



Trabajo individual

Identificación del trabajo

- a. **Módulo:** 4
- b. **Asignatura:** Lógica de Programación
- c. **RA:** Agregar condicionales al código.
- d. **Docente Online:** Richard Orellana Taibo.
- e. **Fecha de entrega:** 18/05/25

Identificación del estudiante

Nombre y apellido	Carrera
Christian Aránguiz	Técnico en Informática

Desarrollo

Código extraído desde PseInt:

```
Proceso juegoAdivinanzas
    Definir limiteMinimo, limiteMaximo, numeroCreado,
    numerosIngresados Como Entero;
    Definir numeroIngresado, maxIntentos, intentosRestantes,
    turnos Como Entero;
    Definir isAdivinado Como Logico;

    // Establecer límites
    Escribir "**** Bienvenido al juego de Adivinanzas ****";
    Escribir "Ingrese número mínimo para adivinar";
    Leer limiteMinimo;
    Escribir "Ingrese número máximo para adivinar";
    Leer limiteMaximo;

    // revisar que el limite inferior no sea mayor al superior
    si limiteMaximo < limiteMinimo Entonces
        Escribir "ERROR ==> El número máximo no puede
        ser menos al mínimo";
    SiNo
        // Pedir número máximo de intentos
        Escribir "Ingrese cantidad máxima de intentos";
        Leer maxIntentos;

        Dimensionar numerosIngresados[maxIntentos];
        intentosRestantes <- maxIntentos;

        Escribir "Adivina el número entre: ", limiteMinimo, " y:
        ", limiteMaximo;

        numeroCreado <- obtenerNumero(limiteMinimo,
        limiteMaximo);

        turnos <- 1;

        isAdivinado <- Falso;

        // Actualizacion Para => Repetir
        Repetir
            Escribir "Ingresa que número crees que es el
            correcto. Tienes: ", intentosRestantes " intentos Restantes";
            Leer numeroIngresado;
            Si revisionDuplicados(numerosIngresados,
            turnos, numeroIngresado) Entonces
```

```

                                Escribir "El número ya fue ingresado,
has usado estos intentos: ";
                                listarNumeros(numerosIngresados,
turnos);
                                SiNo
                                    numerosIngresados[turnos] <-
numeroIngresado;
                                Escribir "Número ingresado: ",
numeroIngresado, " es...";
                                // Revision si número ingresado es
válido entre el rango de números
                                Si numeroIngresado > limiteMaximo
Entonces
                                    Escribir "Error ==> El número
ingresado no puede ser superior al numero máximo a adivinar";
                                    SiNo
                                        si numeroIngresado <
limiteMinimo Entonces
                                            Escribir "Error ==> El
número ingresado no puede ser inferior al numero mínimo a
adivinar";
                                            SiNo
                                                si numeroIngresado <
numeroCreado Entonces
                                                    Escribir "Incorrecto!
El número es mayor. Te quedan ", intentosRestantes, "
intentos";
                                                    FinSi
                                                        si numeroIngresado >
numeroCreado Entonces
                                                            Escribir "Incorrecto!
El número es menor. Te quedan ", intentosRestantes, "
intentos";
                                                            FinSi
                                                                si numeroIngresado ==
numeroCreado Entonces
                                                                    Escribir "Correcto!
Adivinaste el número";
                                                                    isAdivinado <-
Verdadero;
                                                                    FinSi
                                                                        intentosRestantes <-
intentosRestantes -1;
                                                                        turnos = turnos +1;
                                                                        FinSi
                                                                            FinSi
                                                                                FinSi

```

```

        Hasta Que intentosRestantes == 0 o isAdivinado ==
Verdadero
        si isAdivinado == Verdadero Entonces
            Escribir "Ganaste en : ", turnos -1, " intentos.";
        SiNo
            Escribir "Perdiste, el número era:",
numeroCreado;
        FinSi
        mostrarIntentosOrdenados(numerosIngresados,
turnos);
        FinSi
FinProceso

SubProceso resultado <- obtenerNumero ( min, max )
    Definir rango, num, resultado Como Entero;
    Definir random Como Real;
    rango <- max - min + 1;
    random <- Azar(rango) ;
    resultado <- random + min;
FinSubProceso

Funcion listarNumeros (listaDeNumeros, turnos)
    Definir iteracion Como Entero;
    Escribir Sin Saltar " [";
    Para iteracion <- 1 Hasta turnos -1 Hacer
        Escribir Sin Saltar " ", listaDeNumeros[iteracion];
    FinPara
    Escribir " ]";
FinFuncion

Funcion duplicado <- revisionDuplicados (listaDeNumeros,
turnos, nuevo)
    Definir iteracion Como Entero;
    Definir duplicado Como Logico;
    duplicado <- Falso;
    si turnos > 1 Entonces
        para iteracion <- 1 Hasta turnos -1 Hacer
            si nuevo == listaDeNumeros[iteracion]
Entonces
                duplicado <- Verdadero;
            FinSi
        FinPara
    FinSi
FinFuncion

Funcion mostrarIntentosOrdenados(intentos, cantidad)

```

```

    Definir ordenActual, busquedaMinimo, numeroMenor,
    valorIntercambio Como Entero;
    // Ordenar solo si hay elementos
    si cantidad > 0 Entonces
        Para ordenActual <- 0 Hasta cantidad - 2 Hacer
            numeroMenor <- ordenActual;
            Para busquedaMinimo <- ordenActual + 1
                Hasta cantidad - 1 Hacer
                    si intentos[busquedaMinimo] <
                    intentos[numeroMenor] Entonces
                        numeroMenor <-
                        busquedaMinimo;
                    FinSi
                FinPara
            // Intercambiar valores (corregido)
            valorIntercambio <- intentos[ordenActual];
            intentos[ordenActual] <-
            intentos[numeroMenor];
            intentos[numeroMenor] <- valorIntercambio;
        FinPara
    FinSi

    // Mostrar números ordenados
    Escribir "Números ingresados ordenados:";
    Para ordenActual <- 0 Hasta cantidad - 1 Hacer
        Escribir intentos[ordenActual];
    FinPara
FinFuncion

```