PLAN DE TRABAJO

I. Datos de la institución

Plantel	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN DIVISIÓN SISTEMA UNIVERSIDAD ABIERTA Y EDUCACIÓN A DISTANCIA Modalidad: A Distancia		**Grado o Licenciatura**	Licenciatura en Informática

II. Datos del asesor

Nombre	OJEDA VILLAGOMEZ RAUL	**Correo**	rojeda@docencia.fca.unam.mx

III. Datos de la asignatura

Nombre	INTRODUCCION A LA PROGRAMACION	**Clave**	1167	**Grupo**	8291
Modalidad	Obligatoria	**Plan**	2012	**Fecha de inicio del semestre**	12 de febrero de 2024
Horas de asesoria semanal	4	**Horario**	Lunes: 09:00 - 11:00 hrs Miércoles: 09:00 - 11:00 hrs	**Fecha de término del semestre**	19 de junio de 2024

IV. Contenido temático

TEMA	**HORAS**		
	Total	**Teoría**	**Práctica**
I. Introducción a la programación	4	4	0

II. Tipos de datos elementales (Variables, Constantes, declaraciones y expresiones y estructura de un programa.)	6	6	0
III. Control de flujo	14	14	0
IV. Funciones	18	18	0
V. Tipos de datos Compuestos (Estructuras)	14	14	0
VI. Manejo de apuntadores	8	8	0

V. Presentación general del programa

En específico, el estudiante trabajará con una guía didáctica elaborada por docentes de la Facultad, con bibliografía adicional que le permitirá ir a las fuentes de información originales.
Esta ayuda comprende referencias esenciales sobre los temas y subtemas de cada unidad de la asignatura, y propicia que el estudiante se introduzca en el aprendizaje de los mismos, asimilándolos de lo concreto a lo abstracto y de lo sencillo a lo complejo, por medio de ejemplos, ejercicios y casos, u otras actividades que le permitan aplicarlos y vincularlos con la realidad laboral. Es decir, lo induce al "saber teórico" y al "saber hacer" de la asignatura, y lo conduce a encontrar respuestas a preguntas reflexivas que se formule acerca de los contenidos, su relación con otras asignaturas, utilidad y aplicación en el trabajo. Finalmente, pretende dotar al estudiante de la información suficiente para que pueda autoevaluarse sobre el conocimiento básico de una asignatura, motivarse a profundizarlo y ampliarlo con otras fuentes bibliográficas, y prepare adecuadamente su examen, el cual tiene un alto grado de dificultad.
Las sesiones en videoconferencia (Zoom) son:
Sesión 1: Lunes 19 de febrero 9:00 a 11:00 horas / Unidad 1. Introducción a la programación / Dudas
Sesión 2: Lunes 04 de marzo 9:00 a 11:00 horas / Unidad 2. Tipos de datos elementales Dudas

Sesión 3: Lunes 01 de abril 9:00 a 11:00 horas / Unidad 3. Control de Flujo / Dudas
Sesión 4: Lunes 15 de abril 9:00 a 11:00 horas / Unidad 4. Funciones Dudas

Sesión 5: Lunes 06 de mayo 9:00 a 11:00 horas / Unidad 5. Tipos de datos compuestos Dudas
Sesión 6: Lunes 03 de junio 9:00 a 11:00 horas / Unidad 6. Manejo de Apuntadores / Dudas

VI. Forma en que el alumno deberá preparar la asignatura

El alumno tiene la obligación de realizar los cuestionarios al finalizar cada unidad como reforzamiento aunque no se evalué, así como revisar y analizar todo el material de la plataforma, esto le permitirá un mejor desempeño en el examen final. Es deber del alumno tener sentido ético, y no copiar de internet los programas, copiar de internet los programas propicia una buena nota en esa actividad, pero va en detrimento del estudiante y no permite que el asesor le indique cuales son los puntos de oportunidad que pudiera tener el alumno, en consecuencia el examen tendría un grado de dificultad muy grande para aquellos alumnos que no se esfuercen por realizar cada programa que se tienen que entregar por unidad, con un total de 12 programas durante todo el
semestre. TODOS LOS PROGRAMAS SE DEBEN ENTREGAR CON CÓDIGO FUENTE EN LENGUAJE DE PROGRAMACIÓN "C", NO EL EJECUTABLE, Y EL ALUMNO DEBERÁ PROBAR, QUE CORRA ADECUADAMENTE PREFERENTEMENTE EN UN COMPILADOR EN LINEA, el hecho de que le falte alguna librería y no corra el programa implica CERO en dicha actividad, SI LA ACTIVIDAD SE ENTREGA EN FORMATO "PDF", la escala sera sobre 9, en lugar de 10.

