**TALLER No. 1**

1. Resolver:
2. Definir los conceptos fundamentales involucrados en el proceso de resolución de un problema a través de la computadora:

Se pueden definir a partir de cuatro niveles

* Análisis del problema: Planteamiento de un problema por parte del usuario a un programador que deberá resolver mediante su ordenador.
* Diseño o desarrollo de un algoritmo: Especificación clara y concisa de los pasos necesarios para resolver un determinado problema, pero disponiendo de una anotación algorítmica que permita describir las operaciones, los objetos manipulados por le algoritmo y controlar la realización de las acciones descritas, indicando la forma en que estas se organizan en el tiempo.
* Transformación del algoritmo en un programa (codificación)
* Ejecución y validación del programa

1. Definir:

* Lenguaje: Notación constituida por símbolos y reglas que permite escribir programas. Está compuesto por sintaxis (reglas) y semántica (significado de los símbolos y las palabras que utiliza). Permite establecer una comunicación sistematizada y precisa con una computadora.
* Programa: Secuencia lógica de instrucciones escritas en un determinado lenguaje de programación que dicta a la computadora las acciones que debe llevar a cabo. Está compuesta por dos partes: operando y operador.
* Compilador: Es un programa, suministrado por el fabricante del lenguaje, cuyo objetivo es convertir el programa fuente en un programa objeto. Realiza las siguientes funciones: traduce las instrucciones del programa fuente, asigna áreas y direcciones de memoria, suministra constantes y otros datos, produce un diagnóstico de errores y genera el programa objeto.
* Intérprete: Es un programa que va leyendo poco a poco el código que el programador escribe y va traduciéndolo y ejecutándolo según se traduce. En estos casos no hay una traducción completa, ya que no se genera un programa directamente ejecutable. Los lenguajes interpretados son menos eficientes que los compilados.

1. Identificar los antecedentes de la programación orientada a objetos.

* Sus conceptos fundamentales ya existían en lenguajes como Simula en 1067 y Smaltalk en 1070. Entre los 60 y 70 surge una crisis de software a causa de la dificultad de cambiar y modificar los programas existentes y la disminución de programadores dispuestos a dar mantenimiento a los programas. A raíz de esta crisis surge la programación estructurada, mediante la cual se descompone en procedimientos individuales a los programas.

La programación orientada a objetos permite construir programas complejos a partir de entidades de software más simples llamadas objetos, que son instancias reales o muestras de clases, lo cual permite heredar datos y códigos de aplicaciones existentes.

