



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**FUNDAMENTOS DE LA
PROGRAMACIÓN**

Semestre 2022-1

**Trabajo Final: Programa
"Piedra, papel o tijera"**

ESTUDIANTE

Josue Santiago Avilés Mora

PROFESORA

Marco Antonio Martínez Quintana



07 de diciembre, Cd. Mx., 2021

Programación de Piedra, papel o tijera

Resumen del contenido del documento

Este proyecto final contiene una introducción referente al tema que aborda el proyecto final que es un programa inspirado en el juego “Piedra, papel o tijera” conocido mundialmente y que es popular entre los niños, se incluye parte de su contexto histórico y cómo ha evolucionado hasta llegar a ser lo que es hoy en día.

El desarrollo del proyecto lleva una descripción general del proyecto principalmente enfocados principalmente en el programa, el algoritmo completo de la solución, el diagrama de flujo completo, el pseudocódigo completo y el código fuente, este último con la explicación de que hace cada parte del código. La sección de resultados incluye igualmente varios apartados como son la capturas de pantalla del funcionamiento de su proyecto, la tabla de recursos informáticos [software y hardware], una tabla de costos referentes al proyecto y por último un Diagrama de Gantt.

También se realizó un video donde se presenta el proyecto describiendo nuevamente lo que conllevó su realización y se hace una demostración del mismo de forma que se vea su funcionamiento y orden, se adjuntó el link del video en las referencias y en el trabajo está el canal de youtube donde se encuentra el video, igualmente se adjuntó el link del Repositorio de GitHub que contiene el Proyecto Final. Como último anexo de los resultados del proyecto, se incluye un Manual de usuario del proyecto para saber cómo utilizar el programa en una forma muy sencilla.

La última parte del proyecto son las conclusiones, aquí describimos cómo se aplicaron los temas de la clase en su proyecto final y por qué no se aplicaron los temas faltantes y las conclusiones personales sobre el proyecto y cómo cambió la perspectiva del mismo al realizarlo mencionando lo que me gustó y no. Por última parte del documento solo se incluyen las referencias de donde se basó para hacer el trabajo.

Introducción

Se elaborará el conocido juego de piedra, papel o tijeras en lenguaje de programación C, a partir de esta idea amplia empezamos mapeando el algoritmo y diagrama de flujo a su respectivo pseudocódigo. De forma que se vea todo el proceso y se pueda ejecutar de

forma eficiente. Pero primero, hay que conocer todo lo que involucra el juego desde sus inicios.

Saber cuándo se inventó el juego definitivamente es casi imposible, pero si se conoce que la primera mención de una versión parecida al juego de Piedra, papel o tijeras se encuentra en un libro escrito por un escritor chino. El autor se refiere a él como shoushiling y data sus orígenes en algún momento entre el 206 aC - 220 dC. También un reconocido artista que vivió durante la misma época, también habla de esta antigua versión del juego en un libro que escribió.

Después de esto se encontraron referencias al juego en Japón, este país suele ser igualmente nombrado como el lugar de invención del juego pero en 1995, Sepp Linhart confirma el origen en China y su consiguiente popularidad en Japón renombrándolo como sansukumi-ken , aún así, este suceso y popularidad igualmente nos acercó más a la forma de jugarlo hoy en día trayendo consigo ligeros cambios en la forma y reglas que se solían utilizar.

Hoy, tanto dentro como fuera de Japón, la versión más popular de sansukumi-ken incluye los clásicos gestos de piedra, papel y tijera. En Japón, el juego se llama jan-ken o jan-ken-pon. Debido a que Japón y muchos países de Occidente comenzaron a tener un contacto más habitual, el jan-ken finalmente aterrizó en el otro lado del mundo. A principios del siglo XX, los estadounidenses se habían familiarizado con las reglas de piedra, papel o tijera, un hecho que no pasó desapercibido tampoco para los medios de comunicación europeos y se dispersó a todo el mundo.