Se acepta Lenguaje "C", "C++", y Python, sin embargo el examen es en "C", ya que Python no aceptan apuntadores y es tema del capítulo 6. Planear las actividades con base al plan de trabajo de
la asignatura, con la finalidad de que el alumno auto regule su aprendizaje mediante las siguientes acciones:

- Establecer un horario de trabajo escolar.
- Desarrollar hábitos de estudio.
- Asignar espacios adecuados para el estudio.
- Realizar búsqueda de información alterna que propicie análisis y reflexión.
- Seleccionar las estrategias de aprendizaje que le faciliten la adquisición, comprensión y utilización de información (apropiarse del conocimiento).
- Realizar autoevaluaciones.
- Formular dudas concretas para promover el diálogo y la discusión con su asesor y tomar decisiones.

Programas a entregar

No. Programa	unidad	Act. Complementaria	Programa
1	Unidad 1	1	Ordenar
2	Unidad 1	2	GAP
3	Unidad 2	1	Frecuencias
4	Unidad 2	2	Validar CURP
5	Unidad 3	1	Chicos y Grandes
6	Unidad 3	2	Menu Ordenar y Buscar
7	Unidad 4	1	MCD
8	Unidad 4	2	Combinaciones
9	Unidad 5	1	Conjuntos
10	Unidad 5	2	Cuadrado Mágico
11	Unidad 6	1	Buscar Personas
12	Unidad 6	2	VPN

CALENDARIO DE ACTIVIDADES

Fecha	No. Unidad	No. Actividad	Descripción de la de actividad de acuerdo a la plataforma	Ponderación
28 de febrero de 2024	UNIDAD 1: Introducción a la programación	Act. inicial	Presentarse en el foro indicando los siguientes puntos: 1. Nombre, e indicar como prefieren que lo llamen 2. En donde trabaja 3. Expectativas de la materia y de la carrera 4. A que dedica el tiempo libre (pasatiempos favoritos) 5. Frase célebre favorita o con la que se identifica	0 %

28 de febrero de 2024	UNIDAD 1: Introducción a la programación	Act. complementaria 1	Actividad Complementaria 1 de la Unidad 1
			/***
			El siguiente programa lista 100 números aleatorios, tu actividad es que lo corrijas para que en la salida estén con los números ordenados de mayor a menor
			***/
			#include <iostream>
			#include <stdlib.h>
			#include <time.h>
			using namespace std;
			int main() {
			srand(time(0));
			int numeros[100] ={};
			for (int i=0; i<100; i++) {
			numeros[i] = (rand() % 1000)+1;
			}
			for (int i=0; i<=100; i++) {
			cout <<"Consecutivo " << i << " valor " << numeros[i] << endl;
			}
			cout << "Termina programa " << endl;
			}
				6 %

04 de marzo de 2024	UNIDAD 1: Introducción a la programación	Act. complementaria 2	Realiza un programa que genere de manera aleatoria 80 números en el intervalo del conjunto del 1 al 100, que no se repitan, y después genera una lista con los 20 números que hagan falta, ejemplo con un conjunto de 10 números generando 8 al azar: {1,2,4,5,6,7,9,10} la lista con los que faltan son 3 y 8. En términos computacionales este concepto se conoce como “GAP”, y es utilizado como una brecha, se utiliza en auditoria para detectar los folios de documentos que hagan falta. Nota: Ambos conjuntos tanto el de 80 números como el de 20, deben estar ordenados de menor a mayor	6 %
11 de marzo de 2024	UNIDAD 2: Tipos de datos elementales (Variables, Constantes, declaraciones y expresiones y estructura de un programa.)	Act. complementaria 1	Actividad Complementaria 1 de la Unidad 2 Programa de Frecuencias: Frecuencia es una repetición de un hecho o un suceso. Es también el número de veces que se repite un proceso periódico en un intervalo de tiempo determinado. Número de oscilaciones, vibraciones u ondas por unidad de tiempo en cualquier fenómeno periódico. Realiza un programa que genere de manera aleatoria 100 números en el intervalo del conjunto del 1 al 20, y después genera una lista con la frecuencia de cada número. Deberas mostrar tanto los 100 números aleatorios en orden de menor a mayor, como la lista de frecuencias tambien en orden.	6 %

