

Introducción

El problema planteado para el concurso interpreparatoriano 2017-2018 fue el diseño de un juego de “21 Blackjack” en el paradigma de **programación estructurada** en **lenguaje C**.

El **blackjack** es un juego de azar con cartas cuyo objetivo es formar una mano con una **suma** total lo mas proxima a **21** posible sin pasarse, se juega con una (o más) baraja inglesa sin comodines de 52 cartas, en el juego cada carta tiene un valor, **las cartas numéricas** tienen el valor que su carta representa, **las cartas de figuras** (J, Q, K) valen 10 y el **As** tiene un valor de 11 o 1 dependiendo de lo que más convenga.

C es un **lenguaje de programacion** de medio nivel, es decir que tiene características de los lenguajes de alto y bajo nivel, por ello es posible escribir código entendible para las personas y a la vez acceder directamente al **hardware** del ordenador como la **memoria**, **periféricos**, **pantalla**, etc.

Definición del problema

“21 Blackjack” es un juego de cartas cuyo objetivo es obtener una suma de cartas lo mas cercano a 21 posible sin pasarse. En el juego se compite contra la banca, la cual es la encargada de repartir las cartas y pagar las apuestas pertinentes.

Análisis del problema

El juego comienza con una **apuesta inicial** de **cada jugador**, después la banca saca una carta después se le entregan 2 cartas a cada jugador y se les da la opción de **duplicar** su apuesta.

En caso de que la carta de la banca sea un “**As**” el jugador podrá realizar la “**apuesta al seguro**” es decir el jugador apuesta a que la banca obtendrá **blackjack** cuando saque su segunda carta, esta apuesta tiene un valor del 50% de la **apuesta inicial** y en caso de ganar se **paga** con relación **2:1**.

Después de recibir las 2 cartas iniciales, el jugador podrá **dividir su juego** en caso de que tenga cartas con el **mismo número**, en tal caso deberá hacer una apuesta igual a la inicial para su segundo juego. El jugador tendrá la opción de **plantarse** con sus cartas recibidas o **pedir** más cartas (para cada juego en caso de que dividiera su juego); si el jugador al pedir más cartas estas suman más de 21, el jugador **pierde** automáticamente.

Una vez que cada jugador haya jugado su mano, la banca sacara cartas para sí misma, **la banca pide** si sus cartas suman **menos de 17** y se planta si suman **17 o más**. Una vez que la banca se haya plantado, ésta empezará a **pagar** las apuestas a los jugadores ganadores.

El jugador gana si:

- La banca **se pasa** de 21, cobrara con relación 1:1 de su apuesta.

- El jugador quedó **más cerca** de 21 (sin pasarse) que la banca, para pagar se tendrá en cuenta lo siguiente:
 - El jugador formó **blackjack**, cobrará con relación **3:2** de su apuesta.
 - El jugador no formó **blackjack**, cobrará con relación **1:1** de su apuesta.