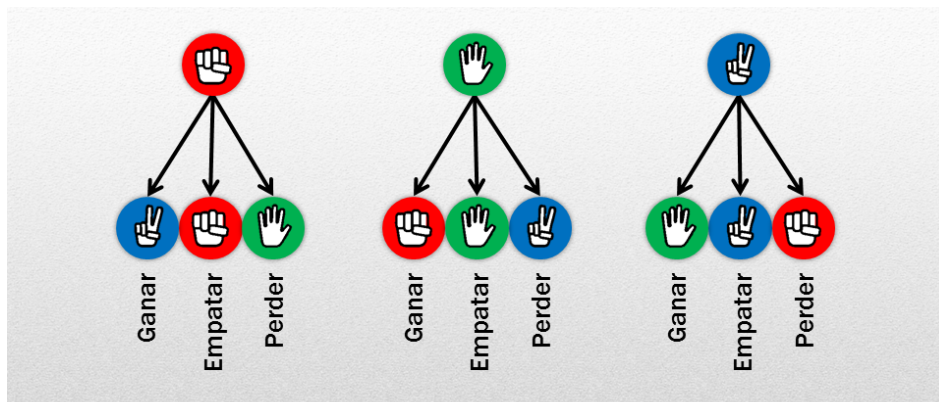
Aunque es aceptado como un juego de suerte y aleatoriedad, lo cierto es que Piedra, papel o tijera ha sido objeto de muchos estudios de juegos mentales, con expertos que reconocen el uso de al menos alguna estrategia por parte de jugadores profesionales, lo que ha llevado a incluso haber competiciones mundiales e ligas nacionales.

En el centro de las estrategias más comúnmente empleadas en el juego se encuentra una sola noción: la única forma de maximizar tus posibilidades de ganar una partida de Piedra, papel o tijera implica una selección aleatoria completa de movimientos.

Por lo tanto, al menos para intentar crear estrategias, los jugadores profesionales han ideado formas de maximizar la aleatoriedad de sus movimientos durante una partida. Para hacerlo, muchos emplean el uso de gambitos, que son series de movimientos preseleccionados elegidos con una intención estratégica específica.

Al predeterminedar sus tácticas, los jugadores más expertos aplacan las tendencias humanas naturales que podrían revelar su plan de juego como son los tics físicos que indican su elección entre una piedra o un movimiento de papel. Cuando los jugadores ganan una ronda con un movimiento, tienden a repetirlo más a menudo que lo que se esperaría al azar. En cambio, cuando pierden, cambian de jugada siempre y generalmente en el orden del nombre del juego. Irónicamente, al eliminar la aleatoriedad, los jugadores pueden ser verdaderamente aleatorios.

Es fundamental siempre tener en cuenta los factores más importantes que caracterizan el juego y que por lo tanto son indispensables para el programa como es el caso de la respuesta aleatoria entre los tres factores que son la piedra, el papel y la tijeras, también saber qué resultado se obtiene de cada combinación posible ya sea el resultado empate, ganar o perder y que el programa lo imprima.



Desarrollo del proyecto

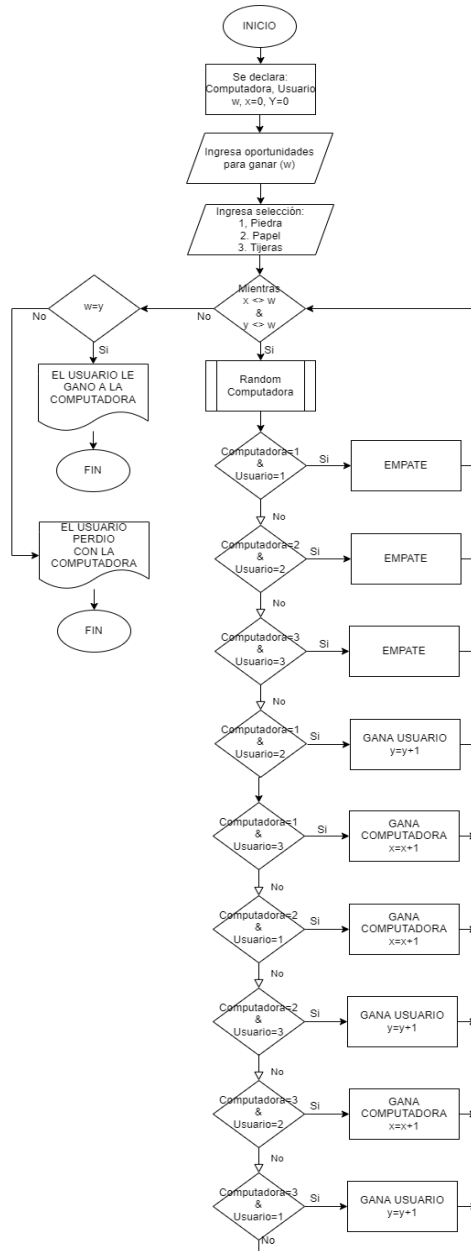
–Descripción general del proyecto (idea con enfoque de emprendimiento)

Diseñar un juego intuitivo de fácil uso que permita a gente de toda edad tener acceso a un pasatiempo a través del juego que ya conocen que es “piedra, papel o tijeras”. El programa podría obtener ganancias a partir de, recordando que es para todo público y ser popular generar interés de una empresa y venderse, también por licencias y el factor de vender los derechos de usar el juego para una marca, al igual que los juegos gratuitos suelen ganar dinero por publicidad.

