



Componentes de la informática

.....

Carlos Augusto Arellano Muro

Unidad de Control y Registros

Es la encargada de organizar datos y ejecutar acciones del usuario o programas.

1. Ir a la memoria y extraer el código de la siguiente instrucción.

Unidad de Control y Registros

Es la encargada de organizar datos y ejecutar acciones del usuario o programas.

Ejecución de un programa:

1. Ir a la memoria y extraer el código de la siguiente instrucción.
2. Decodificar la instrucción.

Unidad de Control y Registros

Es la encargada de organizar datos y ejecutar acciones del usuario o programas.

Ejecución de un programa:

1. Ir a la memoria y extraer el código de la siguiente instrucción.
2. Decodificar la instrucción.
3. Ejecutar la instrucción.

Unidad de Control y Registros

Es la encargada de organizar datos y ejecutar acciones del usuario o programas.

Ejecución de un programa:

1. Ir a la memoria y extraer el código de la siguiente instrucción.
2. Decodificar la instrucción.
3. Ejecutar la instrucción.
4. Prepararse para leer la siguiente casilla de memoria y regresar al paso 1.

Unidad de Control y Registros

Es la encargada de organizar datos y ejecutar acciones del usuario o programas.

Ejecución de un programa:

Ciclo de Fetch o de máquina

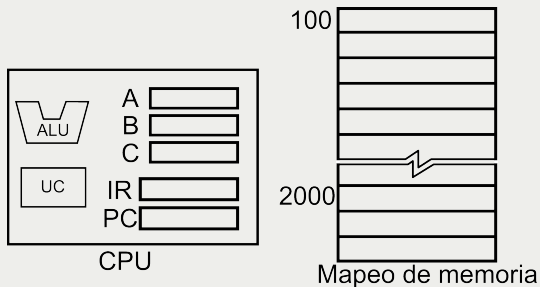
1. Ir a la memoria y extraer el código de la siguiente instrucción.
2. Decodificar la instrucción.
3. Ejecutar la instrucción.
4. Prepararse para leer la siguiente casilla de memoria y regresar al paso 1.

