



**DCC**  
DEPARTAMENTO DE CIENCIA  
DE LA COMPUTACIÓN

**IIC2343**

# **Arquitectura de Computadores**

**Clase 7 - Arquitecturas de Computadores**

**Profesor: Germán Leandro Contreras Sagredo**

## Objetivos de la clase

- Conocer los componentes principales de la arquitectura de un computador.
- Conocer distintos tipos de diseño para la arquitectura de un computador.
- Conocer variantes de la arquitectura del computador básico vista en el curso.

## Hasta ahora...

- Hemos visto cómo representar datos con números en base binaria (enteros y racionales).
- Hemos visto cómo traducir dicha representación en circuitos digitales.
- Hemos construido una máquina programable (correspondiente al computador básico del curso) con componentes digitales.

Ahora, ¿cómo se *categoriza* nuestro computador en la práctica?

## Arquitectura del computador básico

Nuestro computador cuenta con todas las funcionalidades **básicas**:

- Registros, unidades de ejecución y control.
- Realiza cálculos y operaciones de control de flujo (saltos).
- Modularidad limitada, pero existente (subrutinas).

No obstante, pueden existir arquitecturas **distintas**.

## Arquitectura del computador básico - Otras arquitecturas

Nuestro computador es solo una de varias arquitecturas posibles. Otras pueden diferir en **funcionalidades básicas y fundamentales**.

Por otra parte, existen computadores que se **programan de la misma manera** (Intel-AMD x86-64), pero con construcción interna distinta.

Las decisiones que inciden en la cantidad de registros, tamaño de buses de datos/direccionamiento, memorias e instrucciones **definen la arquitectura de un computador**.

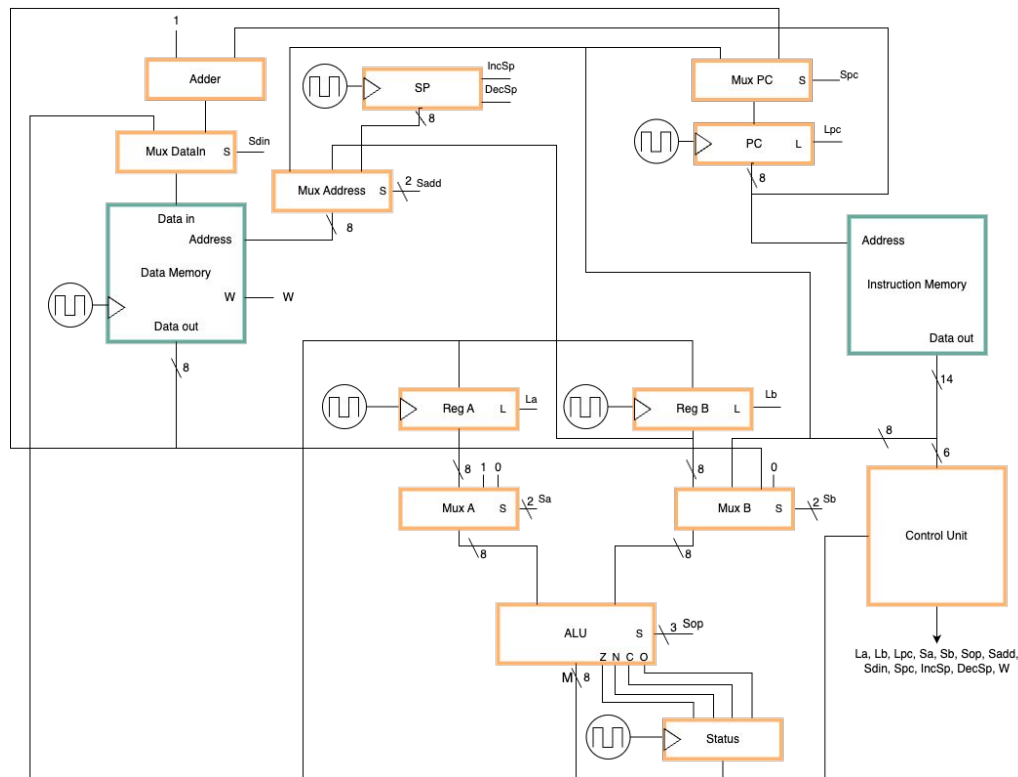
## Arquitectura del computador básico - Definiciones

La arquitectura de un computador se define en base a dos elementos:

- **Microarquitectura:** Componentes de *hardware* presentes en el computador.
- **Arquitectura del set de instrucciones (ISA):** *Instruction Set Architecture*. Tipo, formato y características de las instrucciones soportadas por el computador. Esto es lo referido a su **programación**. Podemos tener dos computadores con la misma microarquitectura pero distintas ISA (instrucciones que pueden estar soportadas en uno pero no en el otro).