-Algoritmo completo de la solución

1. Ponemos el título para saber de qué se trata (piedra, papel o tijeras) y una introducción de que hay que introducir.
2. De acuerdo al valor que se introdujo, se habla de la duración del juego porque se introdujo un valor numérico de a las cuantas veces ganadas se termina el programa.
3. Teniendo esto ahora introducimos 1, 2 o 3 de acuerdo a la jugada que quieras hacer (1- piedra, 2- papel, 3- tijeras).
4. De acuerdo a lo que elegiste y el programa sacó aleatoriamente se define si ganaste, perdiste o empataste.
5. Una vez que la persona o la computadora llega al número de juegos establecidos al principio acaba el programa y dice quién ganó.

-Diagrama de flujo



-Pseudocódigo completo

Piedra, papel o tijeras

INICIO

ESCRIBIR "---Piedra, papel o tijeras---"

ESCRIBIR "El primero en llegar a n ganadas, gana el juego."

ESCRIBIR "EJEMPLO: Ingresas 3, entonces el primero en llegar a 3 gana."

ESCRIBIR "Ingresa tus oportunidades para ganar:"

SOLICITAR valor numérico

Si es 1) Solo se siguen los siguientes pasos una vez

Si es n) Se repite el juego el número de veces necesarias hasta llegar a la cantidad deseada.

ESCRIBIR "Ingresa:"

1. Roca

2. papel

3. Tijeras"

Escribir "Tienes n oportunidades para ganar"

Escribir "Marcador: computadora=n tu=n"

Escribir "Ingresa tu opción"

SELECCIONAR (1,2 o 3) EN

CASO 1 ->

SI roca y el algoritmo aleatorio elige roca

ENTONCES

ESCRIBIR "Roca vs Roca. Empate".

SI papel y el algoritmo aleatorio elige tijeras

ENTONCES

ESCRIBIR "Roca vs Tijeras. Tú Ganas"

SI piedra y el algoritmo aleatorio elige papel

ENTONCES

ESCRIBIR "Roca vs Papel. Computadora Gana"

CASO 2 ->

SI papel y el algoritmo aleatorio elige roca

ENTONCES

ESCRIBIR "Papel vs Roca. Tú Ganas"

SI papel y el algoritmo aleatorio elige tijeras

ENTONCES

ESCRIBIR "Papel vs Tijeras. Computadora gana"

SI papel y el algoritmo aleatorio elige papel

ENTONCES

ESCRIBIR "Papel vs Papel. Empate"

CASO 3->

SI tijeras y el algoritmo aleatorio elige roca

ENTONCES

ESCRIBIR "Tijeras vs Roca. Computadora gana"

SI tijeras y el algoritmo aleatorio elige tijeras

ENTONCES

ESCRIBIR "Tijeras vs Tijeras. Empate"
Si tijeras y el algoritmo aleatorio elige papel
ENTONCES
ESCRIBIR "Tijeras vs Papel. Tu ganas"
DEFECTO ->
Se para el juego y comienza otro sin afectar el marcador
Si ganó el jugador:
ESCRIBIR "##Ganaste el juego##"
Si perdió:
Escribir "##Perdiste el juego##"

FIN SELECCIONAR

-Código fuente COMENTADO del proyecto completo.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  void main()
6  {
7      int computadora, usuario, x=0, y=0, w;
8      //Proyecto final
9      printf(" \n\t\t--- Piedra, Papel o Tijeras ---\n\n");
10     printf("El primero en llegar a n ganadas, gana el juego.");
11     printf("\nEJEMPLO: Ingresas 3, entonces el primero en llegar a 3 gana.");
12     printf("\n\nIngresas tus oportunidades para ganar:");
13     scanf("%d",&w);
14
15     printf("\nIngresas: \n1. Roca, \n2. Papel, \n3. Tijeras\n");
16     printf("\nTienes %d Oportunidades para ganar\n",w);
17
18     while(x!=w && y!=w)
19     {
20         srand(time(0)); //Esto hace que la computadora de una opción aleatoria
21         computadora = (rand()%3)+1;
22
23         printf("\nMarcador: \nComputadora = %d \tTu = %d",x,y);
24         printf("\n\nIngresas tu opción :");
25         scanf("%d",&usuario);
26
27         if(computadora==1 && usuario==1)
28             printf("Roca vs Roca\n>>EMPATE");
29         else if(computadora==2 && usuario==2)
30             printf("Papel vs Papel\n>>EMPATE");
31         else if(computadora==3 && usuario==3)
32             printf("Tijeras vs Tijeras\n>>EMPATE");
33         else if(computadora==1 && usuario==2)
34             printf("Piedra vs Papel\n>>TU GANAS");
35         y++;
36     }
```