Unidad de Control y Registros

Ejecución de un programa: Ejemplo

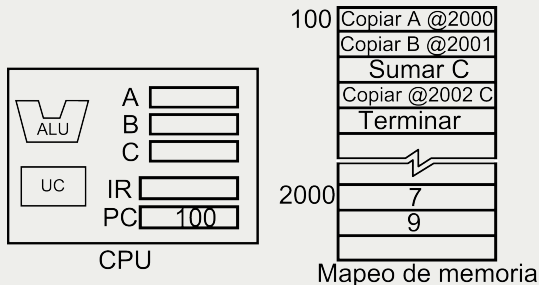
Unidad de Control y Registros

Ejecución de un programa: Ejemplo



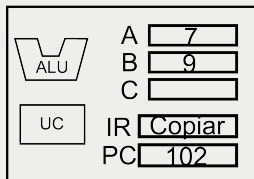
Unidad de Control y Registros

Ejecución de un programa: Ejemplo

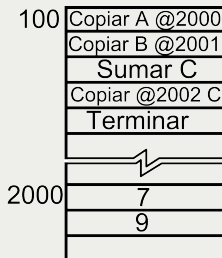


Unidad de Control y Registros

Ejecución de un programa: Ejemplo



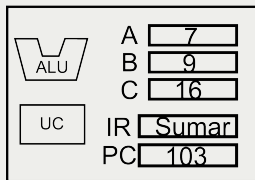
CPU



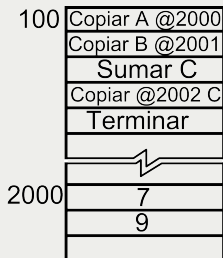
Mapecto de memoria

Unidad de Control y Registros

Ejecución de un programa: Ejemplo



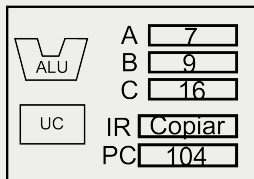
CPU



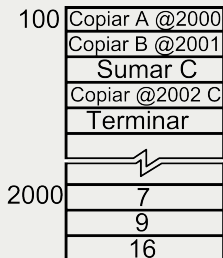
Mapeo de memoria

Unidad de Control y Registros

Ejecución de un programa: Ejemplo



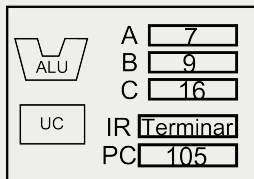
CPU



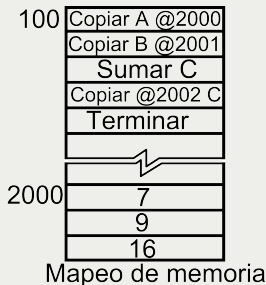
Mapeo de memoria

Unidad de Control y Registros

Ejecución de un programa: Ejemplo



CPU



Unidad de Control y Registros

La secuencia anterior necesita ser traducida a lenguaje máquina.

Unidad de Control y Registros

La secuencia anterior necesita ser traducida a lenguaje máquina. En el **traductor** se especifica la longitud en memoria de cada instrucción.

Unidad de Control y Registros

La secuencia anterior necesita ser traducida a lenguaje máquina. En el **traductor** se especifica la longitud en memoria de cada instrucción. Es decir, para Copiar se necesitan: 1. el código de copiado, 2. el lugar de destino, 3. lo que voy a copiar.

Unidad de Control y Registros

La secuencia anterior necesita ser traducida a lenguaje máquina. En el **traductor** se especifica la longitud en memoria de cada instrucción. Es decir, para Copiar se necesitan: 1. el código de copiado, 2. el lugar de destino, 3. lo que voy a copiar.

Ejemplo de traductor

Instrucción	Código	Longitud
Copiar	718	3
Sumar	111	2
Terminar	224	1
Restar	459	2
Multiplicar	129	2
Dividir	432	2

Unidad de Control y Registros

Otro ejemplo: mapeo de un programa en memoria

718	001	1344	718	002	1101	111	718	1000	@003	224
510	511	512	513	514	515	516	517	518	519	520

Unidad de Control y Registros

Otro ejemplo: mapeo de un programa en memoria

718	001	1344	718	002	1101	111	718	1000	@003	224
510	511	512	513	514	515	516	517	518	519	520

Esta es una representación de lenguaje máquina, la secuencia anterior es una representación de ensamblador.

Unidad de Control y Registros

Otro ejemplo: mapeo de un programa en memoria

718	001	1344	718	002	1101	111	718	1000	@003	224
510	511	512	513	514	515	516	517	518	519	520

Esta es una representación de lenguaje máquina, la secuencia anterior es una representación de ensamblador.

El lenguaje **ensamblador** permite la traducción de un programa fuente a un lenguaje máquina y utiliza mnemónicos como instrucciones:

Unidad de Control y Registros

Otro ejemplo: mapeo de un programa en memoria

718	001	1344	718	002	1101	111	718	1000	@003	224
510	511	512	513	514	515	516	517	518	519	520

Esta es una representación de lenguaje máquina, la secuencia anterior es una representación de ensamblador.

El lenguaje **ensamblador** permite la traducción de un programa fuente a un lenguaje máquina y utiliza mnemónicos como instrucciones:

- Add 1,2

Unidad de Control y Registros

Otro ejemplo: mapeo de un programa en memoria

718	001	1344	718	002	1101	111	718	1000	@003	224
510	511	512	513	514	515	516	517	518	519	520

Esta es una representación de lenguaje máquina, la secuencia anterior es una representación de ensamblador.

El lenguaje **ensamblador** permite la traducción de un programa fuente a un lenguaje máquina y utiliza mnemónicos como instrucciones:

- ▶ Add 1,2
- ▶ Mult 3,4

Unidad de Control y Registros

Otro ejemplo: mapeo de un programa en memoria

718	001	1344	718	002	1101	111	718	1000	@003	224
510	511	512	513	514	515	516	517	518	519	520

Esta es una representación de lenguaje máquina, la secuencia anterior es una representación de ensamblador.