<table>
<tr>
<td>01 de abril de 2024</td>
<td>UNIDAD 2: Tipos de datos elementales (Variables, Constantes, declaraciones y expresiones y estructura de un programa.)</td>
<td>Act. complementaria 2</td>
<td>Programa que valida el campo CURP
El siguiente programa valida si una CURP esta ingresada de manera correcta en cada uno de sus caracteres:
#include <iostream>
#include <iterator>
#include <string>
#include <regex>
int main()
{
std::string s = "OEVR670103HDFJLL08";
std::regex patron("^(([A-Z]{1}[AEIOUX]{1}[AZ]{
2}[0-9]{2}(0[1-9]|1[0-2])(0[1-9]|1[0-9]|2[0-9]|3[0-1])[HM]{1}(AS|BC|BS|CC|CS|CH|CL|CM|DF|DG|GT|GR|HG|JC|
MC|MN|MS|
NT|NL|OC|PL|QT|QR|SP|SL|SR|TC|TS|TL|VZ|YN|ZS|NE)[B-DF-HJ-NP-TV-Z]{3}[0-9A-Z]{1}[0-9]{1}))$");
if (std::regex_match(s, patron))
std::cout << "La CURP es valilda '\n";
else
std::cout << "La CURP NO es valilda '\n";
if (std::regex_match("OTVR670103HDFJLL08", patron))
std::cout << "La CURP 'regular expressions'\n";
else
std::cout << "La CURP OTVR 'NO TIENE una regular expressions'\n";
if (regex_match("OAVR674003HDFJLL08", patron))
std::cout << "La CURP 'regular expressions'\n";
else
std::cout << "La CURP OAVR674003HDFJLL08 'NO TIENE una regular expressions'\n";
}
------Lo deberas modificar para que al momento de ingresar una CURP que no es valida indique porque no es valida, en cada uno de sus posiciones:

<table><tr><td>Posición</td><td>Normas generales para la construcción de la clave</td></tr><tr><td>Posición 1-4</td><td>La letra inicial y la primera vocal interna del primer apellido, la letra inicial del segundo apellido y la primera letra del nombre. En el caso de las mujeres casadas, se deberán usar los apellidos de soltera (alfabetica)</td></tr><tr><td>Posición 5-10</td><td>La fecha de nacimiento en el orden de año, mes y dia. Para el año se tomarán los dos últimos digitos, cuando el mes o el dia sea menor a diez, se antepondrà un cero.
1 de diciembre de 1995, Quedaria: 951201 (numèrica)</td></tr><tr><td>Posición 11</td><td>Sexo M para mujer y H para hombre (alfabetica)</td></tr><tr><td>Posición 12-13</td><td>La letra inicial y última consonante, nacimiento conforme al Catálogo (SEGOB) de Entidades Federativas (alfabética) Ver el script</td></tr><tr><td>Posición 14-16</td><td>Integradas por las primeras consonantes internas del primer apellido, segundo apellido y nombre (alfabética).</td></tr><tr><td>Posición 17</td><td>Diferenciador de homonimia y siglo, caracter progresivo asignado por la Secretaria de Gobernación que se emplea para diferenciar registros homónimos, 1-9 para fechas de nacimiento hasta el año 1999 y A-Z para fechas de nacimiento a partir de 2000 (alfanumérica).</td></tr><tr><td>Posición 18</td><td>Digito verificador, caracter asignado por la Secretaria de Gobernación a través de la aplicación de un algoritmo que permite calcular y verificar la correcta conformación y transcripción de la clave.</td></tr></table></td>
<td>6 %</td>
</tr>
</table>

08 de abril de 2024	UNIDAD 3: Control de flujo	Act. complementaria 1	Actividad Complementaria 1 de la Unidad 3 El juego de dados conocido como “Chicos y Grandes” es popular, realice un programa que simule dicho juego, a continuación se muestran las reglas para los jugadores. 1. Un jugador tira dos dados. Cada dato tiene seis caras. Las caras contienen 1, 2, 3, 4, 5 y 6 puntos. 2. Antes de tirar los dados el jugador debe elegir entre Chicos o Grandes 3. Una vez que los dados se hayan detenido, se calcula la suma de los puntos en las dos caras superiores. El rango de la suma es de 2 a 12 4. • Si a la primera tirada, la suma es inferior a 7, el jugador que eligió Chicos gana. 5. • Si a la primera tirada la suma es superior a 7 el jugador que eligio Grandes gana. 6. • Si a la primera tirada la suma es 7, entonces la casa gana Repetir el juego hasta que el jugador lo decida.	6 %