```

36         else if(computadora==1 && usuario==3)
37         {printf("Piedra vs Tijeras\n>>COMPUTADORA GANA");
38           x++;}
39         else if(computadora==2 && usuario==1)
40         {printf("Papel vs Roca\n>>COMPUTADORA GANA");
41           x++;}
42         else if(computadora==2 && usuario==3)
43         {printf("Papel vs Tijeras\n>>TU GANAS");
44           y++;}
45         else if(computadora==3 && usuario==2)
46         {printf("Tijeras vs Papel\n>>COMPUTADORA GANA");
47           x++;}
48         else if(computadora==3 && usuario==1)
49         {printf("Tijeras vs Piedra\n>>TU GANAS");
50           y++;}
51     }
52
53     if(y==w)
54     printf("\n\n\t\t# # GANASTE EL JUEGO # #\n");
55     else
56     printf("\n\n\t\t# # PERDISTE EL JUEGO # #\n");
57     printf("\n--GRACIAS POR JUGAR--\n");
58
59 }

```

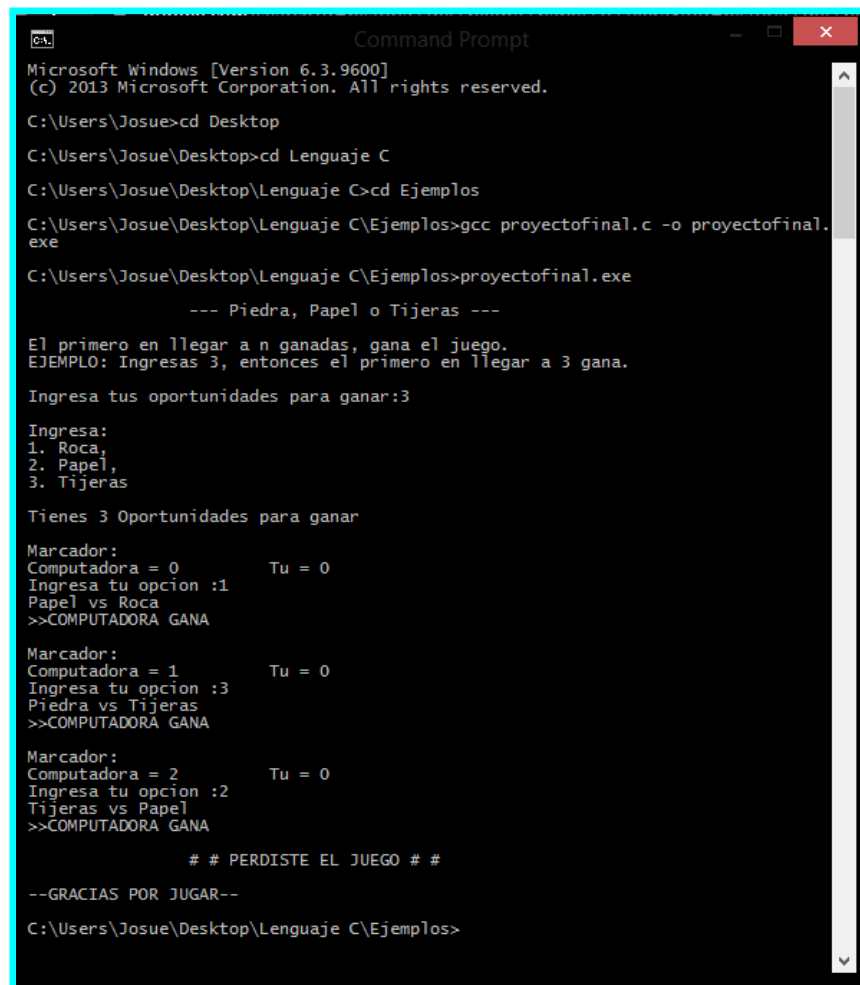
Compilador Recursos Registro de Compilación Depuración Resultados
 Line: 27 Col: 41 Sel: 0 Lines: 59 Length: 1784 Insertar Done parsing

- Para el código fuente se utilizaron primero tres librerías: Una para entrada de datos, uno para el random y la otra que va por default.
- Se utiliza en este caso void main () porque indica que el método main no retorna ningún valor.
- Se utiliza int para declarar una variable de tipo entero.
- Los printf dicen 1. el título del programa, 2. dice la forma en que se tiene que introducir la primera variable del programa, 3. Da un ejemplo para tener en claro cómo funciona esta parte del programa y la 4. Da la instrucción de insertar la variable numérica.
- Se utiliza \n para saltar de línea y \t para alinear a la derecha de forma que el programa se vea más estético.
- Se utiliza scanf() para que el usuario introduzca el valor que eligió.
- Se utiliza nuevamente printf para mostrarnos las opciones que se pueden utilizar para el juego (1-piedra, 2-papel o 3-tijeras) con la finalidad de hacer más ágil el juego y el otro printf nos da el número que el usuario introdujo (%d) para decirnos las oportunidades que necesitamos para ganar.
- Utilizamos while () para decir que mientras el valor de x sea distinto del valor de w y de y, seguirá el ciclo de juego. Mientras se cumplan los dos, si w=y significa que ganaste y si w=x perdiste.
