Imagen que contiene Logotipo

Descripción generada automáticamente

Imagen que contiene Logotipo

Descripción generada automáticamente

**Alumno:** Armendariz Lopez José Jaime

**Grado y Grupo:** 8° - G

**Hora:** 10:00 – 11:00 Am

**Materia:** Diseño electrónico basado en Sistemas Embebidos

# Índice

[Índice 1](#_Toc84132705)

[Prácticas 2](#_Toc84132706)

[Unidad 1 2](#_Toc84132707)

[Práctica 1. Piedra, Papel o Tijera 2](#_Toc84132708)

[Práctica 2. Adivinar numero de la computadora 6](#_Toc84132709)

## Prácticas

## Unidad 1

### Práctica 1. Piedra, Papel o Tijera

**Descripción**: El programa simula el juego de piedra papel o tijera haciendo que el usuario seleccione una de las 3 opciones y la maquina selecciona uno al azar, y dependiendo lo elegido por el usuario se compara con lo elegido por la maquina y determina si gano, perdió o fue empate

**Introducción**:

Para el desarrollo de este programa se utilizaron como instrucciones especiales:

* Switch
* IF
* Random

Primero se utilizó la instrucción Switch para crear los casos y poder navegar entre el menú principal y las 3 diferentes respuestas que del usuario.

Con el IF lo utilice para hacer comparación entre la respuesta del usuario y la de la maquina y con esto así la maquina podría decidir cual de los dos es el ganador o si fue un empate. También se utilizó el método random esto con el fin de que la maquina elija una opción entre: piedra papel o tijera, al mas puro azar, las variables que se utilizaron para almacenar la decisión del usuario y la maquina son simples int del 1 al 3 los cuales cada uno significa la opción de: piedra (1), papel (2), Tijera (3). El hacer la elección de esta manera se permite un fácil uso para el usuario y una escritura de código más corta.

También use varios delay esto con el fin de dar algo de tiempo en la decisión del usuario y se lleve un poco mas lento el mostrado de la información. También usamos las instrucciones de entrada y salida para recibir como dato de entrada la elección del usuario y como datos de salida el resultado de si es ganador o las instrucciones de cómo funciona el programa

**Desarrollo:**

Primero en el programa lo que hicimos fue declarar los programas que utilizaremos cada uno con su correspondiente tipo dependiendo el uso que se le dará y el tipo de información que almacenará en este caso

Texto

Descripción generada automáticamente

Después nos dirigimos al método loop en el cual realizamos todas las instrucciones a seguir del programa.

En este lugar lo que se hizo fue usar un switch con el que podremos movernos entre los diferentes case que creemos, y poder regresar al menú que seria el case 0.

Imagen que contiene Texto

Descripción generada automáticamente

En el caso principal pusimos el menú donde se desarrolla la parte principal del programa. Primero con los Serial.println ponemos unos diálogos para darle la bienvenida al usuario y darle las instrucciones del programa. Después con un readString leemos el dato que ingreso el usuario y lo convertimos en dato de tipo entero para la variable.

Después de eso damos un pequeño tiempo de elección antes de que prosiga el programa ya que es algo difícil pararlo, y ya aquí veremos el por que es importante el haber usado el switch y los diferentes case. Ya que dependiendo de la elección del usuario el numero elegido se pasará a la variable estado el cual hará que nos envié al caso específico. Por ejemplo si elegimos piedra (1) este numero se pasará a la variable estado y provocara que nos lleve al case 1 y esto pasara con los 3 diferentes case.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Tomando como ejemplo que el usuario seleccionara el numero 1 que es la opción de piedra este nos llevaría al Case 1, el cual desarrolla las siguientes cosas:

La variable (a) la utilizaremos para seleccionar un numero al azar del rango 1 al 3 (se ve que hay un 4 en el código pero en verdad no lo toma en cuenta y solo marca del 1 al 3 – si por alguna razón quisiéramos que también estuviera el 4 sería del 1 al 5.)

