**Instituto Nacional de Aprendizaje**

**Programador de Aplicaciones Informáticas**

**Módulo Lógica Computacional**

**Portafolio de Evidencias**

**Estudiante: Mariel Daniela Rojas Sanchez**

**Cedula: 208030487**

**Profesor: Oscar Pacheco Vásquez**

**17 Agosto 2022**

**Índice**

[Introducción: 3](#_Toc111656766)

[Objetivo General: 4](#_Toc111656767)

[Objetivo Específico: 4](#_Toc111656768)

[Descripción de las actividades realizadas: 5](#_Toc111656769)

[Investigación y Exposición #1 5](#_Toc111656770)

[Práctica #1 de Razonamiento Lógico 5](#_Toc111656771)

[Práctica #2 Metodología de Soluciones de Problemas 6](#_Toc111656772)

[Práctica #3 Diseño de Algoritmos 8](#_Toc111656773)

[Práctica #4 Algoritmos en Pseudocódigo. 10](#_Toc111656774)

[Práctica #5 MSP y Algoritmos en Pseudocódigo. 11](#_Toc111656775)

[Práctica #6 MSP y Algoritmos en Pseudocódigo. 13](#_Toc111656776)

[Trabajo cotidiano MSP. 14](#_Toc111656777)

[Práctica #7 MSP y Algoritmos en Pseudocódigo. 14](#_Toc111656778)

[Práctica #8 MSP y Algoritmos en Pseudocódigo. 15](#_Toc111656779)

[Práctica #9 MSP y Algoritmos en Pseudocódigo. 15](#_Toc111656780)

[Práctica #10 MSP y Algoritmos en Pseudocódigo. 16](#_Toc111656781)

[Práctica #11 MSP y Algoritmos en Pseudocódigo. 17](#_Toc111656782)

[Glosario: 24](#_Toc111656783)

[Conclusiones: 32](#_Toc111656784)

[Bibliografía: 32](#_Toc111656785)

# Introducción:

Este portafolio de evidencias tiene como fin el ejemplificar los temas y practicas realizadas en el proceso de este primer Módulo de Lógica Computacional, en el cual podemos encontrar diferentes actividades con sus específicas descripciones, observaciones, resultados y a su vez un glosario de términos necesarios para entender correctamente los ejercicios y definiciones.

Primeramente, encontramos la descripción de las actividades realizadas, en diferentes fechas a lo largo del primer módulo, esta respectiva descripción especifica brevemente de que trataba dicha actividad, así como sus resultados y algunas observaciones importantes; Por ultimo se encuentra un glosario, el cual contiene todos de los términos de suma importancia para poder llevar a cabo la realización del módulo. Estas actividades nos proporcionan la practica para poner en uso el conocimiento adquirido en clase.

Con este portafolio se realiza un resumen de conceptos importantes y observaciones que nos permiten aprender, repasar y mejorar todos los conceptos significativos vistos a lo largo de este primer módulo. Permitiendo tener en mano lo necesario para cumplir de manera eficaz los objetivos del módulo, ya que nos permite brindar soluciones a los posibles problemas a resolver.

# Objetivo General:

Elaborar un portafolio de evidencias de trabajos realizados, recopilando todas las actividades y conceptos vistos a lo largo del módulo de Lógica Computacional

# Objetivo Específico:

Repasar, practicar conceptos, definiciones, resultados y observaciones brindadas por el profesor y vistas en clase.

