

<b>UACM</b> Universidad Autónoma de la Ciudad de México <small>NADA HUMANO ME ES AJENO</small>	<b>Introducción a la ingeniería de software</b> <b>Planeación</b>	
<b>Producto</b>	<b>Proyecto de Brisca</b>	
<b>Emitido por</b>	<b>Flores Trejo Victor Ruben</b>	<b>Fecha: 10/09/23</b>

### 1.- Enunciado del proyecto:

Se juega con una Baraja Española de 40 cartas. Las cartas se dividen en cuatro “palos”: Oros, Copas, Espadas y Bastos. Las cartas en cualquiera de sus palos son: As, Dos, Tres, Cuatro, Cinco, Seis, Siete, Sota, Caballo y Rey; en total 10 cartas por cada palo.

Puede jugarse con dos o más jugadores, normalmente de dos a cuatro jugadores. El orden y valor de las cartas en cualquiera de sus palos es: As, 11 puntos; Tres, 10 puntos; Rey, 4 puntos; Caballo, 3 puntos y Sota, 2 puntos. Cómo se cuentan con 4 cartas del mismo valor, da como suma total para un juego 120 puntos. Las cartas no mencionadas no aportan valor al puntaje, pero suelen servir para determinar quién gana la mano.

Para la implementación de este juego se simulará un juego de brisca de un jugador (usuario) contra la banca (computadora). Debe implementarse para jugar en la terminal. El lenguaje de programación para la implementación es C.

Se escoge aleatoriamente una carta muestra. La muestra es cualquier carta de las 40 posibles.

Luego se reparte tres cartas aleatorias de la baraja a cada jugador.

Se decide por sorteo al jugador que comienza a jugar. El otro juega una carta de las tres que tienen también. No es necesario que la carta sea del mismo palo.

Cuando todos los jugadores han jugado una carta, la mejor mano la gana el jugador que ha jugado la carta de valor más alto del palo de muestra, o si no hay una muestra, la carta de mayor valor del palo que empezó la mano. Después cada jugador toma una carta para volver a tener tres cartas y esto se hace hasta agotar la baraja y la mano de cada jugador.

Se puede cambiar la carta muestra, si la carta muestra es más alta que un siete, el jugador que tenga el siete del palo muestra y haya ganado una mano puede, cambiar este siete por la carta muestra, obteniendo en el cambio una de mayor valor. Si por el contrario la carta muestra es igual o menor a siete, el jugador que tenga un dos del palo de la muestra y haya ganado la mano, puede cambiar el dos por la carta muestra. Esta operación se puede implementar con una leyenda que le indique al jugador si desea hacer el cambio. Una vez agotada la baraja, se acaba la partida. En este momento se suman los valores de las cartas que haya ganado cada jugador y el que más tenga es el ganador. Es posible un empate.

<b>UACM</b> Universidad Autónoma de la Ciudad de México <small>NADA HUMANO ME ES AJENO</small>	<b>Introducción a la ingeniería de software</b> <b>Planeación</b>	
<b>Producto</b>	<b>Proyecto de Brisca</b>	
<b>Emitido por</b>	<b>Flores Trejo Victor Ruben</b>	<b>Fecha: 10/09/23</b>

#### **Adicionales:**

Se dará puntos extras si se implementa una estrategia de juego para la computadora. En otro caso de podrá realizar de manera aleatoria. Implementar una manera de llevar el puntaje de un jugador después de cada mano. Implementar una manera de jugar por puntos. Es necesario llevar un registro del puntaje de cada partida. Esto quiere decir, jugar las partidas necesarias hasta alcanzar el puntaje

### **2.-Planteamiento:**

La implementación de un juego de Brisca para un jugador contra la computadora presenta desafíos significativos en términos de diseño, programación y experiencia del usuario. El juego de Brisca es un juego de cartas tradicional con reglas específicas que deben ser replicadas con precisión. Además, se requiere una estrategia de juego para la computadora que haga que el juego sea desafiante y atractivo.

### **3.- Objetivo:**

Crear un juego de brisca en lenguaje C que sea funcional, en el cual un jugador se pueda enfrentar a la computadora. Además, se deben de replicar las reglas tradicionales de manera correcta y ejecutarse en la terminal del lenguaje.

#### **3.1.-Objetivos principales:**

- Desarrollar un juego funcional para un jugador (usuario) contra la banca (computadora)
- Seguir las reglas del juego brisca, asegurarse que se sigan las reglas básicas (valores de cartas, jugadas validas, puntajes, etc.)
- Diseñar una interfaz en consola, que represente de manera clara y comprensible las cartas y su información.
- Implementar lógica de la banca (computadora), que permita tomar decisiones en el curso del juego, ofreciendo un desafío adecuado al jugador.
- Realizar pruebas rigurosas para lograr identificar y corregir cualquier error o problema que pueda afectar la experiencia de juego.