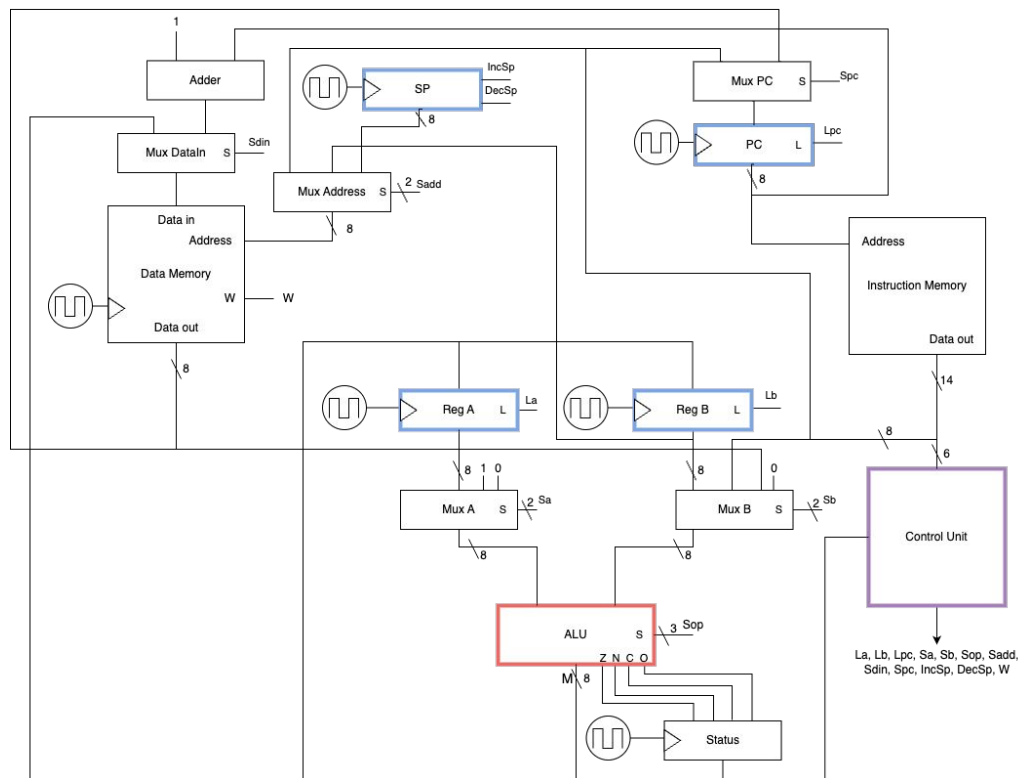
## Arquitectura del computador básico - Microarquitectura

En la microarquitectura de nuestro computador podemos separar el **procesador (CPU)** de las **memorias** del mismo.



## Arquitectura del computador básico - Microarquitectura

A su vez, podemos separar la CPU en lo que son sus **registros**, **unidad de ejecución** y **unidad de control**.





## Arquitectura del computador básico - Microarquitectura

Entonces, la microarquitectura de nuestro computador es así:

- **Registros:**  $A$ ,  $B$ ,  $PC$ ,  $SP$ .
- **Unidad de control:** Simple (*hardwired*).
- **Tamaños:** 8 bits (registros, direcciones de memoria).
- **Condition codes:**  $Z$ ,  $N$ ,  $O$ ,  $C$ .
- **Stack:** Dentro de la memoria de datos.