# Descripción de las actividades realizadas:

|  |  |
| --- | --- |
| Investigación y Exposición #1 | |
| Fecha: | Lunes 13 junio 2022 |
| Descripción: | Se investiga sobre la Maquina Enigma y se realiza una exposición de al menos 20 minutos, donde se explica sobre el tema asignado. |
| Resultado: |  |
| Observaciones: | Esta evaluación se realizó en grupo. Por lo que se dividió el trabajo entre los participantes del grupo. En mi opinión todos trabajamos muy bien y equitativamente. |

|  |  |
| --- | --- |
| Práctica #1 de Razonamiento Lógico | |
| Fecha: | Miércoles 8 junio 2022 |
| Descripción: | Se realiza un ejercicio corto, de razonamiento lógico donde según una serie de indicaciones se debe obtener un resultado haciendo el método de descarte. |
| Resultado: | El día siguiente el profesor realiza la revisión de este ejercicio en clase y mis resultados estaban de manera correcta. |
| Observaciones: | Este ejercicio me pareció muy interesante, ya que hay que ir analizando detenidamente y descartando las diferentes opciones para llegar a el resultado correcto. |

|  |  |
| --- | --- |
| Práctica #2 Metodología de Soluciones de Problemas | |
| Fecha: | Viernes 10 junio 2022 |
| Descripción: | Se resuelven 5 ejercicios mediante la metodología de solución de problemas. Para cada ejercicio es necesario realizar un planteamiento de problema, confinando las entradas y salidas, así como las observaciones o limitaciones, seguidamente se identifican dos o más alternativas de solución, para al final hacer la elección de una de estas alternativas. |
| Resultado: | El profesor realiza la revisión de esta práctica en clase con ejemplos de los ejercicios de algunos de los estudiantes, dando una retroalimentación en conjunto.  Para esta práctica, en mi caso fue necesario realizar varias correcciones, como por ejemplo de algunos datos de salida de uno de los ejercicios.  El profesor dio la indicación de poder realizar correcciones durante el fin de semana y entregar el trabajo corregido el domingo 12 de junio. |
| Observaciones: | Como era la primera vez realizando la plantilla de MSP, se me dificulto un poco, sin embargo, con la retroalimentación del profesor en clase pude comprender un poco mejor la realización de esta. |

|  |  |
| --- | --- |
| Práctica #3 Diseño de Algoritmos | |
| Fecha: | Miércoles 15 de junio 2022 |
| Descripción: | Se analizan 9 ejercicios mediante los primeros tres pasos de la metodología de solución de problemas, por lo que se deben definir las entradas, salidas, se presenta dos alternativas de solución para cada uno de los 9 ejercicios y, por último, se hace la elección de una de las alternativas mediante una justificación. |
| Resultado: | Se deben hacer correcciones en las alternativas del ejercicio 1 y 8; En el ejercicio 1 se debe modificar el cálculo del diámetro y en el ejercicio 8 se debe corregir la formula la variable “lado” en la formula del perímetro. |
| Observaciones: | Se observa que la calificación estuvo en un rango adecuando, sin embargo, hay varios detalles importantes que se deben tener en cuenta a la hora de resolver la plantilla, como por ejemplo tratar de siempre hacer observaciones, limitaciones, para evidenciar algunos valores que se van a usar en el algoritmo, esto para tener una mejor claridad. También es necesario únicamente usar las variables declaradas, no poner variables que no se van a utilizar para hacer el algoritmo más rápido y eficaz. |

|  |  |
| --- | --- |
| Práctica #4 Algoritmos en Pseudocódigo. | |
| Fecha: | Jueves 16 de junio 2022 |
| Descripción: | Se crean algoritmos funcionales en pseudocódigo que resuelvan cada uno de los 9 ejercicios planteados, a los cuales anteriormente ya se les había aplicado la metodología de solución de problemas. |
| Resultado: | El ejercicio 4 posteriormente el profesor lo resolvió en clase para tener una mejor claridad, ya que para este ejercicio era necesario tomar en cuenta muchos detalles importantes. |
| Observaciones: | El ejercicio 4 me pareció muy interesante pero muy complicado de resolver, en mi caso no pude resolverlo del todo ni para la entrega de la practica ni después de esta, sin embargo, cuando el profesor lo resolvió en clase me quedaron claros muchos puntos importantes que yo no estaba tomando en cuenta.  Para el problema 6 se debió tomar en cuenta el cero para el cálculo de los números pares, ya que el cero se toma como un numero par. |

