

## ***Unidad N°1: Actividades sugeridas.***

### ***Preguntas orientadoras***

- 1) ¿A que hace mención la palabra informática?
- 2) ¿Qué determina el paso de una generación a otra en lo que se refiere a la evolución de los ordenadores? Mencionar las características distintivas de cada generación.
- 3) Identifique:
  - a) ¿Cuál fue la primera computadora que tuvo el primer programa diseñado para ser almacenado?
  - b) ¿Qué máquina fue utilizada para descifrar mensajes cifrados alemanes durante la segunda guerra mundial y de qué manera funcionaba?
  - c) ¿En qué hardware se jugó el primer videojuego computarizado? Realice una breve descripción del mismo.
  - d) ¿Cuál fue el primer ordenador en ser atacado por un virus?. ¿Qué denominación recibió ese virus?
  - e) ¿Cuál es el computador predecesor de las actuales computadoras personales? Indicar características generales.
- 4) ¿Qué es un ordenador o también denominado computador o computadora? Indicar las partes que conforman un sistema de cómputo.
- 5) Realice un diagrama de la arquitectura de Von Neumann y realizar una breve descripción de cada parte del diagrama.

INGENIERÍA EN COMPUTACIÓN. Programación Estructurada (4D1). Ciclo 2016.

- 6) Realice un cuadro comparativo entre los distintos tipos de memoria teniendo en cuenta: capacidad de lectura y escritura, volatilidad, usos.
- 7) ¿Qué parte del sistema de cómputo interactúa con el hardware para hacer operativa una computadora?
- 8) Defina programa y sistema operativo.
- 9) ¿Cuál es la finalidad de los lenguajes de programación?. Realizar una breve descripción de los 3 tipos de lenguajes nombrados en clase.
- 10) Realice un esquema de cómo hace el computador para entender lo que escribe un usuario en un programa.
- 11) Explique la diferencia entre un compilador y un intérprete.
- 12) Explique con sus palabras qué se entiende por algoritmo.
- 13) Relacione los siguientes términos: problema, programa, algoritmo lenguaje de programación.
- 14) ¿Qué características debe tener un algoritmo para resolver un problema específico?
- 15) Busque un ejemplo y realice una representación gráfica del algoritmo que resuelve el problema planteado. Para ello utilice todas las herramientas planteadas en clase (Pseudocódigo, Diagramas de flujo u organigramas, Diagramas de Nassi-Schneiderman).
- 16) A partir de la definición de Programación Estructurada diga por qué cree que la materia se dicta en la carrera Ingeniería en Computación.
- 17) Mencione y explique las técnicas que se incorporan en la programación estructurada.

18) Mencione las características principales del lenguaje de programación C y los elementos que constituyen un programa escrito en este lenguaje.

### *Ejercicios*

1) Indique cuántos bytes y cuántos bits son:

a)1 KB

b)1 MB

c)1 GB

d)1 TB

2) Convierta de binario a decimal:

a)101110

b)101

c)11

d)1100

e)101

f)110

g)1

h)111

i)11100

j)10101

3)Convierta de decimal a binario

INGENIERÍA EN COMPUTACIÓN. Programación Estructurada (4D1). Ciclo 2016.

a)632

b)146

c)20

d)2

e)54

f)11

g)323

h)95

i)16

j)242

4)Convierta de hexadecimal a decimal o a la inversa según corresponda:

a)0x3A2

b)0x471B

c)3402

d)0xEF

e)20

f)0x5C6

g)0xD89

h)15

i)4012

j)256

5) Encuentre la representación hexadecimal, de acuerdo al standard IEEE 754, de la expresión  $f = 12.34$ .

6) ¿Qué valor de tipo float representa el hexadecimal 408A45A2 según el standard IEEE 754?

