

```

1  Proceso Juego_Trivia_Ver_05_Fecha_05_22_MainMenu
2      // El presente proceso utiliza la consola para permitir a dos jugadores competir en un juego de trivia. Alterna e
3      // Cada respuesta correcta suma puntos
4      // Quien obtiene más puntos al final del juego, es el ganador.
5      // Al finalizar, es muestra una pantalla con el nombre del ganador.
6
7      Definir jugador_1, jugador_2 Como Caracter;
8      Definir puntuacion_jugador_1, puntuacion_jugador_2 como Entero;
9      Definir ganador como Caracter;
10     Definir dificultad como Entero;
11     Definir tec como caracter;
12
13     puntuacion_jugador_1 ← 0;
14     puntuacion_jugador_2 ← 0;
15     Definir preguntas como Caracter;
16     Dimension preguntas[15,6]; // Reemplazar por la cantidad de preguntas en preguntas
17
18     Definir finalizado como Logico;
19
20     finalizado ← Falso;
21     dificultad ← 0;
22
23     Escribir " _____ ";
24     Escribir " ^ _ ^ ^ = \ ^ \ ^ \ / \ ^ \ ^ _ \ ";
25     Escribir " V ^ V \ \ _ < \ \ \ \ \ / \ \ \ \ \ _ \ ";
26     Escribir " \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ ";
27     Escribir "  V \  V \ / \  V \  V \  V \  V \ V \ V \ ";
28     Escribir " ";
29     Escribir " _____ ";
30     Escribir " ^ _ ^ ^ ^ _ ^ _ ^ _ ^ _ ^ _ ^ _ ^ _ ^ _ ^ _ ";
31     Escribir " \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ ";
32     Escribir " \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ ";
33     Escribir "  V \ \  V \ V \  V \  V \  V \ \  V \ \  ";
34     Escribir " ";
35     Escribir "          Presione enter para continuar ";
36     Leer tec;
37
38     nombres_jugadores(jugador_1, jugador_2);
39     establecer_dificultad(dificultad);
40
41     Limpiar Pantalla;
42     importar_preguntas(preguntas);
43
44     Definir i como Entero;
45
46     Mientras finalizado ≠ Verdadero Hacer
47         Para i ← 0 hasta 15-1 Hacer // Reemplazar por la cantidad de preguntas en preguntas
48             preguntar(i, preguntas, jugador_1, jugador_2, puntuacion_jugador_1, puntuacion_jugador_2, dificultad);
49             Limpiar Pantalla;
50         FinPara
51         finalizado ← Verdadero;
52     FinMientras
53     ganador ← determinar_ganador(jugador_1, puntuacion_jugador_1, jugador_2, puntuacion_jugador_2, ganador);
54     Escribir ganador;
55
56 FinProceso
57

```

SUBPROCESO nombre jugadores

```
57 SubProceso nombres_jugadores(jugador_1 Por Referencia, jugador_2 Por Referencia)
58 // Pregunta los nombres de cada usuario y lo almacena en las variables jugador_1 y jugador_2
59 Limpiar Pantalla;
60 Escribir "Indique el nombre del jugador 1:";
61 Leer jugador_1;
62 Escribir "Indique el nombre del jugador 2:";
63 Leer jugador_2;
64 FinSubProceso
```

SUBPROCESO establecer dificultad

```
SubProceso establecer_dificultad (dificultad Por Referencia)
// Pregunta al usuario entre dos opciones para establecer qué matriz de preguntas serán mostradas durante el juego
Limpiar Pantalla;
Definir opcion como entero;
Definir A Como Caracter;
a="";
Escribir "Seleccione la dificultad: normal (1) y difícil (2)";
Leer a;
Mientras a≠"1" y a≠ "2" Hacer
    Escribir "Intente nuevamente";
    Leer a;
Si a="1" Entonces
    dificultad ← 1;
FinSi
Si a="2" Entonces
    dificultad ← 2;
FinSi
FinMientras
FinSubProceso
```