## Arquitectura del computador básico - Microarquitectura

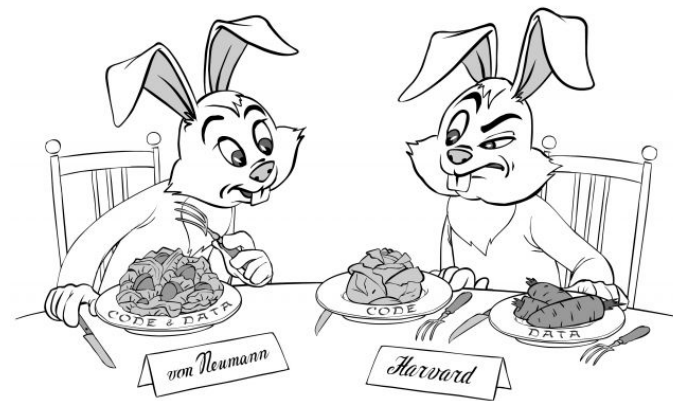
Nuestra microarquitectura puede modificarse de varias formas:

- Más registros ( $C$ ,  $D$ , etc.).
- Aumentar tamaños (registros  $\rightarrow$  números más grandes; direcciones  $\rightarrow$  memorias más grandes).
- Usar unidad de control microprogramada.
- Separar memoria de *stack* y trabajar con las dos separadas; usar solo una memoria de *stack*.

## Arquitectura del computador básico - Microarquitectura

Las memorias presentan dos paradigmas, que derivan en dos tipos de arquitectura:

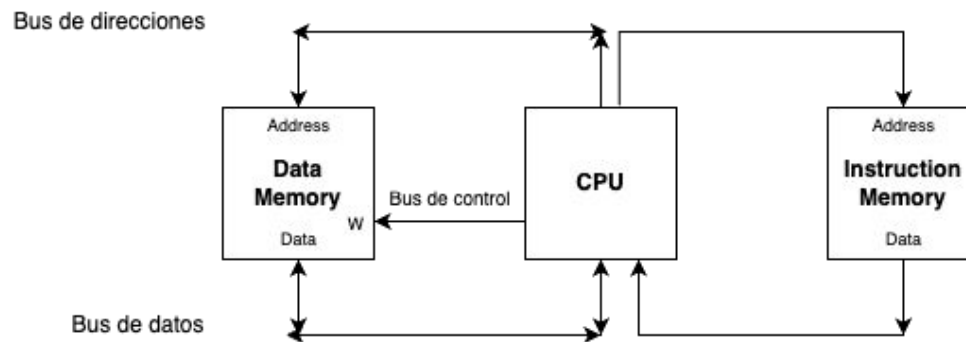
- **Arquitectura Harvard:** Memoria de datos e instrucciones **independientes**.
- **Arquitectura Von Neumann:** Memoria única que **comparte datos e instrucciones**. Permite escribir datos como si fueran instrucciones (**auto programabilidad**).



## Arquitectura del computador básico - Microarquitectura

Si nos abstraemos de la microarquitectura y la vemos por bloques de CPU y memoria, este es el comportamiento de nuestro computador.

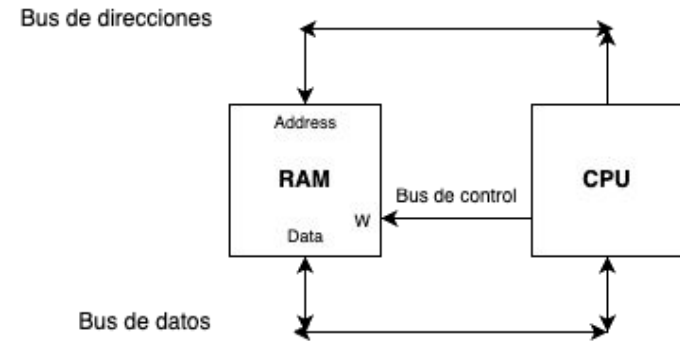
El bus de datos funciona de forma bidireccional entre la memoria de datos y la CPU, mientras que los datos de la memoria de instrucciones van en una única dirección a la CPU.



## Arquitectura del computador básico - Microarquitectura

En el caso de la arquitectura Von Neumann, el bus de instrucciones se vuelve parte del bus bidireccional de datos.

Como se señaló anteriormente, las instrucciones se tratan como si fueran datos, por lo que recorren las mismas conexiones que las variables almacenadas.



## Arquitectura del computador básico - Microarquitectura

¿Qué tipo de arquitectura presenta nuestro computador, en base a este paradigma?



## Arquitectura del computador básico - ISA

Por su parte, la ISA especifica **cómo** escribir los programas en el computador.