El lenguaje **ensamblador** permite la traducción de un programa fuente a un lenguaje máquina y utiliza mnemónicos como instrucciones:

- ▶ Add 1,2
- ▶ Mult 3,4
- ▶ Copy A,12

Unidad de Control y Registros

Tipos de lenguaje ensamblador:

- ▶ Básico: Hace referencias al modo de direccionamiento de la memoria por medio de variables.

Unidad de Control y Registros

Tipos de lenguaje ensamblador:

- ▶ Básico: Hace referencias al modo de direccionamiento de la memoria por medio de variables.
- ▶ Macro: Ejecuta una serie de pasos para una acción. Se le llama ensamblador de una pasada.

Unidad de Control y Registros

Tipos de lenguaje ensamblador:

- ▶ Básico: Hace referencias al modo de direccionamiento de la memoria por medio de variables.
- ▶ Macro: Ejecuta una serie de pasos para una acción. Se le llama ensamblador de una pasada.
- ▶ 32 bits: Hace las operaciones de las dos anteriores. Se le llama ensamblador de dos pasadas o dos pasos

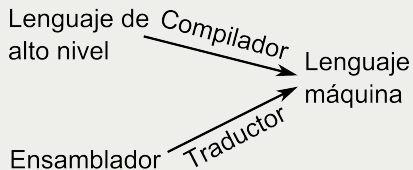
Unidad de Control y Registros

Macro: Segmento de código que permite ejecutarlo únicamente haciendo referencia a su nombre:

Macro suma

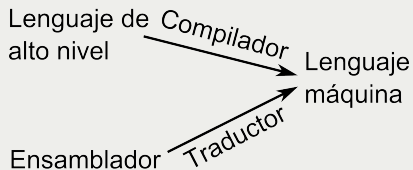
```
Copiar A, v1
Copiar B, v2
    Sumar
Copiar v3, C
    Fin
```

Lenguaje de alto nivel: posibilitan que los científicos, ingenieros y público en general resuelvan sus problemas mediante una terminología familiar, en lugar de usar enigmáticas instrucciones máquina.



Lenguaje de alto nivel: posibilitan que los científicos, ingenieros y público en general resuelvan sus problemas mediante una terminología familiar, en lugar de usar enigmáticas instrucciones máquina.

Para que una computadora entienda un programa escrito en uno de estos lenguajes, es necesario un compilador.



Tres estructuras de control:

Los científicos de la computación han definido tres estructuras de control para un programa o algoritmo estructurado:

1. Secuencia

Tres estructuras de control:

Los científicos de la computación han definido tres estructuras de control para un programa o algoritmo estructurado:

1. Secuencia
2. Decisión (elección)

Tres estructuras de control:

Los científicos de la computación han definido tres estructuras de control para un programa o algoritmo estructurado:

1. Secuencia
2. Decisión (elección)
3. Repetición

Tres estructuras de control:

Los científicos de la computación han definido tres estructuras de control para un programa o algoritmo estructurado:

1. Secuencia
2. Decisión (elección)
3. Repetición

► Secuencia

Tres estructuras de control:

Los científicos de la computación han definido tres estructuras de control para un programa o algoritmo estructurado:

1. Secuencia
2. Decisión (elección)
3. Repetición

- Secuencia
 - acción 1

Tres estructuras de control:

Los científicos de la computación han definido tres estructuras de control para un programa o algoritmo estructurado:

1. Secuencia
2. Decisión (elección)
3. Repetición

► Secuencia

acción 1

acción 2

Tres estructuras de control:

Los científicos de la computación han definido tres estructuras de control para un programa o algoritmo estructurado:

1. Secuencia
2. Decisión (elección)
3. Repetición

► Secuencia

acción 1
acción 2
acción 3

Tres estructuras de control:

- ▶ Decisión

Tres estructuras de control:

- ▶ Decisión

 - if** condición es verdadera

Tres estructuras de control:

- ▶ Decisión

- if** condición es verdadera

- then**

- realizar una serie de acciones

Tres estructuras de control:

► Decisión

if condición es verdadera

then

realizar una serie de acciones

else

realizar otra serie de acciones

Tres estructuras de control:

► Decisión

```
if condición es verdadera  
then  
    realizar una serie de acciones  
else  
    realizar otra serie de acciones  
end if
```