|  |  |
| --- | --- |
| Práctica #5 MSP y Algoritmos en Pseudocódigo. | |
| Fecha: | Lunes 20 de junio 2022 |
| Descripción: | Para esta práctica se analizaron 4 ejercicios diferentes, a los cuales se les debía realizar la metodología de solución de problemas y crear un pseudocódigo funcional para cada uno de los cuatro ejercicios. |
| Resultado: | El segundo problema se corrigió posteriormente, donde se modificó el cálculo de los impuestos de manera correcta para que el código funcionará de manera correcta.  Para el problema cuatro se resolvió muy fácilmente, no era necesario introducir tantas veces el menú dentro del según, ya poner un repetir hasta fuera del según automáticamente el menú se va a ir imprimiendo, sin necesidad de ponerlo en el algoritmo tantas veces, con esto el algoritmo paso de tener 75 líneas de código a tener únicamente 44, haciendo el algoritmo más eficaz. |
| Observaciones: | En el ejercicio 3 hubo una malinterpretación del problema, ya que según mi entendimiento se debía hacer una serie parecida a la de Fibonacci que se detenía de sumar cuando todos los elementos sumados alcanzaran o sobrepasaran un numero indicado por el usuario, por lo que no se realizó en si la serie de Fibonacci como indicaba el problema. |

|  |  |
| --- | --- |
| Práctica #6 MSP y Algoritmos en Pseudocódigo. | |
| Fecha: | Martes 21 de junio 2022 |
| Descripción: | Para esta práctica se analizaron 4 ejercicios diferentes, a los cuales se les debía realizar la metodología de solución de problemas y crear un pseudocódigo funcional para cada uno de los cuatro ejercicios. |
| Resultado: |  |
| Observaciones: | El ejercicio 1, se logro completar de manera satisfactoria se realizo dentro de un menú donde el usuario elige la figura que deseaba saber el área y perímetro y el algoritmo le imprime en pantalla los resultados.  El ejercicio 2, se logró completar de manera satisfactoria sin ningún problema, el algoritmo se realizó dentro de un menú donde el usuario elige el descuento del artículo que deseas saber el precio con descuento el algoritmo le imprime en pantalla los resultados.  El ejercicio 3 también se logro de manera satisfactoria sin embargo se realizó con estructuras de decisión y no es del todo optimo a que se pudo haber simplificado más haciéndolo con la función trunc.  El ultimo ejercicio también se realizo de manera correcta, este con dos ciclos Para y tres variables. En este ejercicio se definió numero como entero, y Numero es una palabra reservada, así que como recomendación se debe usar num u otro nombre de variable que no aparezca en negrita. |

|  |  |
| --- | --- |
| Trabajo cotidiano MSP. | |
| Fecha: | Miércoles 22 de junio 2022 |
| Descripción: | En este trabajo cotidiano se realizaron 4 ejercicios, los primeros dos se resolvieron en clase con ayuda del profesor y los otros dos quedaron pendientes de resolver, por lo que se dio oportunidad de resolver en la casa. |
| Resultado: |  |
| Observaciones: | Ambos ejercicios los logre hacer de manera correcta sin dificultad alguna. |

