

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de ingeniería en ciencias y sistemas
Organización de lenguajes y compiladores 2
Primer semestre 2018

Catedráticos: Ing. Bayron López, Ing. Edgar Sabán, Ing. Erick Navarro
Tutores: José Cano, Daniel Álvarez, Javier Navarro, Mike Gutiérrez



Hoja de calificación proyecto 2

Fecha Calificación: ____/____/____

Nombre: _____ Registro académico: _____

NOTA: Los requerimientos mínimos del proyecto son funcionalidades del sistema que permitirán un ciclo de ejecución básica. En el caso de no cumplir con **uno o más** de los siguientes requerimientos **no se calificará el proyecto y el estudiante tendrá una nota de cero puntos.**

No	Requerimientos mínimos	Cumple completamente	No cumple
1.	Draco Compiler		
2.	IDE		
3.	Editor de Texto		
4.	Consola		
5.	Tabla de Símbolos		
6.	Errores		
7.	Micro-Navegador		
8.	Lenguaje interpretado DracoScript		
9.	Características del lenguaje		
10.	Variables		
11.	Declaración de variable		
12.	Asignación de variables		
13.	Tipos de datos		
14.	Operando		
15.	Condiciones		
16.	Expresiones relacionales		
17.	Expresiones lógicas		
18.	Sentencias de control		
19.	Bucles		
20.	For		
21.	Imprimir		
22.	Funciones nativas de integración		
23.	Funciones nativas de dibujo		
24.	Puntos		
25.	Cuadrados		
26.	Cadenas		
27.	Lenguaje de alto nivel D++		
28.	Case sensitive		
29.	Recursividad		
30.	Variables		
31.	Tipos de datos		

32.	Operaciones aritméticas		
33.	Operaciones relacionales		
34.	Operaciones lógicas		
35.	Declaración de variables		
36.	Asignación de variables		
37.	Arreglos de una dimensión		
38.	Asignación		
39.	Declaración e inicialización		
40.	Sentencia de selección		
41.	Si		
42.	Sentencias cíclicas		
43.	Mientras		
44.	Para		
45.	Sentencias de transferencia		
46.	Detener		
47.	Retornar		
48.	Estructuras		
49.	Acceso a los atributos de estructuras		
50.	Métodos y Funciones		
51.	Llamadas de métodos y funciones		
52.	Método principal		
53.	Funciones nativas de dibujo		
54.	Punto		
55.	Cuadrado		
56.	Cadena		
57.	Generación de código intermedio DASM (Todo)		

Ponderación

Los puntos se encuentran repartidos en cinco secciones. Cada sección será evaluada sobre cien puntos para facilitar la calificación. El puntaje neto de cada sección se encuentra indicado en la siguiente tabla.

DESCRIPCION	Ponderación	Puntaje Obtenido
1. Componentes de la aplicación	18	
2. Manejo de funcionalidades básicas.	18	
3. Manejo de arreglos de una y múltiples dimensiones	19	
4. Manejo intermedio de funcionalidades	20	
5. Manejo avanzado de funcionalidades	25	
TOTAL	100	

Nota: Para tener nota completa en cada punto de cada sección deberá tener una salida exacta y esperada caso contrario se asignará nota de cero puntos en la sección evaluada, es decir para tener derecho nota completa la salida esperada debe ser exacta, si existiese una salida aproximada, errónea o no es la esperada será ponderado con nota cero en el punto evaluado.

Componentes de la aplicación

En esta sección se calificará la **correcta implementación de los componentes** de la aplicación, para ello se comprobará que los mismos hayan sido incluidos y ejecuten su funcionalidad correctamente.

Tiempo estimado para la evaluación de esta sección: **10 minutos**.

DESCRIPCION	Ponderación	Puntaje obtenido
Componentes del IDE	15	
Editor de texto	3	
Agregar sangría	1	
Colores de fuente	1	
Línea y columna	1	
Barra de herramientas	4	
Nuevo archivo	1	
Nueva carpeta	1	
Guardar	1	
Cerrar	1	
Búsqueda	2	
Consola de Salida	2	
Tabla de errores	2	
Tabla de símbolos	2	
Micro-Navegador	5	
Dimensiones de lienzo correctas	2	
Colores y visualización de dibujos correcta	3	
Depurador de código	70	
Depurador de generación de código DASM	20	
Paso a paso (siguiente instrucción)	10	
Continuar (siguiente punto de ruptura)	10	
Depurador de código DracoScript y DASM	20	
Paso a paso (siguiente instrucción)	10	
Continuar (siguiente punto de ruptura)	10	
Estructuras en tiempo de ejecución	30	
Stack	10	
Heap	10	
Pila auxiliar	10	
Documentación	10	
Manuales	4	
Manual de usuario	2	
Manual técnico	2	

Gramáticas	6	
Gramática de D++	2	
Gramática de DracoScript	2	
Gramática DASM	2	
Subtotal (sobre 100)	100	
Total (sobre 18)	18	

Manejo de funcionalidades básicas.

