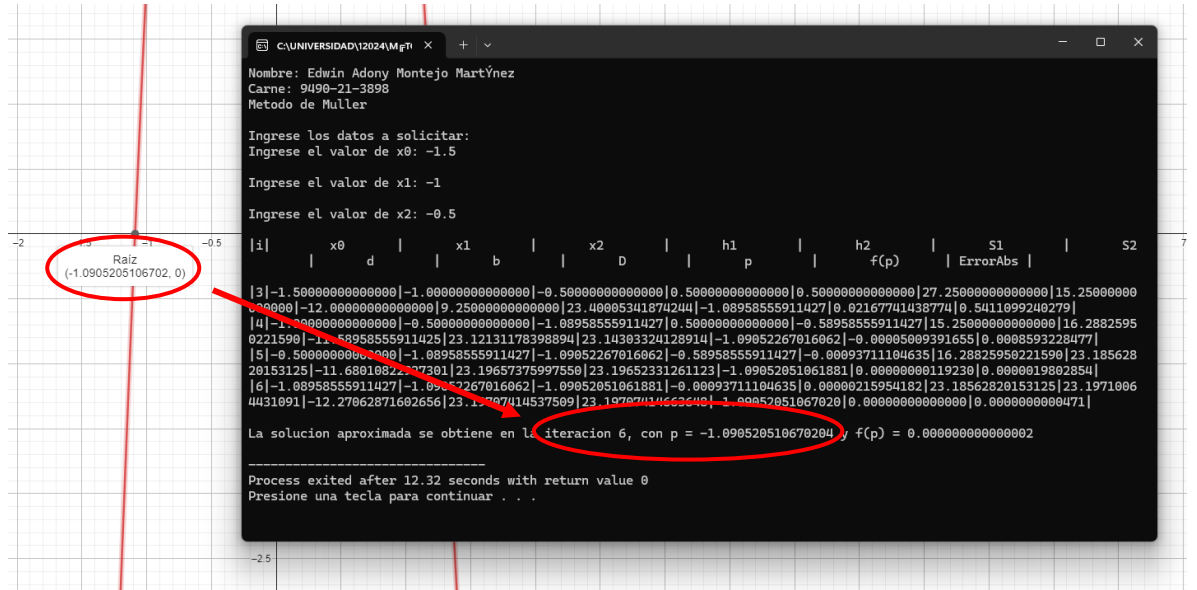
Tiene raíces imaginarias



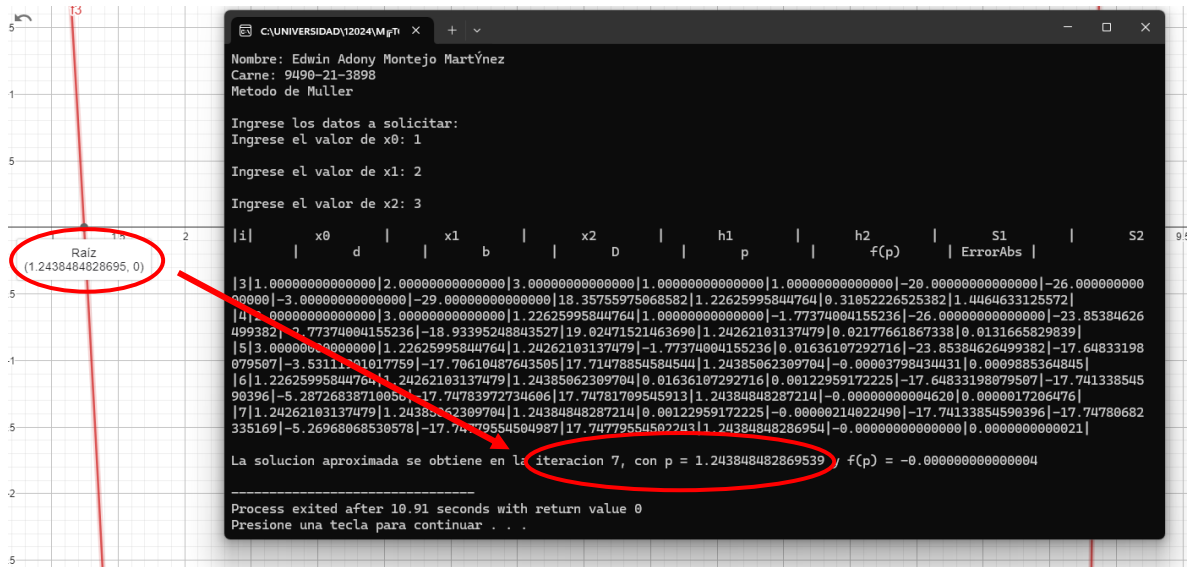
c) $p_3(x) = x^3 - 9x^2 + 12 = 0$

Raíces:

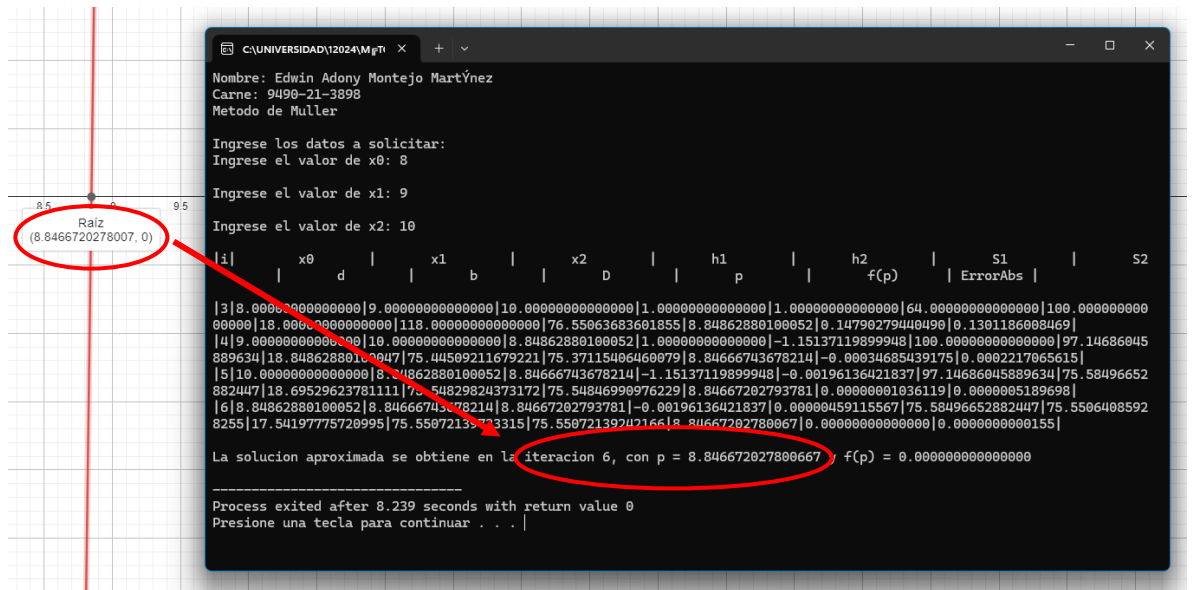
1. -1.090520510670204



2. 1.243848482869539



3. 8.846672027800667



CODIGO C++

```
#include <iostream>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>
#include <cstdlib>
#include <windows.h>
#include <cmath>
#include <iomanip>

using namespace std;

int i, IT, salir;
double x0, x1, x2, TOL, p, h1, h2, S1, S2, b, d, D, E, h, expresion, absh, abs1,
abs2, errAbs;

double f(double x){
    return pow(x,3)-9*pow(x,2)+12;
}

int iteracion(){

    printf("|i|    x0    |    x1    |    x2    |    h1    |    h2    |    S1    |
S2    |    d    |    b    |    D    |    p    |    f(p)    | ErrorAbs |\n\n");

}

int pedirDatos(){

    printf("Nombre: Edwin Adony Montejo Martínez\n");
    printf("Carne: 9490-21-3898\n");
    printf("Metodo de Muller\n");
    printf("\n");
    printf("Ingrese los datos a solicitar: \n");
    printf("Ingrese el valor de x0: ");cin>>x0;
    printf("\n");
    printf("Ingrese el valor de x1: ");cin>>x1;
    printf("\n");
    printf("Ingrese el valor de x2: ");cin>>x2;
    printf("\n");
```

```

    TOL = pow(10,-10);
    IT=100;
}

int salida(int caso){
    if(caso == 0){
        printf("\n");
        printf("FRACASO, se superaron las cantidades maximas de
iteraciones permitidas \n");
        printf("sin que se alcanzara una aproximacion valida. \n");
        printf("\n");
    }
    else{
        printf("\n");
        printf("La solucion aproximada se obtiene en la iteracion %i, con
p = %4.15f y f(p) = %4.15f\n", i, p, f(p));

    }
}

```

```

int metodo(){

    salir = 0;
    i = 3;
    h1 = x1 - x0;
    h2 = x2 - x1;
    S1 = (f(x1) - f(x0)) / h1;
    S2 = (f(x2) - f(x1)) / h2;
    d = (S2 - S1) / (h2 + h1);
    iteracion();

    do{

        b = S2 + h2 * d;
        expresion = (pow(b,2) - 4 * f(x2) * d);
        if(expresion < 0){
            printf("La funcion tiene raices imaginarias");
            return (0);
        }
        else{
            D = pow((pow(b,2) - 4 * f(x2) * d), 0.5);
        }

        abs1 = abs(b - D);
        abs2 = abs(b + D);
    }
}

```

```

        if(abs1 < abs2){
            E = b + D;
        }
        else{
            E = b - D;
        }

        h = -2 * f(x2) / E;
        p = x2 + h;

        errAbs = abs((p - x2) / p);

        printf("|%i|%4.14f|%4.14f|%4.14f|%4.14f|%4.14f|%4.14f|%4.14f|
        %4.14f|%4.14f|%4.14f|%4.14f|%4.13f|\n", i, x0, x1, x2, h1, h2, S1, S2, d, b,
        D, p, f(p), errAbs);

        absh = abs(h);
        if(f(p) == 0 || absh < TOL){
            salir = 1;
            salida(1);
        }
        else{
            if(salir == 0){
                x0 = x1;
                x1 = x2;
                x2 = p;
                h1 = x1 - x0;
                h2 = x2 - x1;
                S1 = (f(x1) - f(x0)) / h1;
                S2 = (f(x2) - f(x1)) / h2;
                d = (S2 - S1) / (h2 + h1);
                i++;
            }
        }
    }

}while((i<IT) && (salir==0));

if(salir == 0 ){
    salida (0);
}

```

```
        return(0);  
    }  
int main(){  
    pedirDatos();  
    metodo();  
}
```