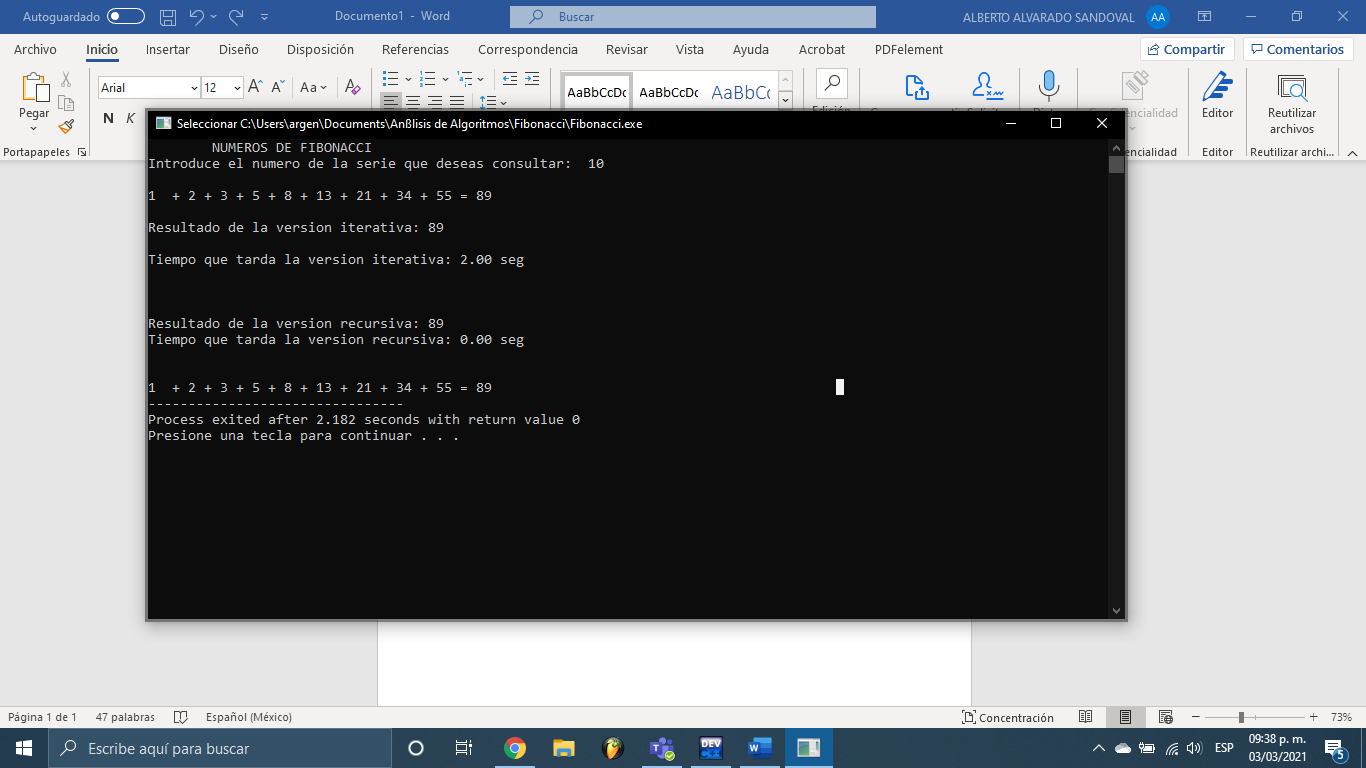
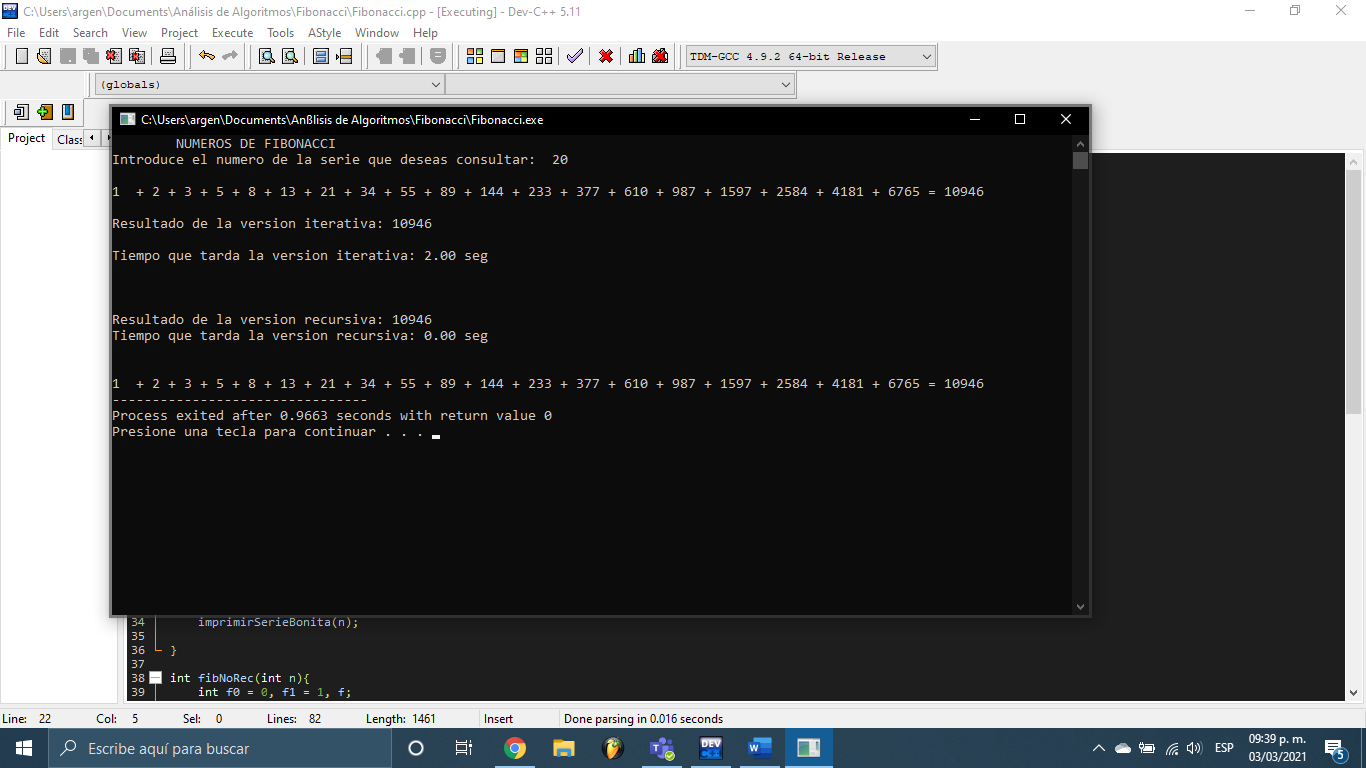
Alvarado Sandoval Alberto 3CV14