15 de abril de 2024	UNIDAD 3: Control de flujo	Act. complementaria 2	Actividad Complementaria 2 de la Unidad 3
			6. Realice un programa que realice las funciones que a continuación se indican. Todas las opciones deben ser presentadas al usuario a través de un menú de opciones.
			• Escriba una función en la que se generen de manera aleatoria 100 números enteros (entre 1 y 1000) y guardarlos en un arreglo
			• Escriba una función que ordene los numeros que se generaron en la opción 1 y guardelos en otro arreglo
			• Escriba una función que busque un número en el arreglo que se generaron los números aleatorios, de forma secuencial
			• Utilice la función "busquedaBinaria" que realice una busqueda binaria en el arreglo ordenado
			Nota: El programa que se muestra a continuación corre de manera adecuada, sólo tiene que modificarlo para que en lugar de tomar valores fijos, tome valores aleatorios.
			
			`#include <stdio.h>`
			`// Función para realizar la búsqueda binaria`
			`int busquedaBinaria(int arr[], int bajo, int alto, int objetivo) {`
			`  while (bajo <= alto) {`
			`  int medio = bajo + (alto - bajo) / 2;`
			`  // Mostrar puntos de control`
			`  printf("Punto de control: [bajo = %d, alto = %d, medio = %d]\n", bajo, alto, medio);`
			`  // Verificar si el objetivo está en la mitad`
			`  if (arr[medio] == objetivo) {`
			`  return medio; // Se encontró el objetivo`
			`  } else if (arr[medio] < objetivo) {`
			`  bajo = medio + 1; // Descartar la mitad izquierda`
			`  } else {`
			`  alto = medio - 1; // Descartar la mitad derecha`
			`  }`
			`  }`
			`  return -1; // No se encontró el objetivo`
			`}`
			`int main() {`
			`  int numeros[100];`
			
			`  // Inicializar el arreglo con números del 1 al 100`
			`  for (int i = 0; i < 100; ++i) {`
			`  numeros[i] = i + 1;`
			`  }`
			`  int objetivo = 35;`
			`  // Llamar a la búsqueda binaria`
			`  int indice = busquedaBinaria(numeros, 0, 99, objetivo);`
			`  // Mostrar el resultado`
			`  if (indice != -1) {`
			`  printf("El valor %d se encuentra en la posición %d.\n", objetivo, indice + 1);`
			`  } else {`
			`  printf("El valor %d no se encuentra en el conjunto.\n", objetivo);`
			`  }`
			`  return 0;`
			`}`
				6 %

