

# Repasemos

---

¿Que vimos la clase pasada?

# Modelo de Von Neumann

# Modelo de Von Neumann

¿Dónde se guardan las instrucciones?

# Modelo de Von Neumann

¿Dónde se guardan las instrucciones?

¿Cómo se guardan las instrucciones?

# Modelo de Von Neumann

¿Dónde se guardan las instrucciones?

¿Cómo se guardan las instrucciones?

¿Cómo distinguimos cada instrucción al estar en código máquina (binario)?

# Modelo de Von Neumann

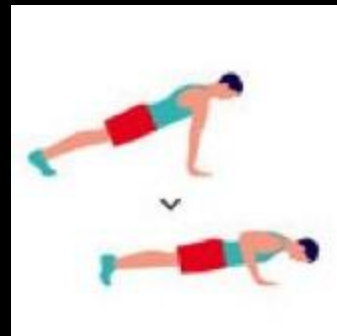
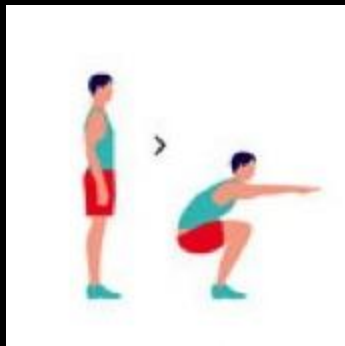
¿Dónde se guardan las instrucciones?

¿Cómo se guardan las instrucciones?

¿Cómo distinguimos cada instrucción al estar en código máquina (binario)?

Formato de instrucción

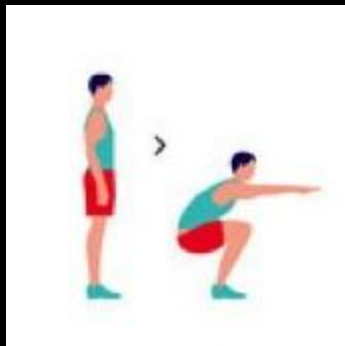
# Formato de instrucción



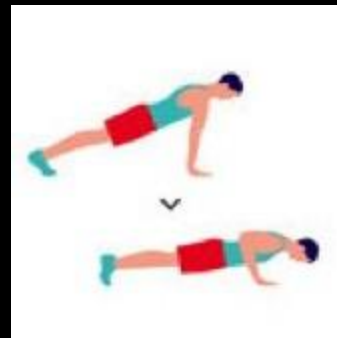
# Formato de instrucción



00



01



10



1

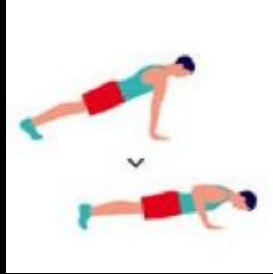
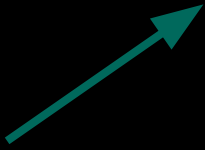


# Formato de instrucción

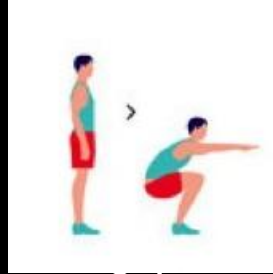
¿1001?

# Formato de instrucción

¿1001?



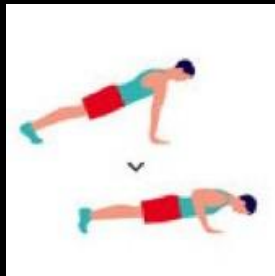
10



01

# Formato de instrucción

¿1001?