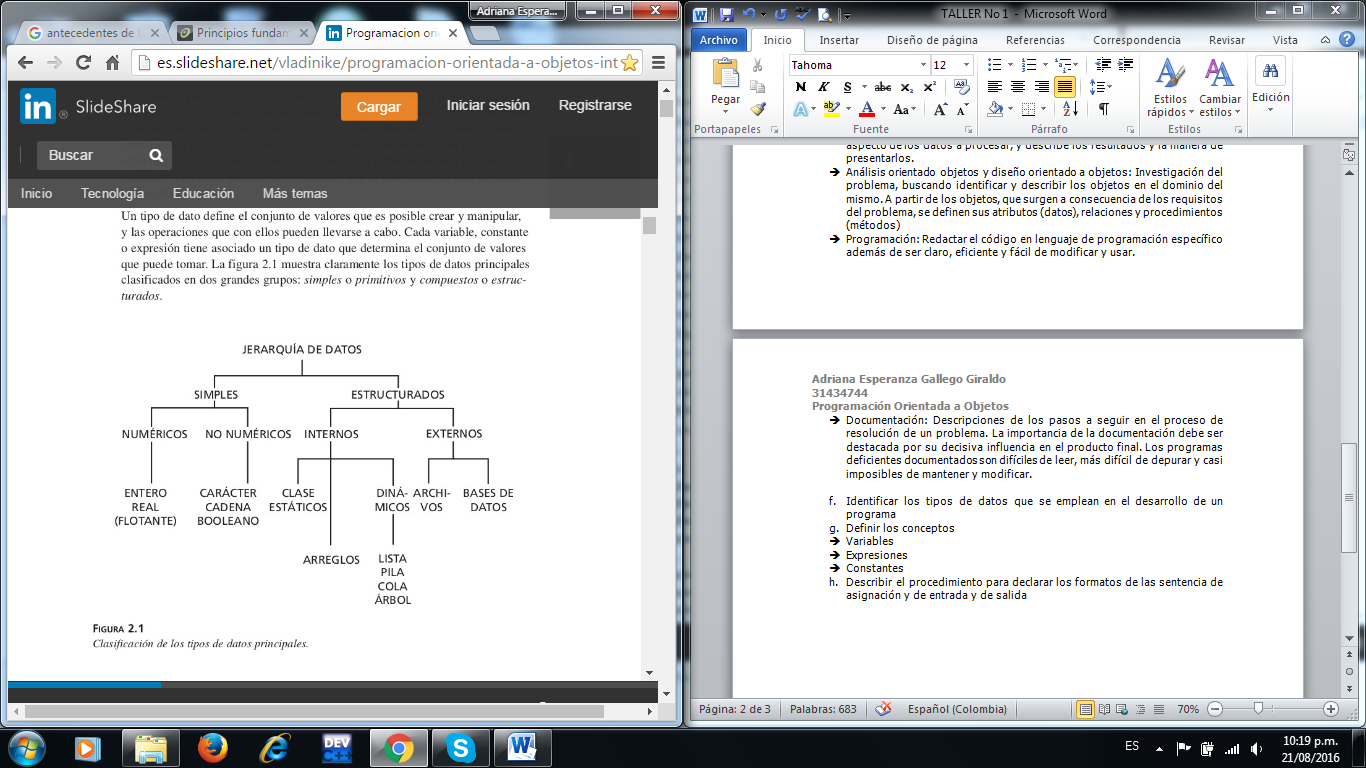
1. Definir los conceptos involucrado en la programación orientada a objetos

* Clases
* Nombre
* Atributos
* Métodos
* Objeto
* Estado
* Comportamiento
* Identidad
* Herencia
* Encapsulamiento
* Público
* Protegido
* Privado

1. Identificar las etapas para la resolución de un problema bajo un enfoque orientada a objetos

* Definición del problema o dominio del problema: Permite describir en forma narrativa o esquemática, de modo claro y concreto y en un lenguaje corriente el problema que ha de resolverse. Presenta la procedencia y el aspecto de los datos a procesar, y describe los resultados y la manera de presentarlos.
* Análisis orientado objetos y diseño orientado a objetos: Investigación del problema, buscando identificar y describir los objetos en el dominio del mismo. A partir de los objetos, que surgen a consecuencia de los requisitos del problema, se definen sus atributos (datos), relaciones y procedimientos (métodos)
* Programación: Redactar el código en lenguaje de programación específico además de ser claro, eficiente y fácil de modificar y usar.
* Documentación: Descripciones de los pasos a seguir en el proceso de resolución de un problema. La importancia de la documentación debe ser destacada por su decisiva influencia en el producto final. Los programas deficientes documentados son difíciles de leer, más difícil de depurar y casi imposibles de mantener y modificar.

1. Identificar los tipos de datos que se emplean en el desarrollo de un programa



1. Definir los conceptos

* Variables: área de almacenamiento temporal a la que se ha asignado un nombre simbólico y cuyo valor puede ser modificado a lo largo de la ejecución de un programa
* Expresiones: Combinación de valores literales, variable y operadores. Las expresiones están sometidas a unas reglas de evaluación que se deben conocer.
* Constantes: valor definido que no cambia durante la ejecución de un programa

1. Describir el procedimiento para declarar los formatos de las sentencia de asignación y de entrada y de salida

* Asignación: nombre\_de\_variable = expresión

La asignación se lleva a cabo evaluando la expresión y asignando el resultado obtenido a la variable, definida a la izquierda del símbolo de asignación.

* Entrada: las operaciones de entrada permiten leer valores y asignarlos a determinadas variables. Los datos se introducen a la computadora mediante dispositivos de entrada y son almacenados en las direcciones de memoria asignadas a las variable
* Salida: Permiten desplegar o imprimir los resultados o valores guardados en memoria en un dispositivo de salida.

2