<table>
<tr><td>22 de abril de 2024</td><td>UNIDAD 4: Funciones</td><td>Act. complementaria 1</td><td>Actividad Complementaria 1 de la Unidad 4
Función Recursiva para el Máximo Común Divisor (MCD): Escribe una función denominada 'mcd 'que devuelve el máximo común divisor de dos enteros, x e y. El Máximo Común Divisor es el número más grande que puede dividir completamente tanto a x como a y.
La definición recursiva del MCD es la siguiente: Si y es igual a 0, entonces mcd(x,y) es x. En caso contrario, mcd(x,y) es igual a mcd(y,x%y). En otras palabras, la función se llama a sí misma con los argumentos y y x%y hasta que y sea igual a 0, momento en el cual el MCD es x. Este proceso se basa en la propiedad matemática de que el MCD de dos números no cambia si reemplazamos uno de los números por su residuo cuando se divide por el otro.</td><td>6 %</td></tr>
<tr><td>29 de abril de 2024</td><td>UNIDAD 4: Funciones</td><td>Act. complementaria 2</td><td>Actividad Complementaria 2 de la Unidad 4
Escriba un programa que despliegue todas las combinaciones de 5 archivos en tres carpetas, el total de combinaciones son 243, vease libros de combinatoria, como ejemplo se muestra las diez primeras combinaciones de las 243, donde C1, representa la carpeta 1, C2, la carpeta 2, y C3 la carpeta 3, y los archivos estan representados del 1 al 5:
<table><tr><th>No</th><th>C1</th><th>C2</th><th>C3</th><th>Representación</th></tr><tr><td>1</td><td>54321</td><td></td><td></td><td>54321 / /</td></tr><tr><td>2</td><td>5432</td><td>1</td><td></td><td>5432 / 1 /</td></tr><tr><td>3</td><td>5432</td><td></td><td>1</td><td>5432 / / 1</td></tr><tr><td>4</td><td>5431</td><td>2</td><td></td><td>5431 / 2 /</td></tr><tr><td>5</td><td>543</td><td>21</td><td></td><td>543 / 21 /</td></tr><tr><td>6</td><td>543</td><td>2</td><td>1</td><td>543 / 2 / 1</td></tr><tr><td>7</td><td>5431</td><td></td><td>2</td><td>5431 / / 2</td></tr><tr><td>8</td><td>543</td><td>1</td><td>2</td><td>543 / 1 / 2</td></tr><tr><td>9</td><td>543</td><td></td><td>21</td><td>543 / / 21</td></tr><tr><td>10</td><td>5421</td><td>3</td><td></td><td>5421 / 3 /</td></tr></table>
Nota:
la primer columna es un consecutivo que debe representar los 243 casos las columnas 2,3 y 4 representan las carpetas y cada uno de los archivos que estan en estas carpetas, no se deben de repetir los números, sólo pueden estar del 1 al 5 en las tres carpetas la columna 5,, representa la separación de archivos por carpetas la diagonal "/"
Restricciones: Debe tener funciones recursivas.
La columna de representación sirve para identificar de manera individual cada combinación por lo que ninguna se debe de repetir y debe estar dentro del programa la función que valide que ninguna combinación se repita</td><td>6 %</td></tr>
<tr><td>06 de mayo de 2024</td><td>UNIDAD 5: Tipos de datos Compuestos (Estructuras)</td><td>Act. complementaria 1</td><td>Actividad Complementaria 1 de la Unidad 5
Programa que despliega subconjuntos.
Escriba un programa que despliegue todos los subconjuntos no vacíos del conjunto de los números del 1 a n. Por ejemplo,si n = 3, debe obtenerse:
123
1 2
1 3
2 3
1 2 3
Restricciones: Debe tener funciones recursivas</td><td>6 %</td></tr>
<tr><td>20 de mayo de 2024</td><td>UNIDAD 5: Tipos de datos Compuestos (Estructuras)</td><td>Act. complementaria 2</td><td>Actividad Complementaria 2 de la Unidad 5
Escriba un programa que encuentre el cuadrado mágico para cualquier n donde 3< n< 10 de orden impar.
Un cuadrado mágico es una cuadrícula en general de n x n, en la que se acomodan ciertos números que cumplen que la suma de cualquier renglón, la suma de cualquier columna y la suma de cualquiera de cualquiera de las dos diagonales es siempre la misma.
Si el cuadrado es de 3 x 3, entonces tendrá 9 casillas y los números que se acomodan en él son todos los números del 1 al 9.
Si el cuadrado es de 4 x 4, entonces tendrá 16 casillas y los números que se acomodan en él son del 1 al 16.
En general, si el cuadrado es de n x n, entonces tendrá n cuadrada casillas y los números que acomodaremos en él serán del 1 a n².
El orden de un cuadrado mágico es el número de renglones o el número de columnas que tiene. Así un cuadrado de 3 x 3 se dice que es de orden 3.</td><td>6 %</td></tr>
</table>

