

```
arreglo global punteoP
arreglo global punteoG
arreglo global nombres
var numeroJug
var llamadaTabla=0
```

INICIO -Principal-

```
var opcion=-1
var bienIngresados=-1
```

```
Si args tiene tamaño >0
    Si args tiene tamaño 1
        bienIngresado=0
        Escribir “ Solo debe ingresar un número como máximo”
    fin si
    Si el valor de args es mayor 3 y menor a 1
        bienIngresado=0
        Escribir “Debe ingresar números entre 1 y 3, incluyéndolos
    fin si
    si no
        Hacer args numero de tipo entero
    fin si no
fin si
```

```
Si bienIngresado==1
```

```
Mientras opcion!=0
    //escribir el formato del menú
    Si opcion===-1
        Escribir “Ingresa numero de opción a ejecutar
        Leer (opcion)
    fin si

    si no
        Si opcion==0
            //escribir msje de agradecimiento por haber usado el programa
        Fin si

        Si opcion ==1
            llamar juego1()
        Fin si
        Si opcion ==2
            llamar juego2()
        Fin si
        Si opcion==3
            llamar juego3()
        Fin si
```

```

    Si opcion ==4
        llamar tabla_Puntuaciones()
    Fin si
    De lo contrario
        Escribir “Debe ingresar números entre 0 y 4”
    Fin de de lo contrario
fin si no
Fin mientras

Fin si

FIN

```

INICIO -juego1-

```

//las importaciones irán en el main, por ello aquí no es necesario colocarlas
var nombre
var palabraOriginal
arreglo String palabraDesordenada
var palabra
var ubicacionAleatoria
var ubicación[ ]
var control=0
var tamañoPalabra
var continuar=1
var iteraciones=0
var intentos=3
var gano=0

```

```

Escribir “Ingrese su nombre”
Leer(nombre)

```

```

Mientras continuar sea igual a 1
    Escribir “Ingrese la palabra a desordenar”
    Leer (palabraOriginal)
    Si palabraOriginal es igual a “m”
        entonces continuar = llamar a menuSecundario(continuar)
    Fin si
    sino, si continuar es igual a 1//esto es para evitar que ejecute el código que ejecuta el juego si
        //desde el ingreso de la palabra desea salir

```

```

Mientras i < tamaño de la palabra
    asignar ubicaciónAleatoria valor de Random (según tamaño de la palabra ingresada)
    ubicación = tamaño [tamañoPalabra]
    ubicación[i] =ubicaciónAleatoria
    palabraDesordenada = tamaño [tamaño de palabraOriginal]

```

```

    Si iteraciones!=0
        Para int x=0 hasta x< iteraciones hacer x++

```

```

        si ubicación==ubicacionALeatoria;
            control++;
        Fin si
        si control==0
            asignar palabraDesordenada[x], letra de palabra en ubicación
            iteraciones++
        Fin si
        control=0
    Fin Para
Fin si
Fin Mientras

Mientras intentos>0 y gano==0 y continuar ==1

    entonces
        Para x=0 hasta x<tamañoPalabra hacer x++
            Escribir palabraDesordenada[x]
        Fin Para
        Escribir “Ingrese la palabra que sea correcta”
        Leer (palabra)
        //hacer minúscula la palabra, para evitar conflictos innecesarios

        Si palabra es igual a “m”
            entonces continuar = llamar menuSecundario(continuar)
        Fin si
        sino, si continuar ==1
            si palabra=palabraOriginal
                gano=1
                Escribir “FELICIDADES HA GANADO!”
                Llamar acomodador(1, nombre, gano)
            Sino
                intentos - -
                Escribir “No ha ingresado la palabra correcta”
                si intentos==0
                    Escribir “A perdido el juego”
                    Llamar acomodador(1, nombre, gano)
                Fin si
            Fin sino
        Fin si
    Fin Mientras
Fin sino
FIN

```

INICIO -juego2-

arreglo jugadores
arreglo valorTiro
arreglo gano[4]

```
var numJugadores=0
var tipoTiro
var valor
var continuar=1
```