|  |  |
| --- | --- |
| Práctica #7 MSP y Algoritmos en Pseudocódigo. | |
| Fecha: | Martes 28 de junio 2022 |
| Descripción: | Para esta práctica se analizaron 6 ejercicios diferentes, a los cuales se les debía realizar la metodología de solución de problemas y crear un pseudocódigo funcional para cada uno de los seis ejercicios. |
| Resultado: |  |
| Observaciones: | El ejercicio 1, se logró completar de manera satisfactoria.  El ejercicio 2, se logró completar de manera satisfactoria sin ningún problema, este problema es de suma importancia ya que realiza varios pasos importantes, como asignarle valores aleatoriamente a la matriz, después con dos para anidados se recorre la matriz para encontrar el numero mayor y la posición de este.  El ejercicio 3 también se logró de manera satisfactoria sin embargo tuve varias complicaciones principalmente al aplicar la lógica, y averiguar como se iba a ir llenando las filas y columnas para luego sumar.  El ejercicio 4 se realizó sin complicación alguna, este ejercicio es de suma importancia ya que se aprende a ordenar un vector de dos maneras distintas.  En el ejercicio 6 se logro elaborar de manera satisfactoria. |

|  |  |
| --- | --- |
| Práctica #8 MSP y Algoritmos en Pseudocódigo. | |
| Fecha: | Jueves 30 de julio 2022. |
| Descripción: | Para esta práctica se analizaron 10 ejercicios diferentes, a los cuales se les debía realizar la metodología de solución de problemas y crear un pseudocódigo funcional para cada uno de los diez ejercicios.  En practica se realizó en equipo. |
| Resultado: |  |
| Observaciones: | Los primeros 4 ejercicios se resolvieron en horas de clase junto con Manfred y el compañero Daniel.  Los ejercicios restantes se resolvieron en casa, a la hora de reunirse con los compañeros del equipo hubo varias dificultades, ya que el compañero Manfred únicamente podía reunirse después de las 10 de la noche ya que a esa hora era donde podría hacer uso de su computadora, por lo nos reuníamos muy tarde para empezar los ejercicios restantes, y el compañero Daniel no logro reunirse ningún día por problemas de internet en su casa.  Para el ejercicio 5 se uso como referencia el ejercicio que el profesor resolvió en clase de la fecha del día anterior.  El ejercicio 10 no lo pudimos resolver del todo, primero por diferencias de disponibilidad de todos los miembros del equipo para reunirse. Sin embargo, al trabajar en equipo es muy bueno ya que todos pensamos de manera diferente de resolver las cosas y a veces el compañero tiene ideas mejores o mas sencillas de aplicar que las que uno tenia pensadas y viceversa., y al explicar los ejercicios uno aprende más. |

|  |  |
| --- | --- |
| Práctica #9 MSP y Algoritmos en Pseudocódigo. | |
| Fecha: | Viernes 29 de julio 2022 |
| Descripción: | Para esta práctica se analizaron 6 ejercicios diferentes, estos ejercicios se resolvieron mediante diagramas de flujo.  Algunos de los ejercicios ya estaban resueltos en pseudocódigo por lo que la realización del diagrama se hizo un poco mas sencilla. |
| Resultado: |  |
| Observaciones: | La realización de los diagramas de flujo es un poco tediosa a la hora de utilizar el programa, en varias ocasiones se me borraba el diagrama y tenía que volver a realizarlo. En la realización del ejercicio 4 hubo un detalle a la hora de llamar el subproceso, ya que yo lo hice fuera del algoritmo principal y era en el cuadro del algoritmo principal donde se imprimía si el número existe o la posición de este, pero ya me quedo claro como se debe realizar el llamado en los subprocesos o funciones. |

|  |  |
| --- | --- |
| Práctica #10 MSP y Algoritmos en Pseudocódigo. | |
| Fecha: | Miércoles 3 agosto 2022 |
| Descripción: | Para esta práctica se analizaron 4 ejercicios diferentes, a los cuales se les debía realizar la metodología de solución de problemas y crear un pseudocódigo funcional para cada uno de los cuatro ejercicios.  Esta práctica tenía como objetivo utilizar la programación Modular. |
| Resultado: | Se revisaron los cuatro ejercicios en clase, con ayuda del profesor y varios compañeros pasaron a la pizarra a explicar cómo resolvieron ellos los ejercicios. |
| Observaciones: | Los 4 ejercicios se lograron resolver. En los cuatro ejercicios use un subproceso para pedir un único valor, lo que no es del todo necesario cuando se trata de solicitar un único valor al usuario, ya que es más simple sin modularizar, como para el último ejercicio que si fue necesario el utilizar un subproceso para solicitar dos números por que se agiliza el trabajo.  Esta práctica fue de ayuda para lograr comprender un poco mejor el uso de los subprocesos y las funciones. |

