**Alumno:** Ortuño, Bruno

**Marco Teórico:**

1. Definir los conceptos de Informática y computadora.

2. Definir Hardware y Software

3. Describir los roles de los Programadores

4. Relacionar los conceptos de la Diapositiva 9

5. Describir el Modelo de Von Neuman

6. Que es un Lenguaje de Programación, que permite.

7. Describir los lenguajes de Alto , Medio y Bajo Nivel, dar ejemplos.

8. Que es un Compilador, que función cumple

9. Que es un Interprete, como funciona

10. Relacionar Java Virtual Machine , .Net Framework con Jit

11. Relacionar Problema, Algoritmo y Programa

12. Describir el concepto que afirma “SOMMERVILLE” en su libro Ingeniería de Software, un modelo del proceso del software es una representación abstracta de un proceso del

software.

13. Describir el Modelo Iterativo e Incremental

14. Relacionar UML, Ingeniería de Software y modelos.

—---------------------------------------------------------------------------------------------------------

1. La informática es el conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores.

En cambio la computadora es una máquina electrónica, analógica o digital, dotada de una memoria de gran capacidad y de métodos de tratamiento de la información, capaz de resolver problemas matemáticos y lógicos mediante la ejecución de programas informáticos.

1. El hardware hace referencia a los componentes físicos de la computadora, aquello que “se puede tocar”, en cambio al hablar de software nos referimos a aquello intangible, contenido en una pieza de hardware o unidad de almacenamiento, que abarca todos los programas, instrucciones y reglas informáticas para ejecutar alguna tarea.
2. Los roles de un programador pueden ser de:

**Gestores**, siendo responsables de supervisar y coordinar proyectos. Encargándose de la planificación de tareas, asignación de recursos, seguimiento del progreso, y asegurarse de que se cumplan con los plazos y calidad.

**Analistas**, encargándose de comprender las necesidades del cliente o usuario final y traducirlas en requisitos técnicos claros y específicos.

**Diseñadores**, centrándose en la creación de la arquitectura y la interfaz de usuario de una aplicación. Trabajan en colaboración con los analistas y los clientes para diseñar la estructura general del software, incluyendo la disposición de la interfaz de usuario, la navegación y la experiencia del usuario.

**Programadores**, siendo responsables de escribir el código que hace que el software funcione según lo diseñado. Utilizan lenguajes de programación específicos y siguen las pautas establecidas por los diseñadores y los analistas para escribir código limpio, eficiente y fácil de mantener.

**Probadores,** o **testers**, se encargan de probar el software para identificar y corregir errores o fallos antes de su lanzamiento. Realizan pruebas de funcionalidad, rendimiento, seguridad, etc. para garantizar que el software cumpla con los requisitos y expectativas.

**Administradores de sistemas**, siendo responsables de mantener y gestionar la infraestructura necesaria para ejecutar el software. Esto incluye la configuración y el mantenimiento de servidores, bases de datos, redes y otros sistemas informáticos para garantizar que el software funcione de manera óptima y segura.

1. La informática es el estudio y la gestión de la información utilizando computadoras, mientras que la programación es el proceso de escribir instrucciones para que las computadoras las ejecuten.   
   El lenguaje máquina es el conjunto de instrucciones binarias que una computadora entiende directamente. El ensamblador es un lenguaje de bajo nivel que utiliza palabras clave para representar instrucciones en lenguaje máquina de una manera más legible para los humanos. De igual modo, los lenguajes de programación que están más cerca del lenguaje humano facilitan la escritura de programas mediante la abstracción de las instrucciones de bajo nivel usando un conjunto de palabras reservadas.  
   Los programas desarrollados usando estos lenguajes de programación se utilizan para resolver situaciones que requiere una solución, los problemas. Así, un algoritmo es una secuencia de pasos bien definidos para resolver un problema, y un programa no es más que la implementación de dicho algoritmo en un lenguaje de programación específico.

La programación se utiliza para crear programas de software mediante la escritura de código. La ingeniería del software es la disciplina que se encarga de aplicar principios de ingeniería al desarrollo, mantenimiento y evolución de software de calidad.

En este contexto el lenguaje de programación C++ es un lenguaje de programación de alto nivel que se utiliza principalmente para el desarrollo de sistemas operativos, juegos, aplicaciones de escritorio, etc. Es una extensión del lenguaje C, que incorpora características de programación orientada a objetos y una gestión de memoria más segura, lo que lo convierte en una opción más potente y versátil para el desarrollo de software.