En esta sección se calificará que las funcionalidades básicas del lenguaje D++, verificar que hayan sido implementadas correctamente. El archivo de entrada estará distribuido de la siguiente manera:

- **Salida correcta de operaciones:** esta parte del archivo de entrada evaluará el manejo correcto de las operaciones aritméticas, relacionales y lógicas. El archivo tendrá una serie de operaciones y serán asignadas a variables con el mismo nombre (pero cambiarán letras entre mayúsculas y minúsculas) para validar que los estudiantes hicieron casteos adecuados en las operaciones, que manejaron los decimales con exactitud y el uso correcto de case sensitive. También tendrá un conjunto de sentencias cíclicas y de selección, dependiendo del resultado de la condición entonces se ingresará a un bloque de instrucciones, si manejaron correctamente la condición entonces se ejecutará el bloque esperado.
- **Dibujar personajes:** esta parte del archivo permitirá evaluar el uso de las estructuras y su respectivo acceso a atributos, en las estructuras se almacenará información importante de cada personaje, para posteriormente ser mostrada gráficamente.
- **Dibujar unión de personajes:** esta parte del archivo permitirá evaluar el manejo de arreglos de una dimensión, se implementará de tal forma que en cada posición del arreglo se guarde un personaje y recorriendo los arreglos se identificará quienes son familiares entre todos los personajes.
- **Pintar personajes:** con base en las estructuras utilizadas en el inciso descrito anteriormente, en esta parte del archivo se evaluarán las funciones nativas del lenguaje (punto, cuadrado, cadena), con estas funciones se podrán dibujar los personajes, imprimirle sus respectivos atributos y crear relaciones entre dichos personajes.
- **Manejo errores:** se tendrá una parte del archivo con los diferentes tipos errores (léxicos, sintácticos y semánticos), estas funciones servirán solo para validar que si se manejó correctamente la detección y recuperación de errores. Estos errores deberán ser mostrados en el reporte de errores para que tenga validez.

Tiempo estimado para la evaluación de esta sección: **20 minutos.**

DESCRIPCION	Ponderación	Puntaje obtenido
Salida correcta de operaciones.	20	
Manejo correcto de Case Sensitive.	2	

Manejo expresiones aritméticas.	4	
Manejo correcto de decimales.	2	
Manejo correcto de casteos.	2	
Manejo expresiones relacionales.	2	
Manejo expresiones lógicas.	2	
Sentencias cíclicas.	5	
Sentencia selección (Si).	5	
Dibujar unión de personajes.	25	
Asignar correctamente la relación entre 2 personajes de forma recursiva.	10	
Almacenar en una posición del arreglo el dato correcto.	10	
Recorrer correctamente los arreglos para obtener el dato almacenado en cada posición.	5	
Dibujar personajes.	20	
Obtener atributos del personaje de una estructura.	10	
Almacenar información del personaje en una estructura.	10	
Pintar personajes.	30	
Pintar ojos de personaje con la función Punto.	10	
Pintar los datos de cada personaje con la función Cadena.	10	
Pintar cuerpo de personaje con la función Cuadrado.	10	
Manejo errores.	5	
Detección y recuperación de errores léxicos.	1.5	
Detección y recuperación de errores sintácticos.	1.5	
Detección y recuperación de errores semánticos.	2	
Subtotal (sobre 100)	100	
Total (sobre 18)	18	

Manejo de arreglos de una y múltiples dimensiones

Esta sección se evaluará el correcto funcionamiento de arreglos, todas las salidas deberán mostrar los resultados por la consola de salida de la aplicación.

