convenciones	
$zero	siempre tiene el valor 0
$ra	dirección de retorno de subrutina
$v0 - $v1	valores de retorno de una subrutina
$a0 - $a3	argumentos para pasarle a una subrutina
$t0 - $t9	registros temporales
$s0 - $s7	registros que deben ser salvados
$sp	stack pointer - puntero tope de pila
$fp	frame pointer - puntero de pila
$at	reservado para ser salvado por el ensamblador
$k0 - $k1	para uso kernel del sistema operativo
$gp	global pointer - puntero zona de memoria

etapas winmips	
if	instruction fetch - toma de la instrucción
id	instruction decoding - decodificacion de la instrucción
ex	execution - ejecución de la instrucción
mem	memory access - acceso a memoria
wb	write back - almacena resultado en registro (de ser necesario)

atascos	
estructurales	*provocados por conflictos de recursos, suelen ocurrir cuando tenemos dos instrucciones que quieren acceder a la etapa **mem** simultáneamente *tiene *propridad la instrucción que entró primero al cauce* (empezó primero)
dependencia de datos	***RAR:** *read after write* - se produce cuandoo queremos leer un dato que aún no está disponible (aún no se guardó) ***WAR:** *write after read* - quiere escribir un registro pendiente de lectura ***WAW:** *write after write* - quiere escribir algo que todavía no fue escrito
dependencia de control	*cuando quedan en espera del salto de la instrucción anterior

atascos.1	
forwarding	permite postergar la necesidad del operando, lo cuál le da tiempo a la próxima instruccion
delay slot	siempre ejecuta la instrucción siguiente al salto, reordenando instrucciones para que no dependan del salto podemos ahorrarle tiempo para que este se calcule
branch target buffer	consiste en tener un flag que indica si debemos saltar incondicionalmente, dependiendo de lo que hizo la instrucción anterior. (predice con **misprediction**) *solo falla la primera y última vez*

manejo de la pila (manual)	
$sp	daddi $sp,$zero,0x400
push	daddi $sp,$sp,-8 sd $t1,0($sp)
pop	ld $t1,0($sp) daddi $sp,$sp,8
retorno subrutina	jr $ra = ret

entrada / salida	
control	**data**
1	números sin signo
2	números con signo
3	punto flotante
4	direccion del string
5	color y coordenadas
6	limpia pantalla texto
7	limpia pantalla grafica
8	entrada de numeros (l.d o ld)
9	entrada caracter
control:	*.word 0x10000*
data:	*.word 0x10008*
debemos guardarnos las direcciones, y despues cargarlas en registros $s	