Tres estructuras de control:

► Decisión

if condición es verdadera

then

realizar una serie de acciones

else

realizar otra serie de acciones

end if

► Repetición

Tres estructuras de control:

- ▶ Decisión

 - if** condición es verdadera

 - then**

 - realizar una serie de acciones

 - else**

 - realizar otra serie de acciones

 - end if**

- ▶ Repetición

 - ▶ **while** condición verdadera

Tres estructuras de control:

► Decisión

if condición es verdadera

then

realizar una serie de acciones

else

realizar otra serie de acciones

end if

► Repetición

► **while** condición verdadera

realizar una secuencia de acciones

Tres estructuras de control:

► Decisión

```
if condición es verdadera  
then  
    realizar una serie de acciones  
else  
    realizar otra serie de acciones  
end if
```

► Repetición

```
► while condición verdadera  
    realizar una secuencia de acciones  
end while
```

Representación

- ▶ **Pseudocódigo:** Las representaciones anteriores son ejemplo de pseudocódigo.

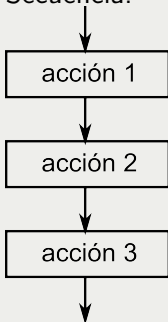
Representación

- ▶ **Pseudocódigo:** Las representaciones anteriores son ejemplo de pseudocódigo.
- ▶ **Diagrama de flujo:**

Representación

- ▶ **Pseudocódigo:** Las representaciones anteriores son ejemplo de pseudocódigo.
- ▶ **Diagrama de flujo:**

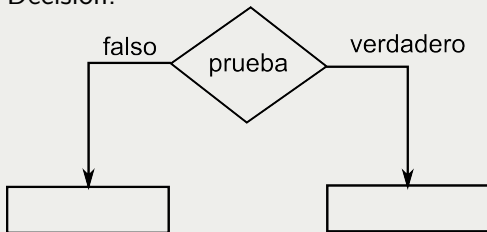
Secuencia:



Representación

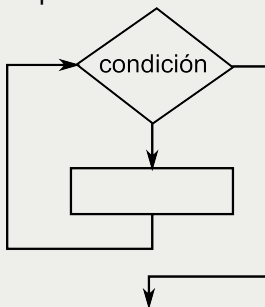
- ▶ **Pseudocódigo:** Las representaciones anteriores son ejemplo de pseudocódigo.
- ▶ **Diagrama de flujo:**

Decisión:



Representación

- ▶ **Pseudocódigo:** Las representaciones anteriores son ejemplo de pseudocódigo.
- ▶ **Diagrama de flujo:**
Repetición:



Ejemplos:

1. Escriba un algoritmo en pseudocódigo que encuentre el promedio de dos números.

Ejemplos:

1. Escriba un algoritmo en pseudocódigo que encuentre el promedio de dos números.

Importante:

Ejemplos:

1. Escriba un algoritmo en pseudocódigo que encuentre el promedio de dos números.

Importante:

- ▶ Nombrar el algoritmo.

Ejemplos:

1. Escriba un algoritmo en pseudocódigo que encuentre el promedio de dos números.

Importante:

- ▶ Nombrar el algoritmo.
- ▶ Numerar los pasos.

Ejemplos:

1. Escriba un algoritmo en pseudocódigo que encuentre el promedio de dos números.

Importante:

- ▶ Nombrar el algoritmo.
- ▶ Numerar los pasos.
- ▶ Definir la entrada.

Ejemplos:

1. Escriba un algoritmo en pseudocódigo que encuentre el promedio de dos números.

Importante:

- ▶ Nombrar el algoritmo.
- ▶ Numerar los pasos.
- ▶ Definir la entrada.
- ▶ Especificar la salida al finalizar.

Ejemplos:

1. Escriba un algoritmo en pseudocódigo que encuentre el promedio de dos números.

Importante:

- ▶ Nombrar el algoritmo.
- ▶ Numerar los pasos.
- ▶ Definir la entrada.
- ▶ Especificar la salida al finalizar.