- Tipos de instrucciones: cargas, aritméticas, saltos, etc.
- Tipos de datos.
- Modos de direccionamiento de memoria.
- Manejo del *stack*.
- Formato de instrucción.
- Palabras por instrucción.
- Ciclos por instrucción.

## Arquitectura del computador básico - ISA

### ¡La ISA no es lo mismo que el código Assembly!

- Podemos tener dos computadores con la misma ISA pero con código en Assembly distinto.
- Por ejemplo:
  - `ADD A, B`  $\rightarrow A = A + B$
  - `SUM A, B`  $\rightarrow A = A + B$

Podemos tener dos instrucciones que deriven en las mismas señales de control, pero con nomenclatura distinta.



# Arquitectura del computador básico - ISA del curso

Instrucción	Operandos	Opcode	Condition	Lpc	La	Lb	Sa0,1	Sb0,1	Sop0,1,2	Sadd0,1	Sdin0	Spc0	W	IncSp	DecSp
MOV	A,B	0000000		0	1	0	ZERO	B	ADD	-	-	-	0	0	0
	B,A	0000001		0	0	1	A	ZERO	ADD	-	-	-	0	0	0
	A,Lit	0000010		0	1	0	ZERO	LIT	ADD	-	-	-	0	0	0
	B,Lit	0000011		0	0	1	ZERO	LIT	ADD	-	-	-	0	0	0
	A,(Dir)	0000100		0	1	0	ZERO	DOUT	ADD	LIT	-	-	0	0	0
	B,(Dir)	0000101		0	0	1	ZERO	DOUT	ADD	LIT	-	-	0	0	0
	(Dir),A	0000110		0	0	0	A	ZERO	ADD	LIT	ALU	-	1	0	0
	(Dir),B	0000111		0	0	0	ZERO	B	ADD	LIT	ALU	-	1	0	0
	A,(B)	0001000		0	1	0	ZERO	DOUT	ADD	B	-	-	0	0	0
	B,(B)	0001001		0	0	1	ZERO	DOUT	ADD	B	-	-	0	0	0
	(B),A	0001010		0	1	0	A	ZERO	ADD	B	ALU	-	1	0	0
ADD	A,B	0001011		0	1	0	A	B	ADD	-	-	-	0	0	0
	B,A	0001100		0	0	1	A	B	ADD	-	-	-	0	0	0
	A,Lit	0001101		0	1	0	A	LIT	ADD	-	-	-	0	0	0
	A,(Dir)	0001110		0	1	0	A	DOUT	ADD	LIT	-	-	0	0	0
	A,(B)	0001111		0	1	0	A	DOUT	ADD	B	-	-	0	0	0
	(Dir)	0010000		0	0	0	A	B	ADD	LIT	ALU	-	1	0	0
SUB	A,B	0010001		0	1	0	A	B	SUB	-	-	-	0	0	0
	B,A	0010010		0	0	1	A	B	SUB	-	-	-	0	0	0
	A,Lit	0010011		0	1	0	A	LIT	SUB	-	-	-	0	0	0
	A,(Dir)	0010011		0	1	0	A	DOUT	SUB	LIT	-	-	0	0	0
	A,(B)	0010100		0	1	0	A	DOUT	SUB	B	-	-	0	0	0
	(Dir)	0010101		0	0	0	A	B	SUB	LIT	ALU	-	1	0	0
AND	A,B	0010110		0	1	0	A	B	AND	-	-	-	0	0	0
	B,A	0010111		0	0	1	A	B	AND	-	-	-	0	0	0
	A,Lit	0011000		0	1	0	A	LIT	AND	-	-	-	0	0	0
	A,(Dir)	0011001		0	1	0	A	DOUT	AND	LIT	-	-	0	0	0
	A,(B)	0011010		0	1	0	A	DOUT	AND	B	-	-	0	0	0
	(Dir)	0011011		0	0	0	A	B	AND	LIT	ALU	-	1	0	0
OR	A,B	0011100		0	1	0	A	B	OR	-	-	-	0	0	0
	B,A	0011101		0	0	1	A	B	OR	-	-	-	0	0	0
	A,Lit	0011110		0	1	0	A	LIT	OR	-	-	-	0	0	0
	A,(Dir)	0011111		0	1	0	A	DOUT	OR	LIT	-	-	0	0	0
	A,(B)	0100000		0	1	0	A	DOUT	OR	B	-	-	0	0	0
	(Dir)	0100001		0	0	0	A	B	IR	LIT	ALU	-	1	0	0
NOT	A,A	0100010		0	1	0	A	-	NOT	-	-	-	0	0	0
	B,A	0100011		0	0	1	A	-	NOT	-	-	-	0	0	0
	(Dir)	0100111		0	0	0	A	B	NOT	LIT	ALU	-	1	0	0