- (srand(time(0))) srand se utiliza para que el valor sea diferente, y la secuencia resultante sea diferente cada vez que se ejecute el programa. Normalmente para que el valor sea diferente se usa la fecha, hora del sistema, usando la función time().
- =(rand()%3+1 hace que la computadora haga la elección entre las tres opciones posibles de forma aleatoria.
- Utilizamos printf() en esta ocasión para generar un marcador que da las veces

- que la computadora ha ganado contra las veces que el usuario gana.
- Usamos nuevamente printf() para indicar la introducción de uno de los tres posibles valores y scanf() para que el usuario pueda ingresar el valor.
 - Hacemos una serie de if y else if donde introducimos las posibles combinaciones que puede dar el juego y sus respectivas respuestas si es empate, ganaste o ganó la computadora.
 - Por último como mencionamos, nos da el último if que nos dice, si w=y ganaste el juego y el else sería w=x por lo que perdiste el juego.
 - Se utiliza el printf() para dar un mensaje de despedida y se cierra el programa.

Resultados del proyecto

-Capturas de pantalla del funcionamiento de su proyecto



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Josue>cd Desktop
C:\Users\Josue\Desktop>cd Lenguaje C
C:\Users\Josue\Desktop\Lenguaje C>cd Ejemplos
C:\Users\Josue\Desktop\Lenguaje C\Ejemplos>gcc proyectofinal.c -o proyectofinal.exe
C:\Users\Josue\Desktop\Lenguaje C\Ejemplos>proyectofinal.exe

--- Piedra, Papel o Tijeras ---

El primero en llegar a n ganadas, gana el juego.
EJEMPLO: Ingresas 3, entonces el primero en llegar a 3 gana.

Ingresa tus oportunidades para ganar:3

Ingresa:
1. Roca
2. Papel
3. Tijeras

Tienes 3 Oportunidades para ganar

Marcador:
Computadora = 0      Tu = 0
Ingresa tu opcion :1
Papel vs Roca
>>COMPUTADORA GANA

Marcador:
Computadora = 1      Tu = 0
Ingresa tu opcion :3
Piedra vs Tijeras
>>COMPUTADORA GANA

Marcador:
Computadora = 2      Tu = 0
Ingresa tu opcion :2
Tijeras vs Papel
>>COMPUTADORA GANA

## PERDISTE EL JUEGO ##

--GRACIAS POR JUGAR--
C:\Users\Josue\Desktop\Lenguaje C\Ejemplos>
```