Los archivos de entrada de esta sección estarán distribuidos de la siguiente manera:

- **Arreglo de una dimensión:** Este archivo de entrada se evaluará la creación e inicialización de arreglos de una sola dimensión, se realizarán inserciones de datos a varios arreglos. Una vez insertados los datos o inicializado los arreglos se realizarán recorridos donde se realizará operaciones aritméticas para obtener la suma, promedio, el número mayor, el número menor. Seguido se hará un ordenamiento burbuja y se realizará la búsqueda binaria de dos valores y se obtendrá los números pares del arreglo origen. Este tendrá una salida grafica de varias filas, de distintos colores.
- **Arreglo de múltiples dimensiones:** Este archivo de entrada se evaluará la creación e inicialización de arreglos de múltiples dimensiones y se realizarán inserciones de datos a

los arreglos. Una vez insertados los datos o inicializados los arreglos se realizarán el recorrido para verificar los datos insertados, seguido por la obtención de la suma y el promedio de los datos por fila. Este generará como salida grafica un cubo.

- **Calculo de DNI:** En este archivo existirá un arreglo de caracteres, donde recibirá de parámetro un número y de este calcula la letra de un DNI y nos devolverá el DNI completo. Este generará como salida grafica el DNI generado.
- **Arreglo de países:** en este archivo existirá un arreglo de cadenas, donde almacenará un país con varias ciudades, seguido realizará un recorrido del arreglo por cada país mostrando sus ciudades.
- **Suma de dimensiones:** En este archivo existirá un arreglo de múltiples dimensiones agregando sus valores directamente en la asignación, seguido realizará la suma de cada una de sus dimensiones por fila y la suma de todas las filas por cada dimensión del arreglo. Este archivo también realizará la suma y el promedio de los valores de forma diagonal y la media de sus valores. Este generará como salida grafica sus resultados obtenidos.

Tiempo estimado para la evaluación de esta sección: **25 minutos.**

DESCRIPCION	Ponderación	Puntaje obtenido
Arreglos de una dimensión	18	
Recorrido	1	
Valores que repite	1	
Suma	1	
Promedio	2	
Número Mayor y Menor	2	
Ordenamiento Burbuja	3	
Búsqueda Binaria	2	
Destino de números pares	2	
Filas (Grafico)	4	
Arreglo de múltiples dimensiones	15	
Recorrido	3	
Suma por filas	4	
Promedio por filas	4	
Cubo (Grafico)	4	
Calculo de DNI	10	
Calculo modular	2	
Generación de número DNI	3	
DNI (Grafico)	5	
Arreglo de países	7	
Cadena de arreglos	3	
Recorrido	4	

Arreglo de personas	15	
Almacenamiento de datos	5	
Recorrido y búsqueda	5	
Nombre y Cuadrado (Grafico)	5	
Sumas de dimensiones	15	
Creación de arreglo	4	
Suma X	4	
Suma W	4	
Grafica de resultados (grafico)	3	
Suma de diagonales	10	
Suma de columna	1	
Suma y promedio de diagonal	3	
Suma y promedio de diagonal inversa	3	
Media de valores	3	
Manejo de Errores léxicos y sintáctico	4	
Manejo de errores semánticos	6	
Error semántico 1	3	
Error semántico 2	3	
Subtotal (sobre 100)	100	
Total (sobre 19)	19	

Manejo intermedio de funcionalidades

- **adjacentElementsProduct**: Esta función recibirá como parámetro un arreglo con varios números, tomará cada número y lo multiplicará con el resto de números dentro del vector, y al final retornará el producto más grande entre todos los números.
- **makeArrayConsecutive2**: Esta función recibirá un arreglo con un conjunto de números aleatorios, el número más grande dentro del arreglo dará el límite máximo, entonces se deberá recorrer el vector y devolver la cantidad de dígitos que faltan en la sucesión. Por ejemplo, si se pasa 2,3,5, entonces retornará que faltan dos dígitos (1,4) para que se complete el total solicitado.
- **shapeArea**: Esta función será una relación de recurrencia, servirá para simular el cálculo de las áreas de alguna figura, se le pasará un número que es el nivel de la figura y retornará el área.
- **matrixElementsSum**: Función que recibirá una matriz de varias dimensiones, y se deberá sumar los números que no estén debajo de un cero (viendo gráficamente la matriz), y retornará la suma de dichos dígitos.
- **allLogestString**: Recibirá un array de cadena de varias longitudes cada cadena, y al final de un análisis deberá retornar un array solo con las cadenas más largas (que tengan la misma longitud)