crear objeto (valorAleatorio) para número aleatorio

```
Mientras numJugadores >4 o numJugadores <0, ejecutar
    Escribir “Ingresa el número de jugadores”
    leer (numJugadores)
    Si numJugadores>4 o numJugadores <0
        Escribir “Ingresa numeros de 1 a 4”
    Fin si
Fin mientras
```

```
jugadores = tamaño [numJugadores]
valorTiro = tamaño [numJugadores]
```

```
Para x=0 hasta x<numJugadores hacer <x++
    Escribir “Ingresa tu nombre”
    Leer (jugadores)
Fin para
```

```
Mientras gano[0]==0 y ... gano[3]==0 y continuar ==1, ejecutar
    Para x=0 hasta x<numJugadores hacer x++
        Mientras tipoTiro !=0 y ... tipoTiro !=4 y continuar ==1
            Mostrar jugadores y su punteo respectivo
            Escribir “Ingrese el tipo de tiro (jugadores[x])”
            Leer (tipoTiro)

            Si tipoTiro==0
                entonces continuar = Llamar menuSecundario(continuar)
                Si continuar==1
                    tipoTiro=-1
                Fin si
            Fin si
        Fin mientras

    Si continuar == 1
        Si tipoTiro==1
            valor=40*(valorAleatorio(2))
        Fin si
        si no, si tipoTiro==2
            valor=10*(1+ valorAleatorio(3))
        Fin si
        si no, si tipoTiro==3
            valor=10*(valorAleatorio(5))
        Fin sino
```

```
valorTiro[x]= valorTiro+valor
Escribir "A obtenido puntos en este tiro" (valor)
Escribir "Lleva un total de %s puntos" (valorTiro]
```

```
Si valorTiro[x]>=200
```

```
    Escribir "Felicidades, ganaste el juego"
```

```
    gano[x]=1
```

```
    x=numeroJugadores
```

```
    Para z=0 hasta z<numeroJugadores hacer z++
```

```
        llamar acomodador(numeroJugadores, jugadores[z], gano[z])
```

```
    Fin para
```

```
Fin si
```

```
    tipoTiro=-1
```

```
Fin para
```

```
Fin mientras
```

```
FIN
```

```
INICIO -juego3-
```

```
arreglo numeros
```

```
int aleaFila= -1
```

```
int aleaCol=-1
```

```
var numeroAleatorio
```

```
var aleatorioFila
```

```
var aleatorioColumna
```

```
var gano=0
```

```
var opMenu=0
```

```
var filaCambiante
```

```
var filaFija
```

```
var columnaCambiante
```

```
var columnaFija
```

```
var suma=0
```

```
var sumaTotal=0
```

```
var cantidad
```

```
var contadorCeros=0
```

```
var coincidenciasY=1
```

```
var coincidenciasX=1
```

```
var continuar=1
```

```
var dir
```

```
var direccion="nada"
```

```
var nombre
```

```
Escribir "Ingresa tu nombre"
```

```
Leer (nombre)
```

//mostrar las instrucciones del juego

Para x=0 hasta x<2 hacer x=x+0
asignar a aleatorioFila valor aleatorio entre 0 y 3
asignar a aleatorioCol valor aleatorio entre 0 y 3

Si aleaFila!=aleatorioFila y aleaCol!=aleatorioColumna
 aleaFila=aleatorioFila
 aleaCol=aleatoroCol
 arreglo[aleatorioFila][aleatorioCol]=2
 x++

Fin si

Mientras gano==0 y continuar==1
 cantidad=1

Mostrar título y puntuación

Para filas=0 hasta filas <4 hacer filas++

Para columnas=0 hasta columnas<4 hacer columnas++

 Mostrar arreglo[filas][columnas]

Fin para

Si coincidenciasY==0 y coincidenciasX==0 y contadorCeros==0

Escribir LOSIENTO HAS PERDIDO EL JUEGO :(

Mostrar puntuación alcanzada (sumaTotal)

llamar acomodador(1, nombre, gano)

 continuar=0

Fin si

Si no

Mientras (dirección == nada)

Escribir “Ingresa la dirección en la que desea mover los cuadros”

Leer (dir)

 //Hacer a dirección igual a dir para luego convertirla en minúsculas

Si direccion ==”m”

 continuar = *llamar* menuSecundario(continuar)

Si continuar=1

 opMenu =1

Fin si

Fin si

Si dirección==”ab”

Si coincidenciasY==0

Escribir Intente con direcciones HORIZONTALES :)

Fin si

Si no

```
Para columna=0 hasta columna<4 hacer columna++  
    filaCambiante=2  
    filaFija=3
```

```
Mientras filaCambiante>=0 y filaFija>=1, ejecutar
```

```
    Si arreglo[filaFija][columna]==0
```

```
        filaFija- -
```

```
    Fin si
```

```
    Si no
```

```
        si arreglo[filaFija][columna]==arreglo[filaCambiante][columna]
```

```
            suma=arreglo[filaFija][columna]+arreglo[filaCambiante][columna]
```

```
            sumaTotal=sumaTotal+suma
```

```
            arreglo[filaFija][columna]=suma
```

```
            arreglo[filaCambiante][columna]=0
```

```
            suma=0
```

```
            filaFija- -
```

```
        Fin si
```

```
    Si no arreglo[filaFija][columna]!=arreglo[filaCambiante][columna] y
```

```
    arreglo[filaCambiante][columna]!=0
```

```
        filaFija- -
```

```
    Fin si no
```

```
    Fin si no
```

```
    filaCambiante--
```

```
    Fin mientras
```

```
    Fin para
```

```
Para col=0 hasta col<4 hacer col++
```

```
    Para fila=2 hasta fila>=0 hacer fila- -
```

```
        var filaUbic= fila +1
```

```
        Mientras filaUbic < 4 y arreglo[filaUbic][col]==0
```

```
            arreglo[filaUbic][col]=arreglo[filaUbic-1][col]
```

```
            arreglo[filaUbic-1][col]=0
```

```
        filaUbic++
```

```
        Fin mientras
```

```
    Fin para
```

```
    Fin para
```

```
    Fin si no
```

```
    Fin si
```

```
Si dirección == "ar"
```

```
    /*ejecutar lo mismo que en ab solo que los límites o valores asignados, serían los  
    contrarios a los que en ese bloque aparece, por ir en la orientación contraria*/
```

```
    Fin si
```

```
Si dirección == "de"
```