7) Realice el análisis y el diseño del algoritmo (pseudocódigo y diagrama de flujo) para los siguientes enunciados:

a) Calcule el área de un triángulo rectángulo conociendo la base y sabiendo que su altura es igual al cuadrado de la base.

b) Indique de un grupo de 30 personas cuántas son mayores de 18 años.

c) Dados los coeficientes de una ecuación de segundo grado determine si las raíces son reales.

### *Actividades extra*

1) Teniendo los registros de valores de temperatura en distintos momentos del día determine el promedio e indique cuáles de esas temperaturas tienen una diferencia con el promedio mayor a 10 grados.

2) Calcule el promedio de las notas del primer parcial de Programación Estructurada sabiendo que, como recompensa se les agrega un 20% de nota a aquellos que tengan nota mayor o igual a 7.

## *ESTÁNDAR RPP*

### *Introducción*

El motivo principal para adoptar este estándar al momento de escribir código en lenguaje C es reducir el número de “bugs”(errores en programas o software que desencadenan resultados indeseados) presentes en los nuevos software embebidos y en el código agregado o modificado por aquellos que actualizan y corrigen ese software.

Siempre que sea posible se recomienda utilizar, dentro de un conjunto de alternativas, aquellas reglas de estilo que permitan prevenir la aparición de “bugs”.

Un estándar de codificación por si sólo no puede eliminar todos los “bugs” de sistemas embebidos complejos, pero debería ser utilizado como piedra fundamental de un desarrollo más amplio de software embebido y como parte de un proceso que garantice la calidad.

Otra razón importante para adoptar este conjunto de reglas incluye el aumento de portabilidad y legibilidad del código fuente, de manera que el firmware pueda ser actualizado y reutilizado con menores costos. Un estándar genera beneficios no sólo para el grupo desarrollador, sino que también beneficia a toda la organización al reducir el tiempo necesario para que los individuos puedan entender o revisar el trabajo de sus pares.

### *Principios Guía*

Este estándar de codificación fue realizado en concordancia con los siguientes principios fundamentales:

- 1) Es más barato y más fácil evitar la formación de "bugs" en el código, que encontrarlos y eliminarlos luego que se han generado y causan errores.
- 2) Para bien o para mal el lenguaje de programación C estándar permite una cantidad significativa de variantes en las decisiones tomadas por los que implementan los compiladores. De esa manera, programas compilados desde códigos fuente en C idénticos, pueden comportarse muy diferente en tiempo de ejecución. Esas zonas grises en el lenguaje reducen enormemente la portabilidad de los programas si estos no son minuciosamente diseñados.
- 3) Estas reglas priorizan la legibilidad del código y su portabilidad por encima de la eficiencia de ejecución o la conveniencia del programador.
- 4) Existen diversas fuentes de error ("bugs") en los programas diseñados. El individuo que da origen al programa crea algunos "bugs". Otros resultan de malentendidos de aquellos que actualizan, extienden o reutilizan el código.

La cantidad y gravedad de los errores que el programador original introduce pueden ser reducidos a través de una disciplinada conformidad con ciertas prácticas de escritura de código.

En el caso de aquellas personas que actualizan el programa, la cantidad y gravedad de los "bugs" generados puede verse influenciada por el programador original, si este utiliza un estándar fijo y portable. Además resulta una buena práctica el uso disciplinado de comentarios y prácticas de estilo fijas, de manera que todos en la organización pueden entender de manera más sencilla el significado y el uso apropiado de variables, funciones, etc.

- 5) En la ausencia de una regla que resulte necesaria o ante la aparición de conflicto entre reglas, estos principios guía deben encauzar la decisión a tomar.

## *1.Reglas Generales*

### 1.1 ¿Cuál versión de C?

- ✓ Todos los programas deberían ser escritos respetando la versión C99 del estándar ISO para la programación en lenguaje C.
- ✓ Siempre que se utilice un compilador C++, se lo debería configurar para restringir el lenguaje a la versión de ISO C elegida.
- ✓ El uso de extensiones de palabras claves del compilador propietario y de #pragmas debería ser reducida al mínimo necesario.