- **commonCharacterCount:** Recibirá dos cadenas como parámetro, cada cadena con frases distintas y al deberá recorrer las cadenas y retornar la cantidad de caracteres que tienen en común ambas cadenas.
- **IsLucky:** Esta función manejará un conjunto de números, realizará casteos para convertir números a cadenas y luego tendrá que dividir la cadena en dos sub-cadenas, se realizará una comparación entre sub-cadenas, y verificará si ambas sub-cadenas son iguales, si son iguales entonces retornará verdadero.
- **sortByHeight:** Función que recibirá un arreglo con distintos valores de tipo cadena, cada posición tendrá una serie de números enteros combinados con números negativos, el método retornará un vector, el cual tendrá ordenado los números enteros, pero no deberá ordenar los números negativos.
- **ReverseParenthesis:** Esta función recibirá una cadena con un conjunto de letras y paréntesis, la función deberá cambiar el orden de las letras que se encuentren dentro de paréntesis, y al final deberá retornar una cadena con las letras fuera de paréntesis en el mismo orden que ingresaron y las cadenas dentro de paréntesis en un orden cambiado.
- Para las funciones TEST1, TEST2 Y TEST3 lo único que cambiará será el tipo de parámetro que reciba y el tipo de parámetro que retorne, esto para validar que se manejó correctamente el manejo de parámetros y el retorno.
- **Dibujo de una cara:** Este dibujo será un conjunto de pixeles que formarán una cara, se evaluará las funciones nativas como ovalo, línea y cuadrado, para poder dibujar se utilizará funciones recursivas.

Tiempo estimado para la evaluación de esta sección: **25 minutos.**

DESCRIPCION	Ponderación	Puntaje obtenido
Manejo de struct	20	
Creación de struct	5	
Acceso a atributos de struct	5	
Modificación de atributos	5	
GetN	5	
Llamada a funciones	20	
Paso de parámetros	10	
Recursión	10	
Manejo de arrays	20	
adjacentElementsProduct	5	
Test 1	2	
Test 2	2	
Test 3	1	
makeArrayConsecutive2	5	
Test 1	2	
Test 2	2	
Test 3	1	
shapeArea	5	

Test 1	2	
Test 2	2	
Test 3	1	
matrixElementsSum	5	
Test 1	2	
Test 2	2	
Test 3	1	
Manejo de cadenas	25	
allLogestString	5	
Test 1	2	
Test 2	2	
Test 3	1	
commonCharacterCount	5	
Test 1	2	
Test 2	2	
Test 3	1	
isLucky	5	
Test 1	2	
Test 2	2	
Test 3	1	
sortByHeight	5	
Test 1	2	
Test 2	2	
Test 3	1	
reverseParenthesis	5	
Test 1	2	
Test 2	2	
Test 3	1	
Dibujo	15	
Cara flag	15	
Subtotal (sobre 100)	100	
Total (sobre 20)	20	

Manejo avanzado de funcionalidades

En esta sección se evalúa además del correcto funcionamiento del proceso de compilación, la eficiencia y la resistencia a condiciones no ideales que requieren algoritmos de complejidad alta.

1.1 Tipos de recursión

Esta sección permitirá evaluar la implementación correcta de los ámbitos locales utilizando las estructuras en tiempo de ejecución. Una función es recursiva si en su definición existen llamada(s) a sí misma.

1.1.1 Recursión simple (Utilizando operador ternario)

Esta parte **permitirá la evaluación de la sentencia de selección operador ternario.**

1.1.1.1 Función factorial de un numero n

La función factorial está definida para números enteros y es estrictamente creciente, lo cual facilitará la evaluación del manejo de errores en tiempo de ejecución ya que permite fácilmente superar el rango establecido para el tipo de datos entero. Se espera como salida el factorial de n , es decir, $n!$, **permitirá evaluar la recursión simple.**

1.1.1.2 Función potencia

La función potencia calculará la siguiente expresión: x^n , de forma recursiva **permitirá evaluar la recursión simple.**

1.1.2 Recursión múltiple

Una función posee recursividad simple si dentro de su bloque de sentencias existe **una o más** llamadas a sí misma.

1.1.2.1 Solución al problema Torres de Hanoi

Las Torres de Hanói es un juego matemático cuya solución es recursiva, esto **permitirá evaluar la recursión múltiple.**

1.1.2.2 Ordenamiento QuickSort

El ordenamiento QuickSort es un algoritmo de ordenación. Este **permitirá evaluar la recursividad múltiple y uso de arreglos.**

1.1.3 Recursión cruzada o mutua

Dos funciones se llaman mutuamente recursivas si la primera función hace una llamada recursiva a la segunda función y la segunda función, a su vez, llama a la primera.