# Arquitectura del computador básico - ISA del curso

Instrucción	Operandos	Opcode	Condition	Lpc	La	Lb	Sa0,1	Sb0,1	Sop0,1,2	Sadd0,1	Sdin0	Spc0	W	IncSp	DecSp
XOR	A,B	0100110		0	1	0	A	B	XOR	-	-	-	0	0	0
	B,A	0100111		0	0	1	A	B	XOR	-	-	-	0	0	0
	A,Lit	0101000		0	1	0	A	LIT	XOR	-	-	-	0	0	0
	A,(Dir)	0101001		0	1	0	A	DOUT	XOR	LIT	-	-	0	0	0
	A,(B)	0101010		0	1	0	A	DOUT	XOR	B	-	-	0	0	0
	(Dir)	0101011		0	0	0	A	B	XOR	LIT	ALU	-	1	0	0
SHL	A,A	0101100		0	1	0	A	-	SHL	-	-	-	0	0	0
	B,A	0101101		0	0	1	A	-	SHL	-	-	-	0	0	0
	(Dir)	0101110		0	0	0	A	B	SHL	LIT	ALU	-	1	0	0
SHR	A,A	0101111		0	1	0	A	-	SHR	-	-	-	0	0	0
	B,A	0110000		0	0	1	A	-	SHR	-	-	-	0	0	0
	(Dir)	0110001		0	0	0	A	B	SHR	LIT	ALU	-	1	0	0
INC	B	0110010		0	0	1	ONE	B	ADD	-	-	-	0	0	0
CMP	A,B	0110011		0	0	0	A	B	SUB	-	-	-	0	0	0
	A,Lit	0110100		0	0	0	A	LIT	SUB	-	-	-	0	0	0
JMP	Dir	0110101		1	0	0	-	-	-	-	-	LIT	0	0	0
JEQ	Dir	0110110	Z=1	1	0	0	-	-	-	-	-	LIT	0	0	0
JNE	Dir	0110111	Z=0	1	0	0	-	-	-	-	-	LIT	0	0	0
JGT	Dir	0111000	N=0 y Z=0	1	0	0	-	-	-	-	-	LIT	0	0	0
JLT	Dir	0111001	N=1	1	0	0	-	-	-	-	-	LIT	0	0	0
JGE	Dir	0111010	N=0	1	0	0	-	-	-	-	-	LIT	0	0	0
JLE	Dir	0111011	N=1 o Z=1	1	0	0	-	-	-	-	-	LIT	0	0	0
JCR	Dir	0111100	C=1	1	0	0	-	-	-	-	-	LIT	0	0	0
JOV	Dir	0111101	V=1	1	0	0	-	-	-	-	-	LIT	0	0	0
CALL	Dir	0111110		1	0	0	-	-	-	SP	PC	LIT	1	0	1
RET		0111111		0	0	0	-	-	-	-	-	-	0	1	0
		1000001		1	0	0	-	-	-	SP	-	DOUT	0	0	0
PUSH	A	1000010		0	0	0	A	ZERO	ADD	SP	ALU	-	1	0	1
PUSH	B	1000011		0	0	0	ZERO	B	ADD	SP	ALU	-	1	0	1
POP	A	0111111		0	0	0	-	-	-	-	-	-	0	1	0
POP		1000100		0	1	0	ZERO	DOUT	ADD	SP	ALU	-	0	0	0
		0111111		0	0	0	-	-	-	-	-	-	0	1	0
	B	1000101		0	0	1	ZERO	DOUT	ADD	SP	ALU	-	0	0	0

## Arquitectura del computador básico - ISA

Al igual que con las memorias, las ISA presenta dos grandes paradigmas:

- **RISC:** *Reduced Instruction Set Computer*. Instrucciones pequeñas y simples. Su diseño permite simplificar el *hardware*, poniendo énfasis en el *software*.
- **CISC:** *Complex Instruction Set Computer*. Muchas instrucciones y con complejidad alta. Énfasis en un *hardware* más complejo para poder ejecutarlas.