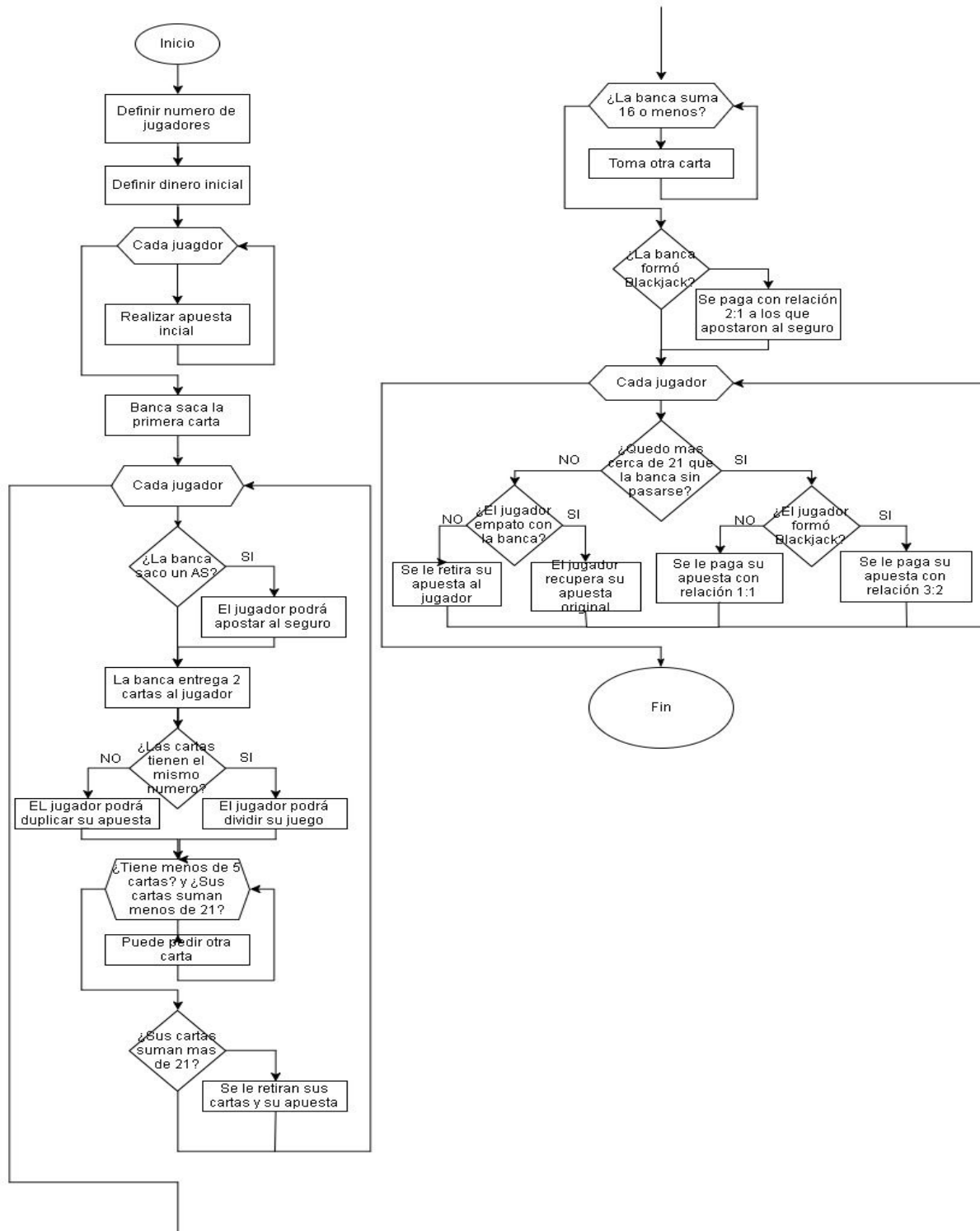
El jugador **empata** si sus cartas suman **lo mismo** que las de la banca, en tal caso solo recuperará su apuesta.

El jugador **pierde** si:


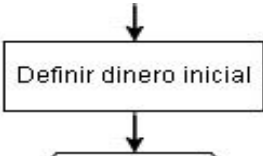
- La banca quedó **más cerca** de 21 (sin pasarse) que el jugador.
- El jugador **se pasó** de 21.

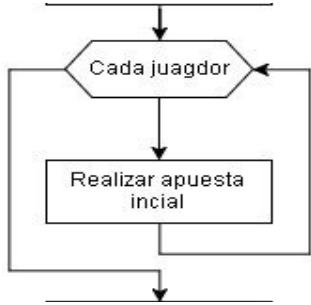
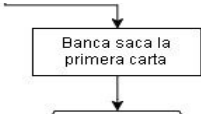
Si el jugador pierde se le retira su apuesta y su mano.