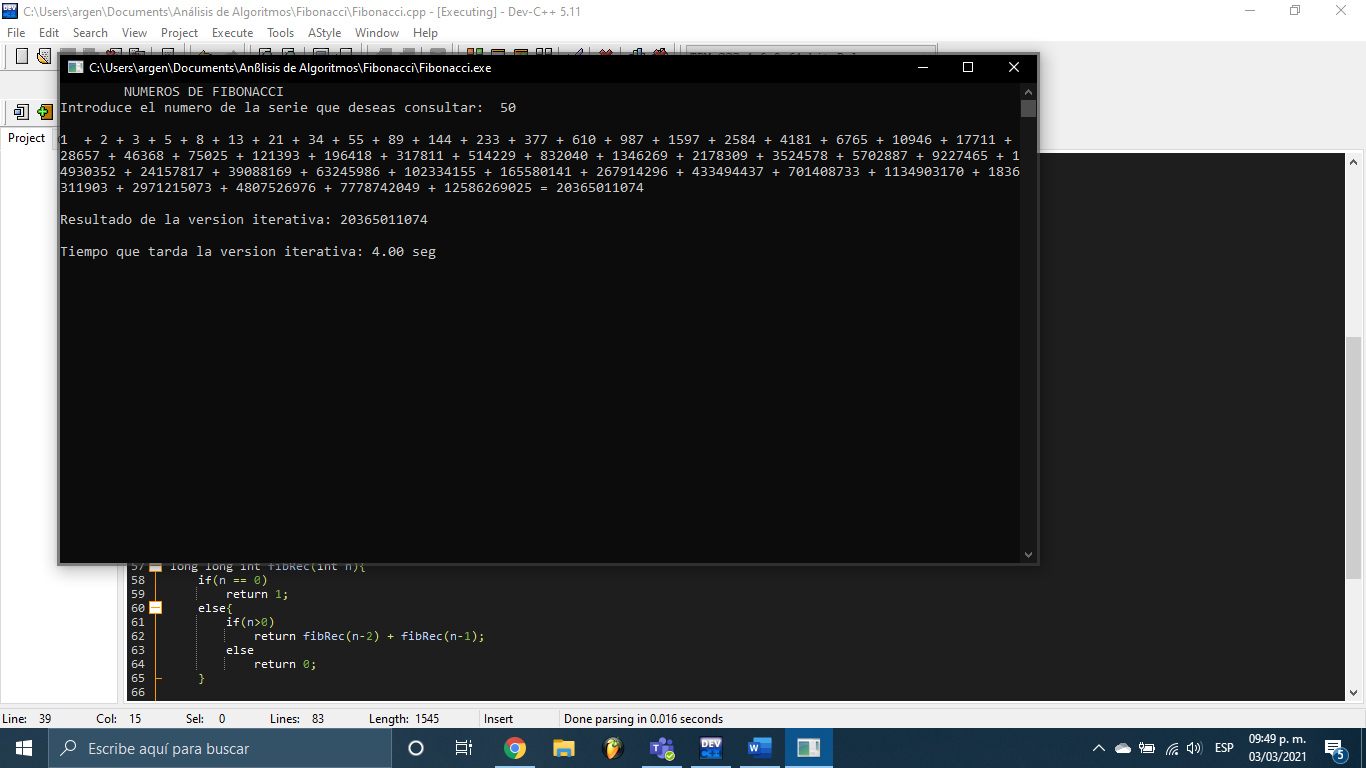
Análisis a versión iterativa y recursiva del Algoritmo de Fibonacci