|  |  |
| --- | --- |
| Práctica #11 MSP y Algoritmos en Pseudocódigo. | |
| Fecha: | Lunes 8 agosto 2022. |
| Descripción: | Para esta práctica se analizaron 3 ejercicios diferentes, a los cuales se les debía realizar la metodología de solución de problemas, crear un pseudocódigo y también crear el diagrama de flujo para cada uno de los tres ejercicios.  Esta práctica tenía como objetivo utilizar la programación Modular y los diagramas de flujo. |
| Resultado: |  |
| Observaciones: | El primer ejercicio ya se tenía resuelto en la práctica seis, sin embargo, era necesario hacerles varias modificaciones a algunas figuras geométricas y crear algunas nuevas, primeramente, hice para cada una de las figuras un subproceso que averiguaba el valor del área y perímetro, luego lo dividí en funciones y cada función únicamente averiguaba o el área o el perímetro, por lo que se realizaron dos funciones para cada una de las figuras analizadas.  El segundo ejercicio era muy vacilón pero muy complicado, se hicieron dos subprocesos para ordenar el numero de mayor a menor y otro para ordenarlo de menor a mayor, antes de eso fue necesario validar fue necesario convertir el numero que estaba en un vector convertirlo a texto, para realizar esto tuve que investigar ya que no tenia idea de cómo funcionaba las subcadenas, ni el convertir a texto un numero o viceversa, pero resultó ser bastante sencillo.  El tercer ejercicio me llevo bastante tiempo de resolver ya que estaba bastante largo y complicado y al final no pude resolverlo satisfactoriamente.  Los diagramas de flujo de los tres ejercicios no los pude realizar por falta de tiempo que se consume al realizar los pseudocódigos. |

# Glosario:

**Abstracción:** Se refiere a la capacidad de lograr entender, analizar o interpretar diferentes problemas.

**Actitud:** La actitud viene siendo algo más mental, así como la motivación, la conducta o bien el compromiso a la hora de realizar distintas actividades cotidianas.

**Acumuladores**: Un acumulador toma una variable y sirve para llevar la suma acumulativa de una serie de valores que se van leyendo o calculando gradualmente.

**Algoritmo**: Es un conjunto de pasos que contienen instrucciones que permiten solucionar un problema, procesar datos o bien crear distintas tareas o actividades.

**Algoritmos cualitativos**: Describen diferentes pasos mediante palabras.

**Algoritmos cuantitativos**: Describen pasos mediante cantidad, utilizando principalmente números o cálculos matemáticos.

**Algoritmos gráficos**: Se representan mediante diagrama de flujo.

**Algoritmos no gráficos**: Se representan mediante pseudocódigo.

**Aptitud:** Habilidad de una persona para realizar algo.

**Árbol**: Es un tipo especial de grafo en donde no se admiten ciclos y siempre existe un camino desde un nodo raíz hacia todos los demás.

**Automatización**: Ejecución de una mezcla determinada de acciones por una máquina, sin la ayuda de personas**.**

**Backend:** El backend es todo lo que esta oculto a los ojos de los usuarios, es toda la lógica de programación que hace que un sitio web funcione

**Bidimensionales**: Cuentan con dos dimensiones de filas y columnas, tal como las tablas o matrices.

**Búsqueda binaria**: Es una búsqueda que empieza en el medio de una lista ordenada, y la búsqueda binaria determina si debe continuar hacia la izquierda o hacia la derecha para encontrar el valor requerido, es mas eficiente que la búsqueda secuencial.