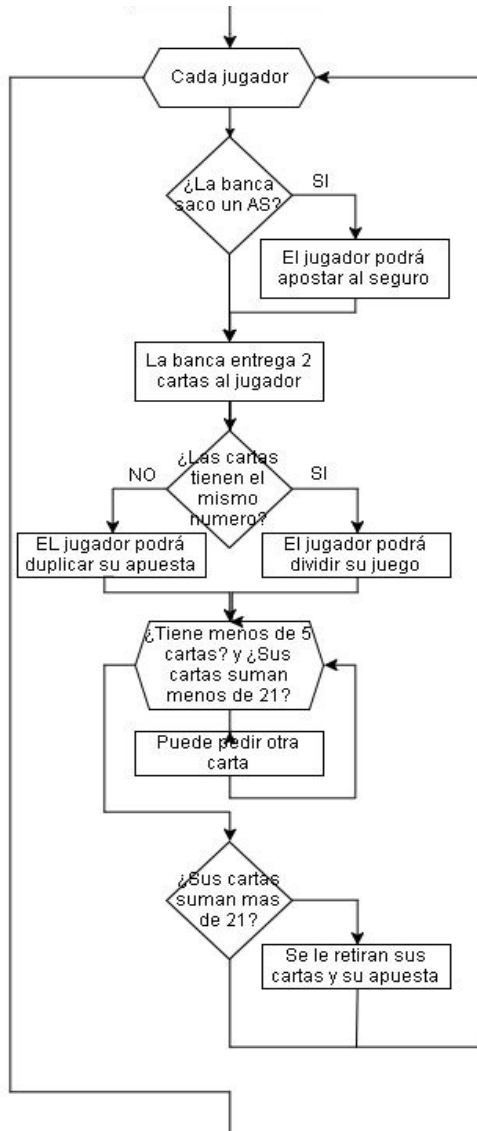
Diseño de la solución del problema



Implementación de la solución del problema

Fragmento de diagrama de flujo	Implementación (pseudocódigo)
 <pre> graph TD Inicio([Inicio]) --> DefinirNumero[Definir numero de jugadores] DefinirNumero --> Next[] style Next fill:none,stroke:none </pre>	<pre> //Menu principal Inicio MainMenu //Opciones para elegir Escribir "Un jugador" Escribir "Multijugador" Escribir "Salir" Leer seleccion Caso seleccion = 0: Ir a SingleMenu Caso seleccion = 1: Ir a MultiplayerMenu Caso seleccion = 2: Salir Fin MainMenu </pre>
 <pre> graph TD Next[] --> DefinirDinero[Definir dinero inicial] DefinirDinero --> Next2[] style Next fill:none,stroke:none style Next2 fill:none,stroke:none </pre>	<pre> //Menu para juego individual Inicio SingleMenu numeroDeJugadores <- 1 //Opciones para elegir Escribir "Nombre" Escribir "Dinero inicial (\$10 - \$15000)" Escribir "Presiona CTRL + C para regresar" Leer nombre Leer dineroInicial Si dineroInicial menor o igual que 1 entonces: Ir a MenuPrincipal Sino Regresar Fin SingleMenu Inicio MultiplayerMenu Escribir "Numero de jugadores" Escribir "Dinero inicial (\$10 - \$15000)" </pre>

	<p> Leer numeroDeJugadores Leer dineroInicial Si dineroInicial menor o igual que 1 entonces: Ir a MenuPrincipal Sino Regresar Fin MultiplayerMenu </p>
 <pre> graph TD Entry(()) --> LoopStart(()) LoopStart --> Process[Realizar apuesta inicial] Process --> LoopStart LoopStart --> Exit(()) </pre>	<p> Inicio ApuestaInicial Para i <- 0 hasta numeroDeJugadores Escribir nombreDeJugador[i] Escribir dineroDeJugador[i] Escribir "Realiza tu apuesta inicial" Leer apuestaInicial dineroDeJugador[i] - apuestaInicial Fin Para Fin ApuestaInicial </p>
 <pre> graph TD Entry(()) --> Process[Banca saca la primera carta] Process --> Exit(()) </pre>	<p> Inicio RepartirCartas manoBanca[0] <- Baraja </p>