/*Hacer lo mismo que en ab, con los mismo límites o valores, solo que en donde aparecen variables que engañan que ver con columnas, irán filas y donde filas columnas, esto por el hecho de ir en una dirección diferente*/

Fin si

Si dirección == "iz"

/*Hacer lo mismo que en ar solo que al igual que en el bloque de arriba, hay que intercambiar las variables de columnas y filas*/

Fin si

Si no

Escribir "Debes ingresar cualquiera de las abreviaturas mostradas con anterioridad"

direccion = "nada" //por la condición del while

Fin si no

Si continuar == 1

direccion = "nada"

contadorCeros=0

Para c=0 hasta c<4 hacer c++

Para f=0 hasta f<4 hacer f++

Si arreglo[f][c]==0

contadorCeros++

Fin si

Si arreglo[f][c]==2048

gano=1

Fin si

Fin para

Fin para

Si contadorCeros!=0 y opMenu==0

Mientras cantidad<2, *ejecutar*

numeroAleatorio=1+ aleatorio(de 0 a 3)

aleatorioFila=aleatorio(de 0 a 3)

aleatorioCol=aleatorio(de 0 a 3)

Si arreglo[aleatorioFila][aleatorioCol]==0 y (numeroAleatorio%2==0)

arreglo[aleatorioFila][aleatorioCol]=numeroAleatorio

cantidad++

Fin si

Fin mientras

Fin si

coincidenciasX=0

coincidenciasY=0

Para col=0 hasta col<4 hacer col++

Para f=1 hasta f<4 hacer f++

var fa= f-1

Si arreglo[f][col]==arreglo[fa][col]

coincidenciasY++


```

        Fin si
    Fin para
Fin para

Para fil=0 hasta fil<4 hacer fil++
    Para c=1 hasta c<4 hacer c++
        var ca=c-1
        Si arreglo[fil][c]==arreglo[fil][ca]
            coincidenciasX++
        Fin si
    Fin para
Fin para

```

```

        Fin si //con condición de que continuar =1
    Fin si no
Fin mientras

```

```

Si gano=1
    Escribir "Felicidades, ha ganado el juego"
    llamar acomodador(1, nombre, gano)
Fin si

```

FININICIO -menuSecundario(continuar)-

```

var opcionCorrecta=0

```

```

Mientras opcionCorrecta==0, ejecutar
    Escribir "Ingrese el numero correspondiente a la opción"
    Escribir "1. continuar          0.salir"
    Leer (continuar)

    Si continuar==1 || continuar==0
        entonces opciónCorrecta=1
    Fin si
Fin mientras

```

```

return continuar

```

FIN

INICIO -acomodador (numeroJ; nombre; ganados)-

```

var nombreLocal[] =[tamaño numeroJug]
var ganadasLocal[] =[tamaño numeroJug]
var perdidasLocal [] =[tamaño numeroJug]
var tamActual=numeroJug

```

Si llamadaTabla>0

Para x=0 hasta x<numeroJug hacer x++
 nombreLocal[x]=nombres[x]
 ganadasLocal[x]=punteoG[x]
 perdidasLocal[x]=punteoP[x]

Fin para

Fin si

numeroJug+=numeroJ

nombres = [tamaño numeroJug]

punteoP= [tamaño numeroJug]

punteoG= [tamaño numeroJug]

Si llamadaTabla>0

Para y=0 hasta y < tamañoActual hacer y++
 nombres[y]=nombreLocal[y]
 punteoP[y]= perdidasLocal[y]
 punteoG[y]=ganadasLocal[y]

Fin del para

Fin si

Si ganados==0

Para x=0 hasta x<numJug hacer x++

Si punteoP[x]==0 y punteoG[x]==0

 nombres[x]=nombre

 punteoP[x]+=1

 x=numeroJug

Fin si

si no

Si nombres[x]==nombre

 punteoP[x]=punteoP[x]+1

 x=numeroJug

Fin si

Fin sino

Fin para

Fin si

Si no

Para x=0 hasta x<numJug hacer x++

SI punteo[x]==0 y punteoG[x]==0

 nombres[x]=nombre

 punteoG[x]+=ganados

 x=numeroJug

Fin si

si no

Si nombres[x]==nombre

 punteoG[x]+=ganados

 x=numeroJug

Fin si

Fin sino

Fin para
Fin si no

llamadaTabla++

FIN

INICIO -tabla_Puntuaciones-
var seguir =-1

Mientras seguir!=0
//mostrar encabezados y subtítulos

Para x=0 hasta x<numeroJug hacer x++
Escribir nombres[x], punteoG[x], punteoP[x]
Fin para

Escribir “0. Salir”
Leer (seguir)

Si seguir!=0
Escribir “solo puede ingresar 0”
Fin si

FIN