SUBPROCESO importar preguntas

```
SubProceso importar_preguntas(preguntas Por Referencia)
// Llenado Manual de las preguntas
// Importar desde el Excel generador de variables
// https://docs.google.com/spreadsheets/d/1ic0G1E6JdyICTNDp9h13GmCFWuQlFowx530RUP18s
// La ultima columna muestra la respuesta correcta
preguntas(0,0) ← 'CATEGORIA: GEOGRAFIA ¿De qué país forma parte Hawaii?'; preguntas(0,1) ← '[1] Estados Unidos'; preguntas(0,2) ← '[2] Argentina'; preguntas(0,3) ← '[3] Brasil'; preguntas(0,4) ← '[4] Ecuador'; preguntas(0,5) ← '[1]';
preguntas(1,0) ← 'CATEGORIA: GEOGRAFIA ¿Cuántos estados tiene integrados Estados Unidos?'; preguntas(1,1) ← '[1] 49'; preguntas(1,2) ← '[2] 15'; preguntas(1,3) ← '[3] 50'; preguntas(1,4) ← '[4] 180'; preguntas(1,5) ← '[3]';
preguntas(2,0) ← 'CATEGORIA: GEOGRAFIA ¿De qué año es la Constitución Española?'; preguntas(2,1) ← '[1] 1999'; preguntas(2,2) ← '[2] 1978'; preguntas(2,3) ← '[3] 1979'; preguntas(2,4) ← '[4] 1848'; preguntas(2,5) ← '[2]';
preguntas(3,0) ← 'CATEGORIA: GEOGRAFIA ¿Cuál es el río más largo de España?'; preguntas(3,1) ← '[1] El río Ebro'; preguntas(3,2) ← '[2] Río Tamesis'; preguntas(3,3) ← '[3] Río Amazonas'; preguntas(3,4) ← '[4] Río Grande'; preguntas(3,5) ← '[1]';
preguntas(4,0) ← 'CATEGORIA: GEOGRAFIA ¿Cuál es el océano más grande del mundo?'; preguntas(4,1) ← '[1] El océano Atlántico'; preguntas(4,2) ← '[2] El océano Pacífico'; preguntas(4,3) ← '[3] El océano Índico'; preguntas(4,4) ← '[4] El océano Ártico'; preguntas(4,5) ← '[1]';
preguntas(5,0) ← 'CATEGORIA: INGLES Modals Verbs of Obligation'; preguntas(5,1) ← '[1] Mustn't'; preguntas(5,2) ← '[2] Have to'; preguntas(5,3) ← '[3] Should'; preguntas(5,4) ← '[4] Dont have to'; preguntas(5,5) ← '[2]';
preguntas(6,0) ← 'CATEGORIA: INGLES Modals Verbs of Prohibition'; preguntas(6,1) ← '[1] Mustn't'; preguntas(6,2) ← '[2] Have to'; preguntas(6,3) ← '[3] Should'; preguntas(6,4) ← '[4] Dont have to'; preguntas(6,5) ← '[1]';
preguntas(7,0) ← 'CATEGORIA: INGLES Modals Verbs of Recommendation, advice'; preguntas(7,1) ← '[1] Mustn't'; preguntas(7,2) ← '[2] Have to'; preguntas(7,3) ← '[3] Should'; preguntas(7,4) ← '[4] Dont have to'; preguntas(7,5) ← '[3]';
preguntas(8,0) ← 'CATEGORIA: INGLES Modals Verbs (lack of necessity)'; preguntas(8,1) ← '[1] Mustn't'; preguntas(8,2) ← '[2] Have to'; preguntas(8,3) ← '[3] Should'; preguntas(8,4) ← '[4] Dont have to'; preguntas(8,5) ← '[4]';
preguntas(9,0) ← 'CATEGORIA: INGLES Modals Verbs of Deduction'; preguntas(9,1) ← '[1] Mustn't'; preguntas(9,2) ← '[2] Have to'; preguntas(9,3) ← '[3] Should'; preguntas(9,4) ← '[4] Dont have to'; preguntas(9,5) ← '[1]';
preguntas(10,0) ← 'CATEGORIA: PROGRAMACION ¿Esta variable true es un dato: ?'; preguntas(10,1) ← '[1] String'; preguntas(10,2) ← '[2] Boolean'; preguntas(10,3) ← '[3] Double'; preguntas(10,4) ← '[4] integer'; preguntas(10,5) ← '[2]';
preguntas(11,0) ← 'CATEGORIA: PROGRAMACION ¿Cuantos es el alcance de una variable dentro de una función?'; preguntas(11,1) ← '[1] Global'; preguntas(11,2) ← '[2] Local'; preguntas(11,3) ← '[3] Mundial'; preguntas(11,4) ← '[4] Otro'; preguntas(11,5) ← '[2]';
preguntas(12,0) ← 'CATEGORIA: PROGRAMACION ¿esta variable está escrita en: mPrimerVariable?'; preguntas(12,1) ← '[1] kebab case'; preguntas(12,2) ← '[2] Snake case'; preguntas(12,3) ← '[3] Pascal case'; preguntas(12,4) ← '[4] Came case'; preguntas(12,5) ← '[3]';
preguntas(13,0) ← 'CATEGORIA: PROGRAMACION ¿Los datos Simple son: ?'; preguntas(13,1) ← '[1] Logicos'; preguntas(13,2) ← '[2] Registros'; preguntas(13,3) ← '[3] Arreglos'; preguntas(13,4) ← '[4] Archivos'; preguntas(13,5) ← '[1]';
preguntas(14,0) ← 'CATEGORIA: PROGRAMACION ¿Este dato de variable=10.5f es: ?'; preguntas(14,1) ← '[1] Integer'; preguntas(14,2) ← '[2] Boolean'; preguntas(14,3) ← '[3] Float'; preguntas(14,4) ← '[4] String'; preguntas(14,5) ← '[3]';
FinSubProceso
```