Para $i \leftarrow 0$ **hasta** numeroDeJugadores:
Si manoBanca[0] **es** As **Y**
dineroDeJugador[i]
es mayor o igual a apuestaInicial / 2
entonces:
Escribir “¿Quieres apostar al seguro?”
Leer respuesta
Si respuesta **es igual a verdadero**
entonces:
apuestaAlSeguro \leftarrow apuestaInicial / 2
dineroDeJugador[i] \leftarrow apuestaInicial / 2

Para $j \leftarrow 0$ **hasta** 2:
manoJugador \leftarrow baraja

Si manoJugador[0].numero **es igual a**
manoJugador[1].numero **entonces:**
Escribir “¿Quieres dividir tu juego?”
Leer respuesta;
Si respuesta **es igual a verdadero**
entonces:
juegoDividido \leftarrow verdadero
manoJugador[0] \leftarrow manoJugador[0]
manoJugador[1] \leftarrow manoJugador[1]

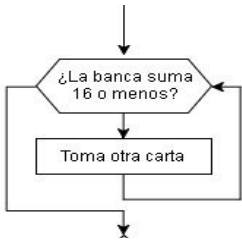
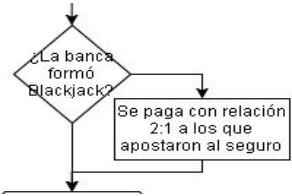
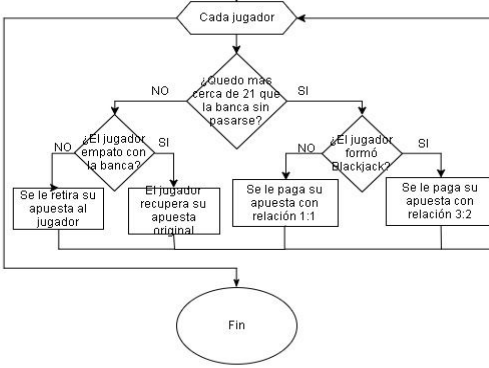
Escribir “¿Quieres duplicar tu apuesta?”
Leer respuesta
Si respuesta **es igual a verdadero**
entonces:
apuestaDelJugador \leftarrow apuestaInicial * 2

Mientras sumaDeCartas **menor o igual a**
21

Y numeroDeCartas **menor que** 5 **entonces:**
Escribir “¿Quieres otra carta?”
Leer respuesta
Si respuesta **es verdadero**
manoJugador \leftarrow baraja

Si sumaDeCartas **mayor que** 21 **entonces:**
apuestaDelJugador \leftarrow 0
Escribir “Perdiste”