## Arquitectura del computador básico - ISA

¿Qué tipo de ISA posee nuestro computador?

- Tipos de instrucciones: cargas, aritméticas, saltos, etc.
- Tipos de datos: Números binarios con y sin signo.
- Modos de direccionamiento de memoria: Directo e indirecto por registro.
- Manejo del *stack*: General.
- Formato de instrucción: Mixto (0, 1 o 2 argumentos).
- Palabras por instrucción: 1 (salvo por RET y POP).
- Ciclos por instrucción: 1 (salvo por RET y POP).

## Arquitectura del computador básico - ISA

Nuestro computador, entonces, presenta una arquitectura de tipo **RISC**. ¿Cómo podríamos permitir que sea CISC? 🙄🙄

- Instrucciones más complejas: multiplicación, división, etc.
- Más modos de direccionamiento: direccionamiento indirecto por registro base, índice y *offset*.
- Manejo de *stack* por convención de llamada: `stdcall`, por ejemplo.
- Más de una palabra y ciclo por instrucción.

## Arquitectura del computador básico - ISA

¿Y en qué casos se suele usar cada una?

- Si bien CISC provee instrucciones más complejas que facilitan la programación, son menos eficientes en términos energéticos. Por este motivo se utilizan en dispositivos que pueden proveer la fuente energética requerida, como computadores de escritorio y *laptops*.
- Las ISA RISC, al requerir menos gasto energético, suelen usarse en dispositivos móviles.

## Arquitectura del computador básico - ISA

A nivel comercial, existen empresas que se dedican exclusivamente a diseñar ISAs.

- **ARM:** Empresa que se dedica al diseño de ISAs de bajo consumo energético (RISC). No fabrican CPUs, venden diseños de ISA y microarquitecturas. Así, otras empresas fabrican CPUs que corren bajo esta misma ISA, como ocurre con los celulares Android al correr con CPUs que soporten ISA ARM.



## Arquitectura del computador básico - ISA

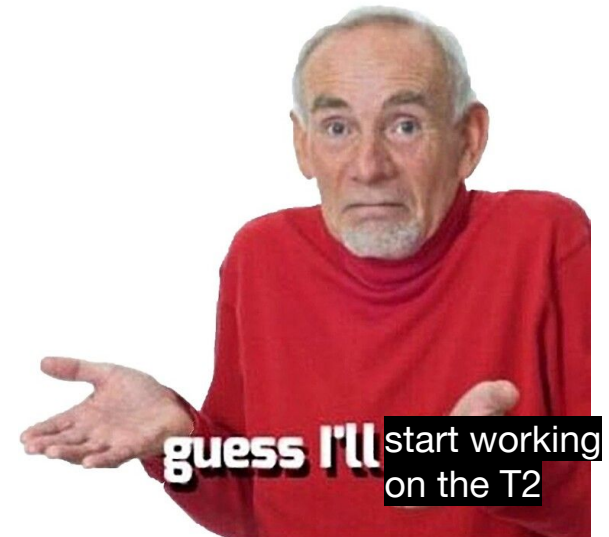
Un último tipo de ISA: **OISC** o *One Instruction Set Computer*.

- El número mínimo de instrucciones para definir un computador completo es uno, por lo que las ISA OISC solo poseen una.
- Ejemplo: **SUBLEQ**  $a, b, c$ 
  - $\text{Mem}[a] = \text{Mem}[a] - \text{Mem}[b]$
  - $\text{if } \text{Mem}[a] \leq 0 \rightarrow \text{JMP } c$
- Con la instrucción anterior podemos emular muchas de las instrucciones de nuestro computador básico.



## Ejercicios

Para este contenido las preguntas suelen ser más bien conceptuales, por lo que no hay nada que ejercitar para esta ocasión.



## Antes de terminar

¿Dudas?

¿Consultas?

¿Inquietudes?

¿Comentarios?





**DCC**  
DEPARTAMENTO DE CIENCIA  
DE LA COMPUTACIÓN

**IIC2343**

# **Arquitectura de Computadores**

**Clase 7 - Arquitecturas de Computadores**

**Profesor: Germán Leandro Contreras Sagredo**