SUBPROCESO mostrar pregunta individual

```
SubProceso mostrar_pregunta_individual(preguntas Por Referencia, nro_pregunta Por Referencia)
// Determina la posición de la matriz (pregunta)
Definir j como Entero;
Para j ← 0 hasta 6-2 Hacer
    Escribir preguntas[nro_pregunta,j];
FinPara
FinSubProceso
```

SUBPROCESO preguntar

```
SubProceso preguntar(nro_pregunta, preguntas, jugador_1 Por Referencia, jugador_2
Por Referencia, puntuacion_jugador_1 Por Referencia, puntuacion_jugador_2
Por Referencia, dificultad Por Referencia)
// Muestra el número de pregunta indicado en la variable nro_pregunta y permite a cada jugador
// contestar entre una de las 4 opciones disponibles.
// Luego de cada respuesta, la función comprobar_respuesta compara con la respuesta correcta y
// le suma la cantidad de puntos correspondientes al jugador

Definir respuesta Como Entero;
Definir rpta Como Caracter;
mostrar_pregunta_individual(preguntas, nro_pregunta);
rpta-"";
Escribir "Turno de ", jugador_1, " para responder la pregunta";
Leer rpta;

Mientras rpta ≠ "1" y rpta ≠ "2" y rpta ≠ "3" y rpta ≠ "4" Hacer
    Escribir "Intente nuevamente";
    Leer rpta;
FinMientras
respuesta-ConvertirANumero(rpta);
puntuacion_jugador_1 ← puntuacion_jugador_1 + comprobar_respuesta(preguntas, nro_pregunta, respuesta, dificultad);
Limpiar Pantalla;

mostrar_pregunta_individual(preguntas, nro_pregunta);

Escribir "Turno de ",jugador_2 , " para responder la pregunta";
Leer rpta;
Mientras rpta ≠ "1" y rpta ≠ "2" y rpta ≠ "3" y rpta ≠ "4" Hacer
    Escribir "Intente nuevamente";
    Leer rpta;
FinMientras
respuesta-ConvertirANumero(rpta);
puntuacion_jugador_2 ← puntuacion_jugador_2 + comprobar_respuesta(preguntas, nro_pregunta, respuesta, dificultad);

FinSubProceso
```

SUBPROCESO premio

```
SubProceso premio ← comprobar_respuesta (preguntas, nro_pregunta, respuesta, dificultad)
// Determina si la respuesta correcta.

// Devuelve:
// premio (Entero)

// DIFICULTAD NORMAL:
// 20 (Entero) Si la respuesta es correcta
// 0 (Entero) Si la respuesta es incorrecta

// DIFICULTAD DIFÍCIL:
// 20 (Entero) Si la respuesta es correcta
// -10 (Entero) Si la respuesta es incorrecta

Definir premio como Entero;

Si respuesta = ConvertirANumero(preguntas[nro_pregunta, 5]) Entonces
    premio ← 20;
SiNo
    Si dificultad = 2 entonces
        premio ← -10;
    SiNo
        premio ← 0;
    FinSi
FinSi
// Escribir "Contestó", respuesta;
// Escribir "Respuesta correcta", preguntas[nro_pregunta, 5];

FinSubProceso
```

SUBPROCESO ganador

[illegible]

FinSubProceso

Proceso	Juego_Trivia_Ver_05_Fecha_05_22_MainMenu
SubProceso	nombres_jugadores(jugador_1 Por Referencia)
SubProceso	establecer_dificultad (dificultad Por Referencia)
SubProceso	importar_preguntas(preguntas Por Referencia)
SubProceso	mostrar_pregunta_individual(preguntas Por Referencia)
SubProceso	preguntar(nro_pregunta, preguntas, jugador_1)
SubProceso	premio ← comprobar_respuesta (preguntas, jugador_1)
SubProceso	ganador ← determinar_ganador(jugador_1, premio)