1.1.3.1 Función generatriz de sucesiones Hofstadter

Esta función posee recursividad cruzada ya que lo cual representa una dependencia circular entre dos funciones, esta **permitirá evaluar la recursión cruzada o mutua**

1.1.3.2 Par o impar

Esta parte será un algoritmo recursivo que determine la paridad de un numero n **permitirá la evaluación de la recursividad cruzada.**

1.1.4 Recursión anidada (Ackerman)

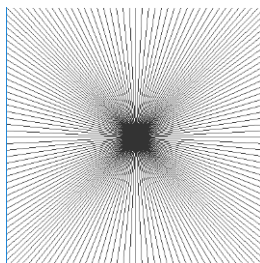
La función de Ackerman es una función es altamente recursiva, que posee dentro de sus argumentos llamadas a sí misma. Esta **permitirá evaluar el rendimiento del compilador desarrollado en el grado de recursión más alto.**

1.2 Gráficos

1.2.1 Patrón de Moire

Es un patrón de interferencia y para simulación del mismo se requerirá la utilización de expresiones complejas, lo cual **permitirá evaluar expresiones aritméticas, lógicas y ciclos de forma más intensiva.**

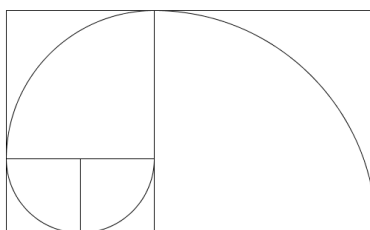
Salida esperada:



1.2.2 Espiral de sucesión de Fibonacci

La sucesión de Fibonacci se define recursivamente y es posible en base a esos números generar la siguiente gráfica, **se evaluará la implementación de las funciones nativas de dibujo en combinación con la recursividad múltiple** de la función Fibonacci.

Salida esperada:

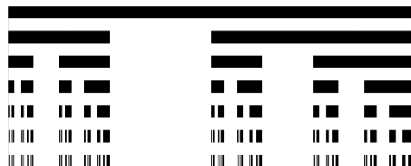


1.2.3 Fractales

La construcción de un fractal¹ puede realizarse de forma iterativa o recursiva de ambas formas la utilización de recursos es significativa, así como el uso de expresiones de complejidad alta, lo cual **permitirá evaluar la correcta implementación de las estructuras en tiempo de ejecución (Heap, Stack, Pila auxiliar) y la ejecución de expresiones complejas.**

1.2.3.1 Conjunto de Cantor

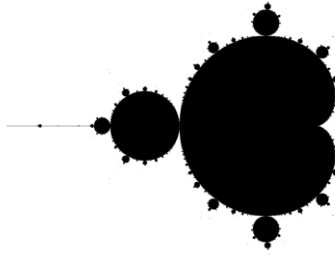
Salida esperada:



¹ Un fractal es un objeto geométrico cuya estructura básica, fragmentada o aparentemente irregular, se repite a diferentes escalas.

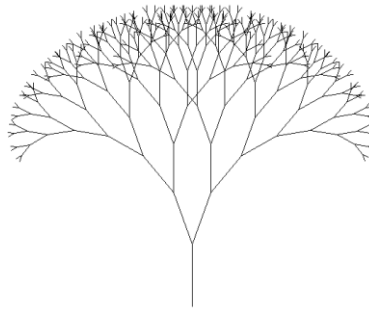
1.2.3.2 Conjunto de Mandelbrot

Salida esperada:

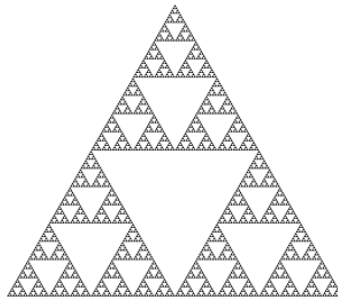


1.2.3.3 Árbol fractal

Salida esperada:



1.2.3.4 Triangulo de Sierpinsky



1.3 Estructuras

En esta parte se evaluarán puntos clave sobre el uso correcto de la estructura **Heap o montículo**, las referencias y las operaciones sobre la misma.

1.3.1 Manejo adecuado de referencias

Se calificará que las direcciones de memoria se hayan trabajado de forma correcta, **las estructuras y cadenas se trabajan por referencia**, el estudiante tendrá acceso a la nota si tomo las consideraciones oportunas, esto para garantizar esta convención.