27 de mayo de 2024	UNIDAD 6: Manejo de apuntadores	Act. complementaria 1	Se cuenta con la base de datos del estado de Aguascalientes con fecha 2021 del mes de abril, y se desea buscar el sueldo del Sr. Humberto Ramos Chavez; Por lo que tendras que realizar un programa que cargue el archivo en Memoria y realice dos busquedas por el mismo ID: 276638993, a) Una busqueda secuencial, anotar el número de condicionantes antes de llegar al resultado, junto con el tiempo que se tarda en dar la respuesta. b) Una busqueda binaria, anotar el número de condicionantes antes de llegar al resultado, junto con el tiempo que se tarda en dar la respuesta. Si realizas de manera adecuada la busqueda binaria te daras cuenta que el resultado es "el logaritmo base 2 de n", es decir Log2 n; o lo que es lo mismo n=2^x; por lo tanto si tomamos esta base de datos con 34595, 34595=2^x; donde x=15.0782759, por lo tanto no deberas ocupar más de 16 condiciones para encontrar cualquier persona dentro de esta base de datos. En el mismo programa deberas dejar un ciclo para realizar cualquier otra busqueda binaria por cualquier ID. La base de datos puedes descargarla de la Liga: https://oficinavirtual.issste.gob.mx/Transparencia/Padr%C3%B3n-de-Derechohabientes-Activos# En caso de que no puedas descargarla del portal de transparencia la dejo disponible en: https://drive.google.com/file/d/1mjek_YABxkLiqgytHXXlrq873sZnQ4Jd/view?usp=sharing	5 %
03 de junio de 2024	UNIDAD 6: Manejo de apuntadores	Act. complementaria 2	Actividad Complementaria 2 de la Unidad 6 Crear un programa para calcular el Valor Presente Neto (VPN) se trae todos los flujos de caja al momento presente descontándolos a un tipo de interés determinado. Por tanto, cada entrada y salida de efectivo se descuenta a su valor presente (VP). Luego se suman, tal y como establece la siguiente fórmula: VAN = Io + F1/(1+k) + F2/(1+k)^2 +...+Fn/(1+k)^n La fórmula del Valor Presente Neto o VAN depende, por tanto, de las siguientes variables: • Inversión inicial previa (Io): es el monto o valor del desembolso que la empresa hará en el momento inicial de efectuar la inversión. • Flujos netos de efectivo (Ft): representan la diferencia entre los ingresos y gastos que podrán obtenerse por la ejecución de un proyecto de inversión durante su vida útil. • Tasa de descuento (k): también conocida como costo o tasa de oportunidad. es la tasa de retorno requerida sobre una inversión. Refleja la oportunidad perdida de gastar o invertir en el presente. • Inversiones durante la operación. • Número de periodos que dure el proyecto (n). Una formula directa es: =F*((1-(1/(1+k)^n)))/k	5 %

VII. Sistema de evaluación

FACTORES	DESCRIPCIÓN

Requisitos	Presentarse en el foro indicando los siguientes puntos: 1. Nombre, e indicar como prefieren que lo llamen 2. En donde trabaja 3. Expectativas de la materia y de la carrera 4. A que dedica el tiempo libre (pasatiempos favoritos) 5. Frase célebre favorita o con la que se identifica Descargar un compilador En el caso de C++ hay multitud de entornos (a veces llamados simplemente compiladores), algunos interesantes y recomendados son: - Turbo C/C++. Gratis desde el enlace http://community.borland.com/article/images/21751/tcpp101.zip - GCC. Compilador C++ con licencia GNU (de distribución gratuita y código abierto) http://gcc.gnu.org> - Dev C/C++. Uno de los mejores entornos de programación en C o C++. De libre distribución y que posee un amigable entorno gráfico. http://www.bloodshed.net/devcpp.html> Compiladores en línea recomendados para C y Python: https://www.onlinegdb.com/online_c_compiler https://ideone.com/ http://www.codepad.org/ http://cpp.sh/ Recomendado para Android: CppDroid // desde Google play REQUISITOS Haber acredito al menos 8 actividades, es decir haber entregado cuando menos 8 de los programas, de los 12 programas con calificación aprobatoria, el hecho de que le falte alguna librería y no corra el programa implica CERO en dicha actividad. - Debido a que dedico un tiempo determinado para revisar las entregas, cualquier actividad entregada después de la fecha límite no recibirá retroalimentación y la calificación dependerá del tiempo de retraso. - La fecha limite para entregar programas fuera de tiempo, es un día antes de que comience la semana de examen final - Por cada semana de retraso será 1 punto menos en tu calificación de la actividad. - Recuerda que en tu examen final, debes estudiar el 100% del temario de la asignatura y sólo tienes un intento con un tiempo límite de aplicación que al terminarse se cerrará automáticamente, enviando la calificación obtenida hasta el momento. - Cuando la realización de una actividad implique hacer una investigación, deberás buscar fuentes oficiales, como libros, revistas, artículos,

Porcentajes		
	Examen Final	30 %
	Act. complementaria	70 %
	TOTAL	100 %

La calificación final de la asignatura está en función de la ponderación del asesor, no de la que se visualiza en la plataforma. Es necesario solicitar por correo electónico la calificación final al asesor.

VIII. Recursos y estratégias didácticas

Elaboración de Actividades de Aprendizaje	(X)
Software Específico	(X)
Programación Computacional	(X)
Plataforma Educativa	(X)

Foro Electrónico	(X)
Chat	(X)
Correo Electrónico	(X)
Sitios de Internet	(X)
Plan de Trabajo	(X)