- Primero utilizamos el comando cd que nos permite cambiar de directorio para entrar a donde está guardado el proyecto.
- Una vez estando en la carpeta que se encuentra el programa utilizamos gcc (GNU Compiler Collection), es un conjunto de compiladores de uso libre para sistemas operativos basados en UNIX. Entre sus compiladores existe el que sirve para programas escritos en C. Esto para hacer ejecutable el programa pasándolo de .c a .exe.
- Al ejecutar el programa actúa de la forma esperada, cumpliendo con el propósito buscado donde el principal problema era encontrar la forma de lograr que el programa nos diera una de las 3 opciones de manera aleatoria como se mencionó anteriormente.
- En este ejemplo del funcionamiento del programa se introdujo la opción de el que llegara primero a ganar 3 veces ganaba el juego. Por eso al haber perdido el usuario 3 veces seguidas el programa no generó otra vez la opción para ingresar otra opción y terminó el programa con el resultado final que en este caso fue perder y el mensaje final.

```

C:\Users\Josue\Desktop\Lenguaje C\Ejemplos>proyectofinal.exe

      --- Piedra, Papel o Tijeras ---

El primero en llegar a n ganadas, gana el juego.
EJEMPLO: Ingresas 3, entonces el primero en llegar a 3 gana.

Ingresa tus oportunidades para ganar:2

Ingresa:
1. Roca,
2. Papel,
3. Tijeras

Tienes 2 Oportunidades para ganar

Marcador:
Computadora = 0      Tu = 0
Ingresa tu opcion :3
Papel vs Tijeras
>>TU GANAS

Marcador:
Computadora = 0      Tu = 1
Ingresa tu opcion :2
Piedra vs Papel
>>TU GANAS

      # # GANASTE EL JUEGO # #

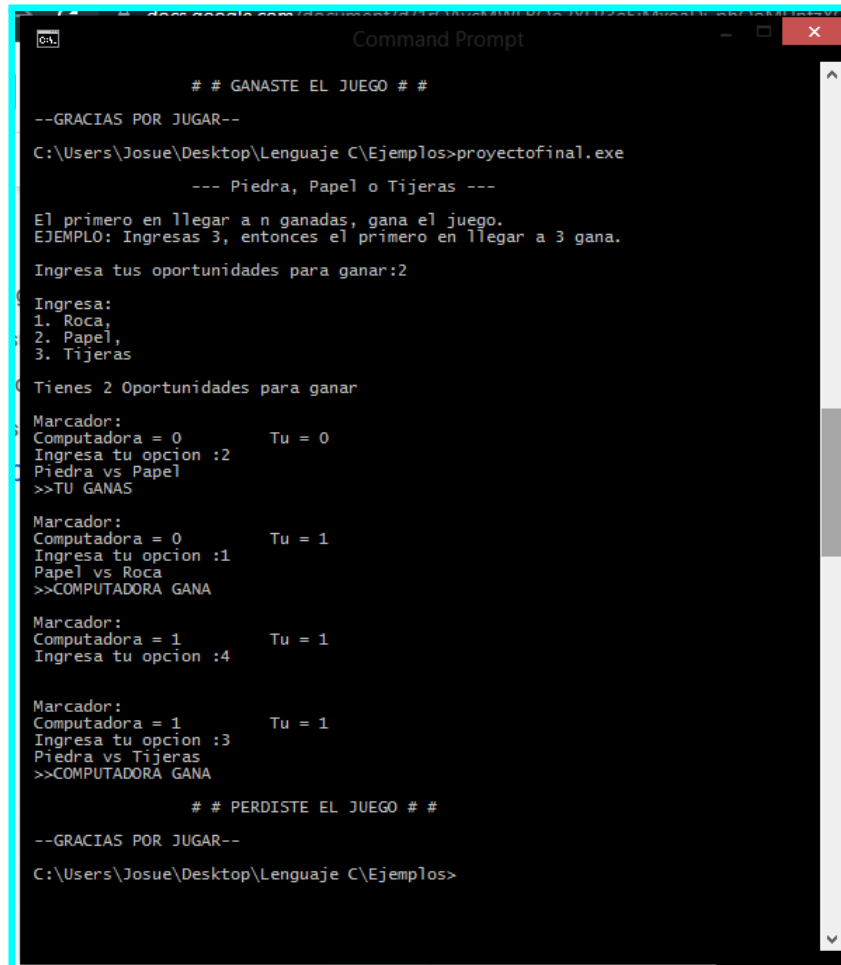
--GRACIAS POR JUGAR--

C:\Users\Josue\Desktop\Lenguaje C\Ejemplos>proyectofinal.exe