Fin RepartirCartas

 <pre> graph TD Start(()) --> Decision{¿La banca suma 16 o menos?} Decision -- SI --> Process[Toma otra carta] Process --> Decision Decision -- NO --> Exit(()) </pre>	<p>Inicio PriceRound</p> <p>Para $i \leftarrow 1$ hasta 5:</p> <p>Si sumaDeCartasBanca menor que 17 entonces:</p> <p style="padding-left: 40px;">manoBanca \leftarrow baraja</p>
 <pre> graph TD Start(()) --> Decision{¿La banca formó Blackjack?} Decision -- SI --> Process[Se paga con relación 2:1 a los que apostaron al seguro] Process --> Exit(()) Decision -- NO --> Exit </pre>	<p>Para $i \leftarrow 0$ hasta numeroDeJugadores:</p> <p>[...]</p> <p>Si apuestaAlSeguro diferente de 0 entonces:</p> <p style="padding-left: 40px;">sumaDeCartasBanca es igual a 21 Y</p> <p style="padding-left: 40px;">numeroDeCartasBanca es igual a 2</p> <p>entonces:</p> <p style="padding-left: 40px;">Escribir “Apuesta de seguro ganada”</p> <p style="padding-left: 40px;">dineroDelJugador[i] +</p> <p style="padding-left: 40px;">apuestaAlSeguro * 3</p> <p>Sino</p> <p style="padding-left: 40px;">Escribir “Apuesta de seguro perdida”</p> <p style="padding-left: 40px;">apuestaAlSeguro \leftarrow 0</p>
 <pre> graph TD Loop[Cada jugador] --> Decision1{¿Quedo mas cerca de 21 que la banca sin pasarse?} Decision1 -- SI --> Decision2{¿El jugador formó Blackjack?} Decision2 -- SI --> Process1[Se le paga su apuesta con relación 3:2] Decision2 -- NO --> Process2[Se le paga su apuesta con relación 1:1] Decision1 -- NO --> Decision3{¿El jugador empató con la banca?} Decision3 -- SI --> Process3[El jugador recupera su apuesta original] Decision3 -- NO --> Process4[Se le retira su apuesta al jugador] Process1 --> Fin((Fin)) Process2 --> Fin Process3 --> Fin Process4 --> Fin </pre>	<p>Para $i \leftarrow 0$ hasta numeroDeJugadores</p> <p>Si sumaDeCartas es mayor que 21 entonces:</p> <p style="padding-left: 40px;">Escribir “Perdiste”</p> <p style="padding-left: 40px;">apuestaDelJugador \leftarrow 0</p> <p>O Bien Si sumaDeCartas es igual a sumaDeCartasBanca entonces:</p> <p style="padding-left: 40px;">Escribir “Empate”</p> <p style="padding-left: 40px;">dineroDelJugador + apuestaDelJugador</p> <p>O Bien Si numeroDeCartas es igual a 2 entonces:</p> <p style="padding-left: 40px;">Escribir “Blackjack”</p> <p style="padding-left: 40px;">dineroDelJugador + apuestaDelJugador * 2.5</p> <p>O Bien:</p> <p style="padding-left: 40px;">Escribir “Ganaste”</p> <p style="padding-left: 40px;">dineroDelJugador + apuestaDelJugador * 2</p> <p>[...]</p> <p>Fin PriceRound</p>

Glosario de términos

Lenguaje C .- Es uno de los lenguajes de programación mas populares y usados, usado para hacer aplicaciones de sistema operativo principalmente

lenguaje de programación de alto nivel .- Es un lenguaje que está más cerca del entendimiento del humano que de la computadora, esto permite un mejor entendimiento de los programas escritos pero implica una pérdida de aprovechamiento de la capacidad de procesamiento de la computadora.

lenguaje de programación de bajo nivel .- Es un lenguaje que depende totalmente de la máquina, es decir, que el programa realizado para una máquina específica no se puede usar en otra máquina diferente. Debido a esto usualmente estas aplicaciones aprovechan al máximo la capacidad de la máquina.

Lenguaje de programación de medio nivel .- Es un lenguaje que se encuentra entre los lenguajes de alto nivel y los de bajo, otorgando un buen entendimiento natural de los programas escritos sin perder la capacidad de manipulación del hardware.

Hardware .- Es el conjunto de componentes físicos que forman una computadora. Por ejemplo: monitor, teclado, raton, gabinete, etc.

Blackjack .- En el juego de 21 el blackjack es la mejor mano que se puede formar, consta de una suma de 21 con solo 2 cartas es decir un As y una carta con valor de 10.

Apuesta al seguro.- En el juego 21, la apuesta al seguro se puede hacer cuando la primera carta de la banca es un As y los jugadores al realizar la apuesta, apuestan a que la banca sumará 21 con su siguiente carta, es decir, formara blackjack.

Plantarse.- En el juego 21, plantarse se refiere a conformarse con las cartas obtenidas hasta el momento, es decir dejar de pedir cartas, es decir, concluir su turno.

Referencias

Harvey M. Deitel, Paul J. Deitel. (2004). Como programar en C/C++ y JAVA. México: Pearson Educación.

Anónimo. (2014). C Tutorial. 4/02/2018, de w3schools Sitio web: <https://www.w3schools.in/c-tutorial/>

Flowchart Maker & Online Diagram Software. (2018). Sitio web: <https://www.draw.io/>