C++, así como todos los lenguajes de programación, responde a una sintaxis particular, que es el conjunto de reglas que define la estructura y el formato válidos para escribir código que funcione de forma acorde a lo esperado.

Así, un primer programa en C++ sería un ejemplo inicial de código, que generalmente suele ser un programa simple como "Hola Mundo".

Para desarrollar ese y otros programas se utilizan herramientas de desarrollo, que son software y recursos que facilitan el proceso de desarrollo de software, como editores de código, compiladores, depuradores, IDEs y sistemas de control de versionado.

1. El Modelo de Von Neumann describe una computadora con cuatro componentes: una unidad de procesamiento, una memoria, una unidad de control y dispositivos de entrada/salida.

La característica principal de este modelo es que tanto los datos como las instrucciones se almacenan en la misma memoria y se manejan de manera similar, lo que permite que los programas se modifiquen y manipulen como datos.

1. Un lenguaje de programación es el conjunto de instrucciones y reglas utilizadas para escribir programas. Permiten comunicarse con las computadoras para realizar diversas tareas, facilitando la solución de diversos problemas mediante el razonamiento y aplicación de algoritmos en el desarrollo de programas.
2. Los lenguajes de programación se pueden clasificar en:

- **Lenguajes de bajo nivel:** Estos lenguajes están más cerca del lenguaje de máquina y son difíciles de entender para los humanos. Por ej: el lenguaje ensamblador.

- **Lenguajes de medio nivel:** Estos lenguajes ofrecen un equilibrio entre la cercanía a la maquina y la legibilidad para humanos. Por ej: C y C++.

- **Lenguajes de alto nivel:** Estos lenguajes están más alejados de la máquina y son más fáciles de entender para los humanos. Por ej: Python, JavaScript, Ruby.

1. Un compilador es un programa que traduce el código fuente escrito en un lenguaje de programación de alto nivel a código máquina, en un archivo de tipo objeto, que la computadora puede posteriormente entender y ejecutar una vez enlazadas todas sus dependencias.
2. Un intérprete es un programa que lee y ejecuta el código fuente de un programa línea por línea, traduciendo y ejecutándolo al mismo tiempo. A diferencia de un compilador, que traduce todo el código de una vez, un intérprete lo hace progresivamente. Esto permite una ejecución más rápida en comparación con la compilación, pero generalmente es más lento durante la ejecución del programa.
3. Tanto Java Virtual Machine (JVM) como .NET Framework utilizan técnicas de compilación en tiempo de ejecución. De esta forma se compila el código de byte en código nativo de la máquina en tiempo de ejecución, lo que mejora el rendimiento del programa al ejecutarlo. JVM interpreta inicialmente el código de byte y luego utiliza JIT para compilar partes del código en código nativo cuando se detecta que se ejecutan con frecuencia.
4. Un problema es una situación que requiere una solución, y un algoritmo es el conjunto de pasos ordenados y bien definidos que llevan a la solución del problema. Así, un programa es la implementación de un algoritmo en un lenguaje de programación específico.
5. El concepto que afirma Sommerville es que un modelo de proceso de software proporciona una representación abstracta y simplificada de cómo se lleva a cabo el desarrollo de software. No describe todos los detalles, sino que se centra en las fases clave y las actividades principales del proceso para el desarrollo.
6. El modelo iterativo e incremental es un enfoque de desarrollo en el que el proceso de desarrollo se divide en pequeñas iteraciones o incrementos. Cada iteración incluye el desarrollo, la prueba y la entrega de una parte del proyecto. El modelo permite una retroalimentación temprana y frecuente, lo que facilita la adaptación a los cambios en los requisitos y reduce los riesgos asociados al desarrollo de software.
7. El UML, o “Unified Modeling Language”, es un lenguaje estándar para modelar sistemas de software. Se utiliza en ingeniería de software para visualizar, especificar, construir y documentar las partes del sistema. UML proporciona un conjunto de herramientas y notaciones para representar diferentes aspectos de un sistema de software, como su estructura, comportamiento, etc. Los modelos UML son utilizados por los ingenieros de software para comprender, comunicar y gestionar la complejidad de los sistemas durante todo el ciclo de vida del proyecto.