Me dediqué a realizar el siguiente código que implementa ambos algoritmos, y mide su tiempo. En base a mediciones de ingresar como parámetros 10, 20 y 50 al programa obtuve los siguientes resultados.





El caso de 50 tardó demasiado.



Con números pequeños, el algoritmo funciona mucho mejor con la versión recursiva pero después vemos que la iterativa tarda mucho menos.

Código:

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

long long int fibNoRec(int);

long long int fibRec(int);

void imprimirSerieBonita(int);

int main(void){

clock\_t t0n, t0r, tfn, tfr;

double sn, sr;

int n, i=0;

printf("**\t**NUMEROS DE FIBONACCI**\n**Introduce el numero de la serie que deseas consultar: ");

scanf("%d", &n);

t0n = clock();

printf("**\n\n**Resultado de la version iterativa: %lld**\n** ",fibNoRec(n));

tfn = clock();

sn = (double)(tfn - t0n) / CLOCKS\_PER\_SEC \* 1000;

printf("**\n**Tiempo que tarda la version iterativa: %.2f seg**\n\n**", sn);

t0r = clock();

printf("**\n\n**Resultado de la version recursiva: %lld **\n**",fibRec(n));

tfr = clock();

sr = (double)(tfr - t0r) / CLOCKS\_PER\_SEC \*1000;

printf("Tiempo que tarda la version recursiva: %.2f seg**\n\n**", sr);

imprimirSerieBonita(n);

}

long long int fibNoRec(int n){

long long int f0 = 0, f1 = 1, f;

int i;

for(i = 0; i<n; i++){

f = f0 + f1;

f0 = f1;

f1 = f;

if(i == 0)

printf("**\n**%lld ",f);

else{

if(i == n-1)

printf(" = %lld", f);

else

printf(" + %lld", f);

}

}

return f;

}

long long int fibRec(int n){

if(n == 0)

return 1;

else{

if(n>0)

return fibRec(n-2) + fibRec(n-1);

else

return 0;

}

}

void imprimirSerieBonita(int n){

int i = 1;

long long int f;

for(i = 1; i<n+1; i++){

f = fibRec(i);

if(i == 1)

printf("**\n**%lld ",f);

else{

if(i == n)

printf(" = %lld", f);

else

printf(" + %lld", f);

}

}

}

Líneas de código iterativo: 17

Líneas de código recursivo: 9