Con esto pasamos el dato a la variable (elección\_maquina) y como ya sabemos en este case que el usuario eligió piedra no sirve de nada comparar la elección de la maquina con el del usuario, así que simplemente ponemos en el código:

If(elección\_maquina==1){}

Esta línea solo comparara la elección de la maquina con los números 1,2 y 3. Y como por default tenemos seleccionados a 1 (que es piedra), 2(que es papel) y 3 (que es tijera), simplemente escribiremos líneas de salida dando información de los datos elegidos por el usuario, los de la maquina y el resultado final y ya después nos regresara al case 0 que sería el menú para empezar de nuevo.

Texto

Descripción generada automáticamente

Como podemos ver hay 3 if los cuales son los 3 posibles resultados que se darían si el usuario eligió piedra, y así es lo mismo con los 3 casos del switch, en esto solo cambiaria la distribución de los datos de salida que se muestran al usuario

Texto

Descripción generada automáticamente con confianza media

En cuanto al diseño del Arduino no se implemento nada mas que el simulino uno y la terminal virtual.

Gráfico

Descripción generada automáticamente con confianza media

**Resultados:**

Como podemos ver este es la forma de operación del programa pidiéndole al usuario que elija entre los 3 posibles acciones y dependiendo de ello y la elección random de la maquina se decidirá quien es el ganador.

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Texto

Descripción generada automáticamente**

**Conclusiones:**

El desarrollar este programa me hizo ver cómo usar el método switch es una buena manera de navegar en el método loop a voluntad y así en cierto modo evitar que el programa Arduino haga un repetir hasta que al menos el programa acabe ya que cierto modo al principio se dificulto el desarrollo del programa por cuestiones del método loop que tiende a repetir las acciones y no se puede parar, lo máximo que se puede hacer es darle tiempo al usuario de elegir , también pude ver como utilizar los random en Arduino.

### Práctica 2. Adivinar numero de la computadora

**Descripción**: El programa hace un pequeño juego para el usuario el cual tiene que adivinar un numero del 1 al 5

**Introducción**:

Para el desarrollo de este programa se utilizaron como instrucciones especiales:

* Switch
* IF
* Random

Primero se utilizó la instrucción Switch para crear los casos y poder navegar entre el menú principal y el case 1 donde resolvemos el problema el porqué de hacer esto es para controlar de mejor manera el método loop del Arduino.

El IF se utilizó para hacer comparación entre la respuesta del usuario y la de la maquina y a si el programa podría verificar si el numero que ingreso el usuario es el mismo que selecciono el programa y el método random se declara para poder asignar un numero al azar de los posibles. Con esto terminado la asignación de si el usuario adivino o no el numero de la maquina entonces se devuelve al caso 0 y comienza de nuevo.

**Desarrollo:**

Para el desarrollo del programa primero se inicializaron las variables que se utilizaran y el tipo de dato del que serán.

**Texto

Descripción generada automáticamente**

Después lo que hacemos es crear un switch en el método Loop con el cual crearemos dos casos y podremos regresar de nuevo al menú una vez nos dé el resultado.

Imagen que contiene Texto

Descripción generada automáticamente

En el case 0 lo que hacemos es poner varios instrucciones de salida para que se visualicen en la terminal y el usuario pueda ver las instrucciones del programa, después con un ReadString leeremos los datos de entrada del usuario el cual será el numero que el cree que el programa tiene. Con el delay le damos un poco de tiempo al usuario de pensar y después agregamos un 1 a la variable estado para cambiar al Case 1.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Ya en el case 1: lo que hacemos es inicializar el método random y asignarle un valor del 1 al 5 a la variable (a), después lo pasamos a otra variable que la usaremos para que el programa haga la siguiente comparación, si ambos números son iguales entonces se mostrara un mensaje de que el usuario a adivinado el número, si no es así pues mandara un mensaje diciendo que no adivino, en cualquiera de los dos casos se agrega el valor 0 a la variable estado para al romper el ciclo del caso 1 este vuelva al menú.

Tabla

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente con confianza mediaEn cuanto al diseño del Arduino no se implementó nada más que el simulino uno y la terminal virtual.

**Resultados:** Como resultados podemos ver los mensajes que arroja cuando el usuario da con el numero o falla

Texto

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

**Conclusión:**

En conclusión podemos ver que utilizar el método random es perfecto para resolver algunos problemas, al igual que el método switch el cual es muy útil para crear menús y crear una interfaz con la que el usuario y el programa pueda moverse conforme se le indique y no sigua un cierto patrón.