10



01



1

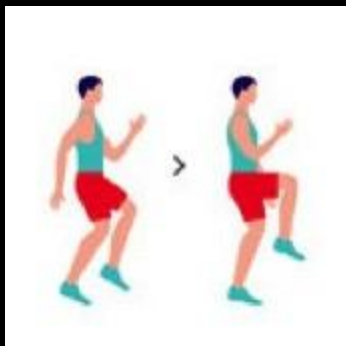


00

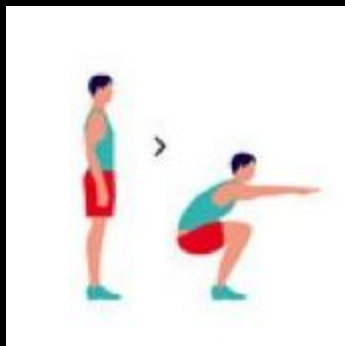


1

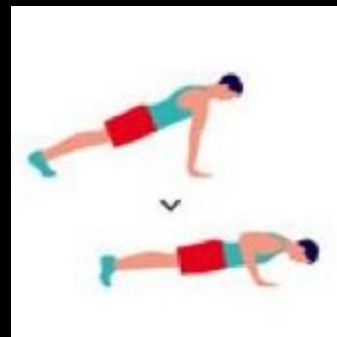
# Formato de instrucción



00



01



10

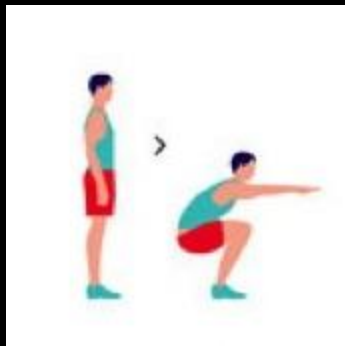


1

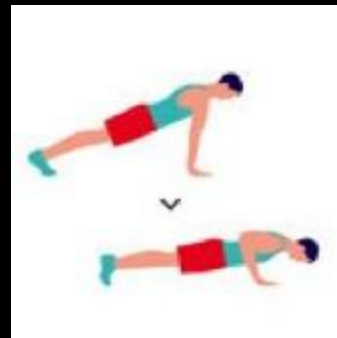
# Formato de instrucción



00



01



10



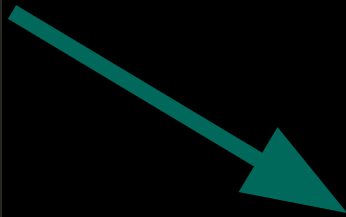
11

```
1 module(mutex).  
2 -export([create/0, lock/1, unlock/1]).  
3 create() ->  
4     spawn(fun() -> unlocked() end).  
5  
6 lock(Mutex) ->  
7     Mutex ! {lock, self()},  
8     receive  
9         ok -> ok  
10    end.  
11  
12 unlock(Lock) ->  
13     Lock ! {unlock, self()}.  
14  
15 unlocked() ->  
16     receive {lock, LockedBy} ->  
17         LockedBy ! ok,  
18         locked(LockedBy, 1)  
19     end.  
20
```

```
1 module(mutex).
2   -export([create/0, lock/1, unlock/1]).
3   create() ->
4     spawn(fun() -> unlocked() end).
5
6   lock(Mutex) ->
7     Mutex ! {lock, self()},
8     receive
9       ok -> ok
10    end.
11
12   unlock(Lock) ->
13     Lock ! {unlock, self()}.
14
15   unlocked() ->
16     receive {lock, LockedBy} ->
17       LockedBy ! ok,
18       locked(LockedBy, 1)
19    end.
```



```
1 module(mutex).
2   -export([create/0, lock/1, unlock/1]).
3   create() ->
4     spawn(fun() -> unlocked() end).
5
6   lock(Mutex) ->
7     Mutex ! {lock, self()},
8     receive
9       ok -> ok
10    end.
11
12   unlock(Lock) ->
13     Lock ! {unlock, self()}.
14
15   unlocked() ->
16     receive {lock, LockedBy} ->
17       LockedBy ! ok,
18       locked(LockedBy, 1)
19     end.
```





Ensamblar

```
1 module(mutex).  
2 -export([create/0, lock/1, unlock/1]).  
3 create() ->  
4     spawn(fun() -> unlocked() end).  
5  
6 lock(Mutex) ->  
7     Mutex ! {lock, self()},  
8     receive  
9         ok -> ok  
10    end.  
11  
12 unlock(Lock) ->  
13     Lock ! {unlock, self()}.  
14  
15 unlocked() ->  
16     receive {lock, LockedBy} ->  
17         LockedBy ! ok,  
18         locked(LockedBy, 1)  
19     end.  
20
```



```
1 module(mutex).  
2 -export([create/0, lock/1, unlock/1]).  
3 create() ->  
4     spawn(fun() -> unlocked() end).  
5  
6 lock(Mutex) ->  
7     Mutex ! {lock, self()},  
8     receive  
9         ok -> ok  
10    end.  
11  
12 unlock(Lock) ->  
13     Lock ! {unlock, self()}.  
14  
15 unlocked() ->  
16     receive {lock, LockedBy} ->  
17         LockedBy ! ok,  
18         locked(LockedBy, 1)  
19     end.  
20
```

Ensamblar

Desensamblar



# Arquitecturas Q: Q1

# Arquitecturas Q: Q1

Instrucciones:

# Arquitecturas Q: Q1

Instrucciones:

MUL, MOV, ADD, SUB, DIV

# Arquitecturas Q: Q1

Instrucciones:

MUL, MOV, ADD, SUB, DIV

Operandos (Modos de direccionamiento):

# Arquitecturas Q: Q1

Instrucciones:

MUL, MOV, ADD, SUB, DIV

Operandos (Modos de direccionamiento):

Registro (modo registro)

# Arquitecturas Q: Q1

Instrucciones:

MUL, MOV, ADD, SUB, DIV

Operandos (Modos de direccionamiento):

Registro (modo registro)

Constante (modo inmediato )



# Arquitecturas Q: Q1

Operación	Código
MUL	0000
MOV	0001
ADD	0010
SUB	0011
DIV	0111

Modo	Código
Registro	100RRR
Inmediato	000000

Cod Op (4 bits)	Modo destino (6 bits)	Modo origen (6 bits)	Origen (16 bits)
--------------------	--------------------------	-------------------------	---------------------

# Arquitecturas Q: Q1

Operación	Código
MUL	0000
MOV	0001
ADD	0010
SUB	0011
DIV	0111

Modo	Código
Registro	100RRR
Inmediato	000000

Cod Op (4 bits)	Modo destino (6 bits)	Modo origen (6 bits)	Origen (16 bits)
--------------------	--------------------------	-------------------------	---------------------

**Ensamblar:** MOV R3, 0xFFEF; ADD R5, R3

# Arquitecturas Q: Q1

Operación	Código
MUL	0000
MOV	0001
ADD	0010
SUB	0011
DIV	0111

Modo	Código
Registro	100RRR
Inmediato	000000

Cod Op (4 bits)	Modo destino (6 bits)	Modo origen (6 bits)	Origen (16 bits)
--------------------	--------------------------	-------------------------	---------------------

**Ensamblar:** MOV R3, 0xFFEF; ADD R5, R3

**Desensamblar:** 0000 1000 0110 0000  
0010 1000 0000 0000  
0000 0000 0000 0111