```

- En este ejemplo del funcionamiento del programa se introdujo la opción de el que llegara primero a ganar 2 veces ganaba el juego. Por eso al haber ganado el usuario 2 veces seguidas, el programa no generó otra vez la opción para ingresar

otra opción y terminó el programa dando el resultado final que en este caso fue ganar y el mensaje final.



```
## GANASTE EL JUEGO ##
--GRACIAS POR JUGAR--
C:\Users\Josue\Desktop\Lenguaje C\Ejemplos>proyectofinal.exe
--- Piedra, Papel o Tijeras ---
El primero en llegar a n ganadas, gana el juego.
EJEMPLO: Ingresas 3, entonces el primero en llegar a 3 gana.
Ingresas tus oportunidades para ganar:2
Ingresas:
1. Roca,
2. Papel,
3. Tijeras
Tienes 2 Oportunidades para ganar
Marcador:
Computadora = 0      Tu = 0
Ingresas tu opcion :2
Piedra vs Papel
>>TU GANAS
Marcador:
Computadora = 0      Tu = 1
Ingresas tu opcion :1
Papel vs Roca
>>COMPUTADORA GANA
Marcador:
Computadora = 1      Tu = 1
Ingresas tu opcion :4
Piedra vs Tijeras
>>COMPUTADORA GANA
# # PERDISTE EL JUEGO # #
--GRACIAS POR JUGAR--
C:\Users\Josue\Desktop\Lenguaje C\Ejemplos>
```

- En este ejemplo del funcionamiento del programa se introdujo la opción de el que llegara primero a ganar 2 veces, ganaba el juego. ¿Por qué entonces hay más juegos dentro del programa que cuando se eligió 3? Esto pasa porque primeramente debido a que el usuario ganó un juego y la computadora dos después, esto da 3, el 4to juego muestra que pasa si se introduce un dato invalido (fuera de 1,2,3), lo que hace el programa es automáticamente pasar a un nuevo juego sin alterar el marcador. Finalmente el programa da el resultado final acerca de quién ganó más juegos que en este caso fue la computadora nuevamente y el mensaje final.

-Tabla de recursos informáticos

Para llevar a cabo el proyecto fue necesario el uso de software y hardware.

Software
Sistema operativo Windows versión 10
Office 360
Google docs
Dev-C++ Version: 2, June 1991
Command Prompt
Hardware
Laptop HP modelo: ENVY

-Tabla de costos asociados al proyecto

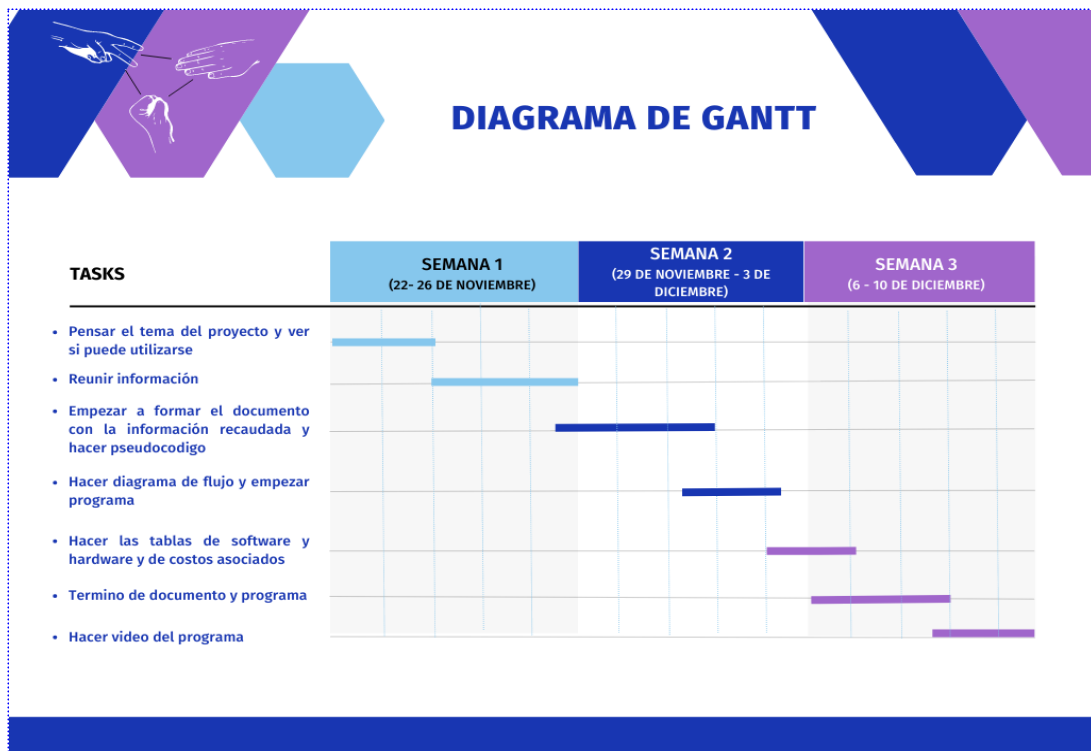
Esta es una tabla que relaciona los costos implicados en la realización del proyecto tomando en cuenta las fechas del 22 de noviembre del 2021 al 10 de diciembre del 2021.

