

# Microproyecto No. 1

---

## I. Objetivos del Proyecto:

El objetivo de este proyecto es que el alumno, mediante el desarrollo de un programa de mayor complejidad a los ya hechos en los laboratorios, integre los conocimientos y destrezas adquiridas en la asignatura. En este sentido, se persigue que el estudiante demuestre el dominio y conocimiento del proceso de conversión de números entre diferentes bases numéricas y las destrezas necesarias para realizar operaciones lógicas y aritméticas mediante el uso práctico del lenguaje ensamblador IA32.

## II. Requisitos del Proyecto:

Es **MUY IMPORTANTE** que antes de hacer cualquier consulta sobre el proyecto lo haya leído detalladamente y detenidamente, haciendo un esfuerzo por entender el enunciado. Por favor, plantee sus dudas por escrito de forma clara y concreta.

1. Para la realización del presente proyecto son necesarios y suficientes los conceptos ya dictados en clase asociados a los temas 2 y 3 del contenido de la asignatura.
2. El proyecto puede ser realizado por grupos de hasta 2 estudiantes del mismo horario de laboratorio.
3. La entrega del proyecto es obligatoria y **NO habrá prórroga** de la fecha prevista para su entrega.
4. Cada diseño e implementación del proyecto debe ser única, los grupos cuyos proyectos sean copias serán sancionados con la reprobación de la materia.
5. Cada grupo debe entregar un reporte de proyecto y el programa funcionando.
6. Debe ser implementado en lenguaje ensamblador de la arquitectura IA32.
7. El programa debe ensamblar sin errores, ya que, de no ser así se considerará “no entregado”.
8. **El proyecto se entregará el lunes 20 de marzo de 2017**, sin prórroga. Para ello, todos los estudiantes deben enviar a su respectivo preparador de laboratorio, vía email, el proyecto y el informe:

### Recuerde:

- **La NO entrega a tiempo (en las fechas establecidas) del proyecto se considerará como no entregado el proyecto.**

## III. Enunciado del problema:

A los estudiantes de la asignatura *Organización y Estructura del Computador I* de la Escuela de Computación de la UCV se les ha encomendado la tarea de realizar una Shell llamada **Shell Calculator**.

Un *shell* es un programa que tiene como finalidad interactuar con el computador por medio de un conjunto de órdenes o comandos de tal manera que el usuario tenga la posibilidad de usar los recursos que esta dispone. El objetivo de este microproyecto es construir una sencilla calculadora interpretativa, que tiene como objeto realizar operaciones aritméticas (**suma, resta**) y lógicas (**and, or y xor**). La representación de los números está en **complemento a 2 (CA2)** en una precisión de **8 bits** y que las entradas de los datos son representados en sistema **binario** o **hexadecimal**, establecido con el comando **setsys** que define el sistema ha utilizar.

### Comando Setsys:

Sintaxis: setsys <Numero>

Donde <Numero> es el número que indica el sistema a usar.

### Valor de retorno:

Establece el sistema de la calculadora en base <numero>. Si <numero> es negativo o no existe, debe retornar mensaje de error indicando la invalidez de la misma.

### Ejemplo:

```
>>setsys 2 //Establece los operando en sistema binario
>>setsys 16 //Establece los operando en sistema hexadecimal
```

## Sección de Convenio de entrada de datos y salidas de resultados:

### Datos de Entrada:

Su microproyecto debe comenzar con un indicador de *shell* (>>) que indica que su programa está en la espera de alguna orden. En la siguiente tabla, ilustraremos los siguientes comandos:

Comando	Descripcion
Add <operando1> <operando2>	Realiza la suma de los números
Sub <operando1> <operando2>	Resta dos números
And <operando1> <operando2>	Realiza operación lógica “y”
Or <operando1> <operando2>	Realiza operación lógica “o”
Xor <operando1> <operando2>	Realiza operación lógica “o exclusivo”

Para realizar dicha operación debe especificar el sistema en que se va representar los operando por medio del comando **setsys**. Para cambiar de sistema se debe introducir el comando **chsys** que indica al usuario que debe elegir el sistema en base n. Mientras no se elija el sistema, no se podrá realizar las operaciones correspondientes. Para salir del programa se introduce el comando **exit**.

### Datos de Salida:

Cada comando debe generar como dato de salida la impresión por pantalla el valor de cada operando y el resultado final de la siguiente manera:

<operador> <operando 1>  
                  <operando 2>

---

<resultado>

Donde <operador> es el símbolo de la operación. Para el caso del comando setsys, debe indicar el sistema en que se ha cambiado, en caso que existe un desbordamiento en sus resultado se debe indicar un mensaje indicando esto.

### Ejemplos:

Entrada	Salida
>>setsys 2	--cambiando a sistema en base 2
>>add 10010101 00000111	+ 10010101
>>sub 00000001 00011000	00000111
>>chsys	100111100
>>setsys 16	
>>add f0 10	+ 00000001
>>chsys	11101000
>>setsys 2	11101001
>>add 01111111 00000010	--Selecciona base
>>exit	--Cambiando a sistema en base 16
	+ f0
	10
	00
	--Selecciona base
	--Cambiando a sistema en base 2
	+ 01111111
	00000010
	10000000
	--Overflow

## IV. Condiciones de entrega:

1. El código fuente debe estar intradocumentado, es decir, se deben usar nombres de etiquetas, variables y comentarios autoexplicativos que reflejen su función dentro del código y faciliten la comprensión del código fuente del programa.
2. Además, la cabecera del código fuente debe indicar: fecha, nombres y apellidos con nros. de cedula de identidad del grupo, horario de laboratorio y nombre del preparador de laboratorio, así como también una breve descripción de lo que hace el programa.
3. El nombre del archivo del código fuente debe tener el siguiente formato: *Apellido1-Apellido2-seccion\_lab*, donde Apellido 1 y Apellido2 son los apellidos de cada estudiante del grupo y seccion\_lab es el horario de laboratorio. Ejemplo: **Perez-Gonzalez-L1**.
4. El reporte o informe del proyecto debe incluir lo siguiente:
  - a. **Portada (máximo un página):** con nombres, apellidos y cedula de identidad de cada miembro

del grupo y su horario de laboratorio.

- b. Introducción (máximo una página):** breve explicación del problema y de la solución, así como de los conceptos asociados a la solución del problema.
- c. Diseño de la solución (máximo cuatro páginas):** Descripción de la solución mediante un algoritmo pseudoformal de la solución basada en módulos (enfoque modular).
- d. Implementación de la solución (máximo cuatro páginas):** Programación modular usando subrutinas. Explicación y descripción del código fuente en lenguaje ensamblador, descripción de las variables y procedimientos (subrutinas) utilizados.
- e. Conclusiones (máximo una página):** Explicación de los resultados esperados, conceptos aprendidos que justifiquen los objetivos del proyecto.

**Nota:** cada cara de una hoja es considerada una página, así una hoja posee dos páginas.