**ENTREGABLE**

**COMPLEJIDAD**

**Ejercicio 1.a)**

static double f1a(int n) {

int prod = 1;

double suma = 0.0;

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

prod = 1;

for (int j = 1; j <= i; j++)

prod = prod \* j;

suma = suma + 1 / prod;

}

return suma;

}

**Ejercicio 1.b)**

static double f1b(int n) {

double sum = 0.0;

int j = 0;

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

for (int j = 100; j > 0; j--){

sum = sum + j / i;

}

}

return sum;

}

**Ejercicio 1.c)**

static boolean esSinRepetidos(String s)

{

int n = s.length()

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

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

if (s.charAt(i) == s.charAt(j)){

return false;

}

return true;

}

}

**Ejercicio 2)**

static void invertir(int[] arreglo) {

int i = 0;

int n = arreglo.length;

while (i < n / 2) {

swap(arreglo, i, n - 1 - i);

i++;

}

}

**Ejercicio 3)**

static void f3(int n) {

int i = 1;

while (i < n) {

if (i % 2 == 0) {

/\* bloque de código de orden O(n) \*/

} else {

/\* bloque de código de orden O( log n) \*/

}

i++;

}

}

**Ejercicio 4)**

static int f4(int n) {

int c = 0;

int a = 1;

while (a < n) {

a = 2 \* a;

c++;

}

return c;

}

**Ejercicio 5)**

static int potenciaDe2(int n) {

if (n == 0){

return 1;

}

return 2 \* potenciaDe2(n - 1);

}