item	Descripción	cant (hrs)	Precio Unitario	Precio Total
1	Project Manager	120	\$ 200.00	\$ 24,000.00
2	Programador	120	\$ 80.00	\$ 9,600.00

Subtotal	\$ 33,600.00
Iva	\$ 5,376.00
Total	\$ 38,976.00

-Diagrama de Gantt

Para la elaboración del proyecto fue necesaria la creación de un diagrama de Gantt que inicia del 22 de noviembre del 2021 al 10 de diciembre del 2021 que son las fechas de inicio y final del proyecto.



-Canal de YouTube donde se encuentra el video: <https://youtu.be/X8oQ4U6tsG8> Canal Josue Aviles

-Repositorio de GitHub del Proyecto Final:
<https://github.com/JosueSAvilesM/Proyecto-final.git>

-Manual de usuario del proyecto

1. Introduce un valor numérico que va a representar la cantidad de veces que alguno de los jugadores (computadora o tú) para ganar, no hay empate al final por lo que si siguen empatando se sigue jugando hasta que uno pierda.
2. Introduce 1, 2 o 3 respectivamente de acuerdo a la variable que quieras jugar. 1-piedra, 2-papel, 3-tijeras.
3. Cada juego puedes introducir una opción diferente.
4. Una vez que uno alcance la cantidad establecida de partidas ganadas, gana el juego sin importar si en el intermedio el otro jugador ganó alguna de las partidas.

Conclusión

-Describir cómo aplicaron los temas de la clase en su proyecto final y por qué no aplicaron los temas faltantes.

Los temas de la clase estuvieron siempre presentes en el proyecto ya que muchos de ellos son indispensables para la realización del código por ejemplo `printf()` y `scanf()`, igual se usaron temas por ser requeridos por las especificaciones como el algoritmo completo de la solución, diagrama de flujo, pseudocódigo, etc pero igualmente no se sienten extra porque ayudan a llegar al objetivo principal del proyecto que es la realización del código fuente y ejecutarlo, que incluso en eso se utilizan temas de la clase como fue el utilizar `cd` y `gcc` para poder pasar el programa de `.c` a `.exe`.

Los temas de clase que no se utilizaron fueron pocos y si no se utilizaron fue porque no eran necesarios o no podían utilizarse para el trabajo.

-Conclusiones personales

Aunque fue un proyecto final laborioso, me gustó mucho realizarlo aunque en un principio creía que iba a ser aburrido en realidad es interesante el desarrollo y como cada paso te va acercando a la solución y dando nuevas ideas respecto al programa, al final te sientes satisfecho de haberlo hecho logrando el objetivo que planteaste desde un inicio, lo único que no me gusta mucho hacer son videos pero igual no se sintió pesado hacerlo aunque durara 15 min aproximadamente.

Referencias

- Digital, R. (2020, 21 mayo). *Juegos para el aislamiento: Piedra, papel o tijera*. COPE. Recuperado 8 de diciembre de 2021, de https://www.cope.es/actualidad/vivir/juegos-tradicionales/noticias/juegos-para-aislamiento-piedra-papel-tijera-20200521_728128
- El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.
- Guía práctica de estudio 01-: La computación como herramienta de trabajo del profesional de ingeniería (2021).
- Guía práctica de estudio 02-: GNU / Linux (2021).
- Guía práctica de estudio 03-: Solución de problemas y Algoritmos (2021).
- Guía práctica de estudio 04-: Diagramas de flujo (2021).
- Guía práctica de estudio 05-: Pseudocódigo (2021).
- Guía práctica de estudio 06-: Entorno y Fundamentos de Lenguaje C (2021).

- Guía práctica de estudio 07-: Estructuras de selección (2021).
- Guía práctica de estudio 08-: Estructuras de repetición (2021).
- Guía práctica de estudio 09-: Arreglos unidimensionales (2021).
- Guía práctica de estudio 10-: Arreglos multidimensionales (2021).
- Guía práctica de estudio 11-: Funciones (2021).
- Guía práctica de estudio 12-: Lectura y escritura de datos (2021).