**Búsqueda Secuencial**: Es una búsqueda a lo largo de una lista, que va de principio a fin buscando un valor requerido.

**Byte:** Conjunto de 8 bits usado para designar un carácter, letra o número.

**C**: Lenguaje de programación usado principalmente para la programación de sistemas o

**C++:** Lenguaje de programación orientado a objetos, basado en el lenguaje C.

**Camel Case**: Es un estilo de escritura de buena práctica, que se utiliza para identificar variables. La primera letra de la primera palabra es en minúscula y la primera letra de la segunda palabra en mayúscula, ejemplo camelCase.

**Carácter**: Es un únicamente digito individual que puede representar números del 1 al 9, o bien las letras del abecedario.

**Centinela**: Es un valor que representa un punto de entrada o salida.

**Cliente**: Es un equipo o bien aplicación, que consume el servicio brindado en otro equipo y se hace cargo de la interacción necesaria con el usuario.

**CloudComputing:** Se trata de grandes servicios de software remotos, en los cuales se puede acceder a través de la conectividad y gran escala de internet.

**Código fuente**: Líneas de código que contienen instrucciones que la computadora debe seguir para ejecutar un programa específico.

**Compilador**: Programa que traduce instrucciones escritas en un lenguaje de programación de alto nivel a un lenguaje de máquina.

**Compilar**: Generar un programa en lenguaje de máquina a partir de un lenguaje de programación de alto nivel.

**Concatenar**: Se refiere a la unión o al enlace de distintos operandos de tipo cadena.

**Constant Case**: Es un estilo de escritura de buena práctica, que se utiliza para identificar las constantes en un programa. La primera palabra se escribe en mayúscula, luego separa la primera palabra de la segunda palabra con un guion bajo y la segunda palabra también en mayúscula, ejemplo NUMERO\_PI.

**Constante**: Valor almacenado en la memoria que no puede cambiar en la ejecución de un programa.

**Contador**: Es una variable que aumenta de uno en uno dentro de un ciclo.

**Convertir**: Se convertir un conjunto de numero a texto o viceversa.

**DataScience:** Se basa en explorar y analizar datos de múltiples fuentes, extrayendo así conocimiento a partir de los datos para poder responder a las preguntas que se le formulan.

**Datos carácter o cadena**: Cadenas de caracteres es decir cualquier letra, palabra frase u oración.

**Datos lógicos**: Cuando necesitamos guardar una expresión lógica. (True o False)

**Datos numéricos enteros**: son valores numéricos que no tienen decimales (Byte, short, int, long).

**Datos reales**: Datos numéricos que poseen parte entera y también parte decimal (Float, Double).

**Depuración**: Detección, localización y eliminación de errores en un programa. También llamado Debugging.

**Diagrama de flujo**: Es una representación gráfica de un conjunto de pasos para solucionar un problema, en pocas palabras es la representación grafica de un algoritmo.

**Dimensiones**: Es el conjunto de datos del mismo tipo que se almacenan en posicione consecutivas de memoria y reciben un nombre común**.**

**Documentación del algoritmo**: Son una base de comentarios a lo largo del código para tener una idea de que se fue realizando, esto con el fin de tener más orden y claridad a la hora de que otra persona vaya a revisar el código o bien para explicarlo.

**Eficiencia**: Básicamente es la capacidad de realizar o cumplir de forma correcta una función, con una menor cantidad de recursos.

**Ensambladores**: Es un programa informático encargado de traducir un archivo fuente escrito en un lenguaje ensamblador a un lenguaje máquina.

**Ensayo y error**: Se basa en probar alternativas y verificar si funciona, si no funciona se descarta y se prueba otra alternativa diferente.