1.3.2 Funcionalidad avanzada utilizando estructuras

Se evaluará intensivamente el uso de las estructuras y arreglos.

1.4 Errores en tiempo de ejecución

Se **evaluará la detección de un error inducido**, se debe detener la ejecución y mostrar el reporte de error.

1.4.1 Rango numérico superado

Se definieron rangos para los diferentes tipos de datos y para los arreglos de no cumplirse estos rangos deberá reportarse un error.

1.4.2 División por cero

No es posible la división por cero y por eso deberá reportarse un error

1.4.3 Rango definido superado en arreglos

Los arreglos se declaran con ciertas dimensiones y límites de no cumplirse debe reportarse un error.

1.4.4 Errores de tipo

En una asignación el valor de inicialización posee implícito un tipo diferente al declarado debe reportarse un error

Tiempo estimado para la evaluación de esta sección: **35 minutos**.

DESCRIPCION	Ponderación	Puntaje obtenido
Tipos de recursión	30	
Recursividad simple (Utilizando operador ternario)	2	
Función factorial de un numero n	1	
Función potencia	1	
Recursividad múltiple	4	
Solución al problema Torres de Hanói	2	
Ordenamiento QuickSort	2	
Recursividad cruzada	9	
Función generatriz de sucesiones Hofstadter	5	
Par o impar	4	
Recursividad anidada	15	
Ackerman	15	
Gráficos	35	
Patrón de Moiré	5	
Espiral de sucesión de Fibonacci	5	
Fractales	25	
Conjunto de Cantor	5	
Conjunto de Mandelbrot	10	
Árbol fractal	5	
Triangulo de Sierpinsky	5	
Estructuras	20	
Manejo adecuado de referencias	10	

Funcionalidad avanzada utilizando estructuras	10	
Errores en tiempo de ejecución	15	
Rango numérico superado (función factorial)	5	
División por cero	2	
Rango definido superado en arreglos	4	
Errores de tipo	4	
Subtotal (sobre 100)	100	
Total (sobre 25)	25	

Restricciones:

- La calificación se realizará sobre archivos ejecutables.
- El proyecto es individual.
- La calificación del proyecto se realizará presencialmente.
- No se puede agregar o quitar algún símbolo en el archivo de entrada. El proyecto deberá funcionar con los archivos que sean proveídos por lo auxiliares para la calificación, sin modificación alguna.
- No será permitido compartir los archivos de entrada durante ni después de la calificación.
- La calificación del proyecto será personal y durará como máximo 120 minutos. Se debe tomar en cuenta que no pueden estar personas ajenas a la calificación, de lo contrario no se calificará el proyecto.
- Anomalías o Copias detectadas de proyectos tendrán de manera automática una nota de 0 puntos y los involucrados serán reportados a la Escuela de Ingeniería en Ciencias y Sistemas, para que se apliquen las sanciones correspondientes.
- Anomalías detectadas en los archivos entregables tendrá de manera automática una nota de 0 puntos, por ejemplo: no se envió código el código correcto, se envió parte del código y no el código completo, archivos ajenos a los entregables del proyecto, no se hizo uso de las herramientas descritas en el enunciado de cada proyecto, entre otras.
- Los archivos de entrada contendrán errores semánticos, sintácticos y léxicos para la verificación de recuperación de errores de la aplicación.
- El estudiante deberá generar su código intermedio y ejecutar el código generado para tener derecho a calificación.
- **Si el incumplimiento de alguna de las anteriores restricciones o bien alguna anomalía detectada durante la calificación los tutores académicos finalizaran la calificación y anularan la nota obtenida hasta el momento.**

Observaciones:

En este apartado se podrá agregar todas las observaciones encontradas en la aplicación y calificación del estudiante.

Observación	Descripción
Se finalizó la calificación por tiempo excedido	Hora inicial: Hora final: Tiempo utilizado: ____/120 minutos Detalles:

Se detectó copia y se anuló entrega	Detalles:
El estudiante modifico código durante la calificación y se anuló entrega.	Detalles:
El estudiante se sorprendió utilizando ejecutables diferentes a los entregados.	Detalles:
Otros:	

Estoy conforme con la nota obtenida y acepto como correcta la nota asignada en cada uno de los aspectos evaluados:

Nota obtenida:	
-----------------------	--

José Cano
Tutor Académico, sección A

Daniel Álvarez
Tutor Académico, sección B+

Mike Talavera
Tutor Académico, sección A

Javier Navarro
Tutor Académico, sección B-

Estudiante