PromedioDeDos

Entrada: Dos números

Ejemplos:

1. Escriba un algoritmo en pseudocódigo que encuentre el promedio de dos números.

Importante:

- ▶ Nombrar el algoritmo.
- ▶ Numerar los pasos.
- ▶ Definir la entrada.
- ▶ Especificar la salida al finalizar.

PromedioDeDos

Entrada: Dos números

1. Sumar los números

Ejemplos:

1. Escriba un algoritmo en pseudocódigo que encuentre el promedio de dos números.

Importante:

- ▶ Nombrar el algoritmo.
- ▶ Numerar los pasos.
- ▶ Definir la entrada.
- ▶ Especificar la salida al finalizar.

PromedioDeDos

Entrada: Dos números

1. Sumar los números
2. Dividir el resultado entre 2

Ejemplos:

1. Escriba un algoritmo en pseudocódigo que encuentre el promedio de dos números.

Importante:

- ▶ Nombrar el algoritmo.
- ▶ Numerar los pasos.
- ▶ Definir la entrada.
- ▶ Especificar la salida al finalizar.

PromedioDeDos

Entrada: Dos números

1. Sumar los números
2. Dividir el resultado entre 2
3. Devolver el resultado del paso 2

Ejemplos:

1. Escriba un algoritmo en pseudocódigo que encuentre el promedio de dos números.

Importante:

- ▶ Nombrar el algoritmo.
- ▶ Numerar los pasos.
- ▶ Definir la entrada.
- ▶ Especificar la salida al finalizar.

PromedioDeDos

Entrada: Dos números

1. Sumar los números
2. Dividir el resultado entre 2
3. Devolver el resultado del paso 2

end

Otro ejemplo:

2. Escriba un algoritmo para traducir una calificación numérica a una etiqueta de aprobado o reprobado:

Otro ejemplo:

2. Escriba un algoritmo para traducir una calificación numérica a una etiqueta de aprobado o reprobado:

Aprobado/Reprobado

Entrada: a

Otro ejemplo:

2. Escriba un algoritmo para traducir una calificación numérica a una etiqueta de aprobado o reprobado:

Aprobado/Reprobado

Entrada: a

1. **if**($a \geq 6$)

Otro ejemplo:

2. Escriba un algoritmo para traducir una calificación numérica a una etiqueta de aprobado o reprobado:

Aprobado/Reprobado

Entrada: a

1. **if**($a \geq 6$)
 then

Otro ejemplo:

2. Escriba un algoritmo para traducir una calificación numérica a una etiqueta de aprobado o reprobado:

Aprobado/Reprobado

Entrada: a

1. **if**($a \geq 6$)
 then
2. $cal \leftarrow$ "aprobado"

Otro ejemplo:

2. Escriba un algoritmo para traducir una calificación numérica a una etiqueta de aprobado o reprobado:

Aprobado/Reprobado

Entrada: a

1. **if**($a \geq 6$)
 then
2. $cal \leftarrow$ "aprobado"
 else

Otro ejemplo:

2. Escriba un algoritmo para traducir una calificación numérica a una etiqueta de aprobado o reprobado:

Aprobado/Reprobado

Entrada: a

1. **if**($a \geq 6$)
 then
2. $cal \leftarrow$ "aprobado"
- else**
3. $cal \leftarrow$ "reprobado"

Otro ejemplo:

2. Escriba un algoritmo para traducir una calificación numérica a una etiqueta de aprobado o reprobado:

Aprobado/Reprobado

Entrada: a

1. **if**($a \geq 6$)
 then
2. $cal \leftarrow \text{"aprobado"}$
 else
3. $cal \leftarrow \text{"reprobado"}$
 end if

Otro ejemplo:

2. Escriba un algoritmo para traducir una calificación numérica a una etiqueta de aprobado o reprobado:

Aprobado/Reprobado

Entrada: a

1. **if**($a \geq 6$)
 then
2. $cal \leftarrow$ "aprobado"
 else
3. $cal \leftarrow$ "reprobado"
 end if
4. Devuelve cal

Otro ejemplo:

2. Escriba un algoritmo para traducir una calificación numérica a una etiqueta de aprobado o reprobado:

Aprobado/Reprobado

Entrada: a

1. **if**($a \geq 6$)
 then
2. $cal \leftarrow$ "aprobado"
 else
3. $cal \leftarrow$ "reprobado"
 end if
4. Devuelve cal

end

Uno más:

3. Escriba un algoritmo para encontrar el número mayor en una lista sin conocer la cantidad en ella:

Uno más:

3. Escriba un algoritmo para encontrar el número mayor en una lista sin conocer la cantidad en ella:

EncontrarMayor

Entrada: lista de números

Uno más:

3. Escriba un algoritmo para encontrar el número mayor en una lista sin conocer la cantidad en ella:

EncontrarMayor

Entrada: lista de números

1. $M \leftarrow 0$

Uno más:

3. Escriba un algoritmo para encontrar el número mayor en una lista sin conocer la cantidad en ella:

EncontrarMayor

Entrada: lista de números

1. $M \leftarrow 0$
2. **while**(exista números en lista)

Uno más:

3. Escriba un algoritmo para encontrar el número mayor en una lista sin conocer la cantidad en ella:

EncontrarMayor

Entrada: lista de números

1. $M \leftarrow 0$
2. **while**(exista números en lista)
3. **if**(número $> M$)

Uno más:

3. Escriba un algoritmo para encontrar el número mayor en una lista sin conocer la cantidad en ella:

EncontrarMayor

Entrada: lista de números

1. $M \leftarrow 0$
2. **while**(exista números en lista)
3. **if**(número > M)
 then

Uno más:

3. Escriba un algoritmo para encontrar el número mayor en una lista sin conocer la cantidad en ella:

EncontrarMayor

Entrada: lista de números

1. $M \leftarrow 0$
2. **while**(exista números en lista)
3. **if**(número $> M$)
 then
4. $M \leftarrow \text{número}$

Uno más:

3. Escriba un algoritmo para encontrar el número mayor en una lista sin conocer la cantidad en ella:

EncontrarMayor

Entrada: lista de números

1. $M \leftarrow 0$
2. **while**(exista números en lista)
3. **if**(número $> M$)
 then
4. $M \leftarrow \text{número}$
 end if

Uno más:

3. Escriba un algoritmo para encontrar el número mayor en una lista sin conocer la cantidad en ella:

EncontrarMayor

Entrada: lista de números

1. $M \leftarrow 0$
2. **while**(exista números en lista)
3. **if**(número $> M$)
 then
4. $M \leftarrow \text{número}$
 end if
- end while**

Uno más:

3. Escriba un algoritmo para encontrar el número mayor en una lista sin conocer la cantidad en ella:

EncontrarMayor

Entrada: lista de números

1. $M \leftarrow 0$
2. **while**(exista números en lista)
3. **if**(número $> M$)
 then
4. $M \leftarrow \text{número}$
 end if
- end while**
5. Devuelve M

Uno más:

3. Escriba un algoritmo para encontrar el número mayor en una lista sin conocer la cantidad en ella:

EncontrarMayor

Entrada: lista de números

1. $M \leftarrow 0$
 2. **while**(exista números en lista)
 3. **if**(número $> M$)
 then
 4. $M \leftarrow \text{número}$
 end if
 - end while**
 5. Devuelve M
- end**

Ejercicio:

Escriba el pseudocódigo para encontrar el promedio de una lista de números. Se desconoce la cantidad de números

Ejercicio:

Escriba el pseudocódigo para encontrar el promedio de una lista de números. Se desconoce la cantidad de números

Tarea:

Escribir en diagrama de flujo el algoritmo de ordenamiento por selección, ordenamiento burbuja y ordenamiento por inserción.

Aplicaciones ofimáticas

Procesadores de texto básico

- ▶ Bloc de notas
- ▶ WordPad
- ▶ Microsoft Word
 - ▶ Formato de párrafos (Alineación, justificación, tabulaciones, sangrías)
 - ▶ Formato de texto
 - ▶ Formato de documento (Títulos, Encabezado y pie de página)

Aplicaciones ofimáticas

Otros procesadores de texto

- ▶ \LaTeX (Overleaf)
- ▶ Markdown (Haroopad)
- ▶ Markup