**Entrada**: Son los datos que el algoritmo recibe por medio del teclado.

**Estructura Anidada**: Es una estructura ya sean ciclos o estructuras de decisión, dentro de otros.

**Estructura**: Es la sucesión que lleva la lógica de programación, en secuencia, selección repetición.

**Expresiones**: Una expresión se entiende como la combinación de variables, constantes a través de símbolos de operación denominados operadores. Estas expresiones utilizan las operaciones básicas de la matemática, como la suma, resta, multiplicación, división, entre otros.

**Frontend:** Se refiere a el desarrollo web que se dedica a la parte frontal de un sitio web, o sea a todo el diseño de un sitio web, desde la estructura del sitio hasta los estilos como colores, fondos, tamaños hasta llegar a las animaciones y efectos.

**Función Aleatorio (x, y)** devuelve un número entero aleatorio entre dos números agregados.

**Función Azar(x):** Devuelve un número entero aleatorio desde 0 hasta el número deseado.

**Función Mod**: devuelve el resultado de una división.

**Función Trunc**: Calcula la parte entera de un número.

**Grafos**: Es una estructura formada por una serie de nodos en donde cada nodo contiene un valor y una o más referencias a nodos asociados.

**Hardware**: Es toda la parte física de un sistema informático, conformado por componentes electrónicos, eléctricos, mecánicos, hidráulicos.

**Heurístico**: El método heurístico es un conjunto de técnicas necesarias para lograr llegar a una solución óptima.

**Identificador**: Indica el nombre que se le da una variable o elemento de un algoritmo, si es entero, real, lógico o de tipo carácter; Existen identificadores predefinidos que ya están contenidos en los diferentes lenguajes de programación.

**Indentar**: Se utiliza para ordenar y que se pueda leer, entender de mejor manera un código

**Información**: Es el resultado del procesamiento de datos. Todo aquello que permite adquirir cualquier tipo de conocimientos.

**Instrucción**: Es un conjunto de caracteres que enumera una operación a realizarse y el valor o ubicación de uno o más operandos solicitados.

**Instrucciones de asignación**: Consisten en demostrar el resultado de una expresión a una variable. (=,🡨).

**Instrucciones de escritura:** Demuestran los valores obtenidos de la evaluación de una expresión (Escribir…).

**Instrucciones de lectura:** Asignan a una, dos o más variables, valores recibidos desde un dispositivo de entrada (Leer…).

**Interfaz**: Conexión entre dos componentes de hardware, entre dos aplicaciones o entre un usuario y una aplicación.

**Intérpretes**: Son programas capaces de analizar y ejecutar otros programas.

**La función Redon**: Redondea un número real al entero más cercano por abajo o por arriba.

**Lenguaje de máquina**: es el sistema de códigos directamente interpretable por un circuito micro programable, como el microprocesador de una computadora u otros dispositivos electrónicos.

**Lenguaje ensamblador:** Lenguaje de programación simbólico de bajo nivel. Tiene una correspondencia uno a uno en las instrucciones y formato de datos con el lenguaje de máquina.

**Lenguajes de alto nive**l: Están entablados a programas más sencillos y rápidos.

**Lenguajes de bajo nivel**: Los lenguajes de bajo nivel mediante diferentes instrucciones ejercen un control directo sobre el hardware.

**Lenguajes de programación**: Se utilizan para escribir programas, como por ejemplo Python, Java, C, C++, entre muchos más.

**Lógica natural:** es una destreza de razonar sin tomar en cuenta la ciencia.

**Lógica:** En pocas palabras, se puede entender que la lógica es un estudio de diferentes alternativas validas de deducción, para identificar el razonamiento correcto y el que no lo es.

**Lógicos Booleanos**: Son valores lógicos, que únicamente se representan como verdadero o falsos.

**Longitud:** Retorna la cantidad de caracteres de una cadena x.

**Matriz**: Es una serie de elementos, normalmente de números ordenados por filas y columnas.

**Mayúscula:** Convierte a mayúscula una cadena dada.

**Menú**: Lista de opciones mostrada sobre una pantalla de las cuales el usuario puede seleccionar.

**Minúsculas:** Convierte a minúsculas una cadena.

**Modularizar**: Es un proceso donde se divide un algoritmo en diferentes módulos o subprogramas mas simples. Esto con el fin de hacerlo mas sencillo, comprensible y se pueden volver a utilizar.

**Modulo**: Unidad de un algoritmo que tiene una tarea explica.

**Multidimensionales:** Contiene tres o más dimensiones.

**Optimidad u optimo**: Es la mejor opción o la opción más adecuada para un programa.

**Palabras reservadas**: Son palabras que ya las trae el lenguaje, como por ejemplo el repetir para, mientras, según, entre otros.

**Parámetros por referencia**: Permiten que se modifique su valor.

**Parámetros por Valor**: Los parámetros por valor no permiten ser modificados.

**Probar paso a paso**: Se ejecuta de manera lenta el algoritmo para analizar y evaluar la evolución de los datos, el orden y cómo funcionan.

**Proceso:** Manipular datos o realizar otras operaciones de acuerdo con un programa.

**Programa**: Instrucciones dadas a una computadora.

**Programación**: Se refiere a el desarrollo, creación y escritura de programas de software.

**Programador:** Persona que define la solución a un problema y escribe las instrucciones solicitadas por una computadora para llevar a cabo esa solución.

**Pseudocódigo**

**Pseudocódigo**: Se refiere a los pasos lógicos necesarios para empezar a resolver un problema en código, es desarrollado para que un ser humano pueda leerlo y entenderlo, no interpretado por una computadora como normalmente es.

**Python:** es un lenguaje de programación de alto nivel que se utiliza para desarrollar aplicaciones de todo tipo.

**Recursividad:** es el proceso de resolver un problema reduciéndolo a uno o más subproblemas que son idénticos en su estructura al problema original y más simples de resolver.

**Salida**: La salida se encarga de mostrar los datos de un seudocódigo en la interfaz de un monitor o bien de una impresora.

**Semántica**. Es el significado que se le atributa a la combinación de palabras o expresiones de un lenguaje de programación.

**Sintaxis**: Se refiere a reglas meramente gramaticales que se encuentran en los lenguajes de programación.

**Sistema numérico binario**: Sistema de numeración de base 2, es decir, que sólo usa dos dígitos 0 y 1, o bien True o False.

**Sistema numérico hexadecimal**: Sistema numérico de base 16, generalmente usando los dígitos 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

**Software**: Es un conjunto de programas y aplicaciones meramente informáticas que cumplen funciones muy específicas; Ejerce el control sobre el hardware.

**Subcadena**: Extrae una parte de la cadena indicando el inicio y el fin de esta.

**Tipo de datos**: Es el tipo de valor que se le asigna a una variable, puede ser com0o carácter, booleano, real, entero.

**Unidimensionales**: Únicamente cuentan con una dimensión ya sean filas o columnas, como por ejemplo los vectores.

**Variable**: Es un espacio en la memoria donde se almacena un dato especifico, este dato puede variar.

**Vectores o arreglos**: Se identifica como una serie de elementos ordenados particularmente.

# Conclusiones:

* A lo largo del módulo de lógica computacional he aprendido nuevas definiciones y conceptos de programación, con las que anteriormente no tenía conocimiento.
* Se ha desarrollado lógica, aprendiendo a razonar y analizar diferentes formas para resolver distintos problemas, mediante la metodología de solución de problemas.
* Diseñar distintos algoritmos mediante el uso de pseudocódigo para resolver problemas según una serie de requerimientos dados.

# Bibliografía:

José González Moris. Glosario. Manual Introducción a la Programación Utilizando Pseint (19). Colegio: San Martin de Porres: La Serena.

Iván García. (2014). Glosario. Fundamentos de la programación. (152). Ecuador: UPEC.