<b>UACM</b> Universidad Autónoma de la Ciudad de México NADA HUMANO ME ES AJENO	<b>Introducción a la ingeniería de software</b> <b>Planeación</b>	
<b>Producto</b>	<b>Proyecto de Brisca</b>	
<b>Emitido por</b>	<b>Flores Trejo Victor Ruben</b>	<b>Fecha: 10/09/23</b>

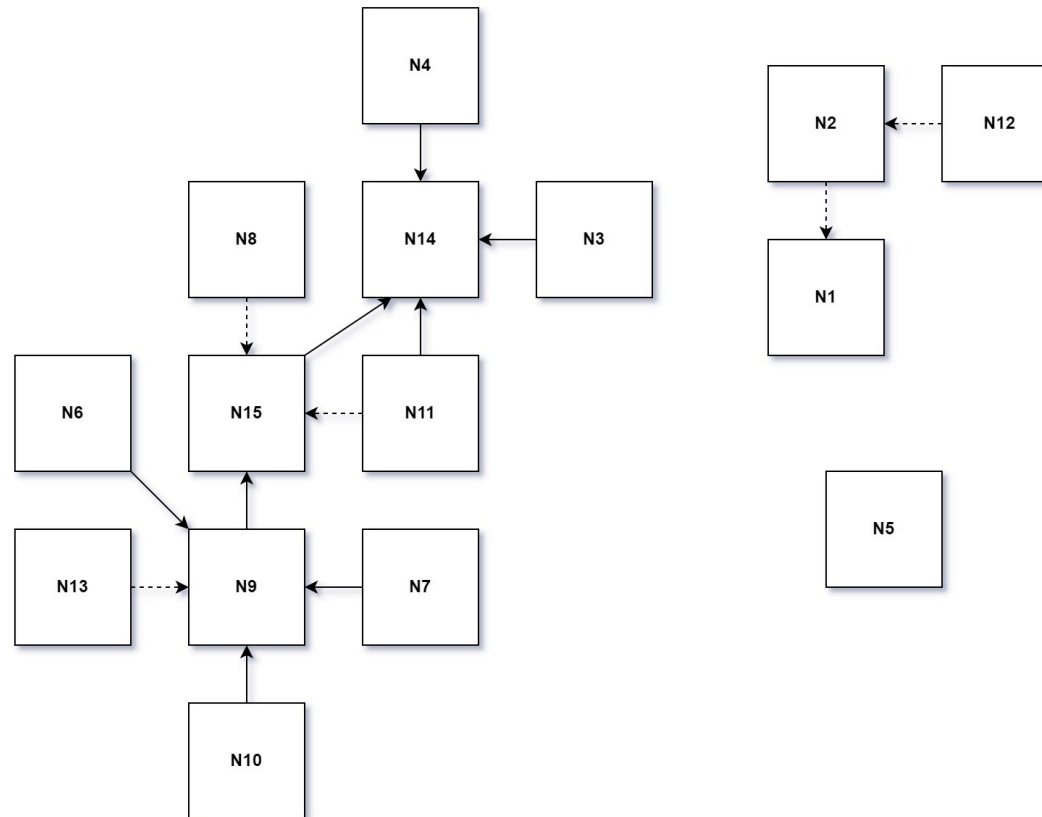
Forma de Registro de riesgos	
<b>Nombre: Victor Ruben Flores Trejo</b>	<b>Fecha: 03/09/2023</b>
<b>Semana: 1</b>	<b>Ciclo: 1er ciclo</b>

RIESGO	PROBABILIDAD DE OCURRENCIA	IMPACTO	ESTRATEGIA PARA MANEJAR EL RIESGO
1.-Representación de la baraja en la terminal: Desafíos en términos de diseño y visualización, lo que podría impactar la estética del juego.	Alto	Alto	Considera el uso de bibliotecas o herramientas específicas para el diseño de la interfaz de usuario en la terminal. Además, realiza pruebas de usabilidad
2.-Errores en las reglas del juego: La valoración de las cartas y las condiciones de victoria, podría ser más compleja de lo esperado y requerir un tiempo significativo de desarrollo.	Moderado	Alto	Es esencial comprender completamente las reglas del juego y verificar constantemente que se implementaron de manera correcta, esto con pruebas exhaustivas para garantizar la precisión.
3.-Implementación de la estrategia de juego para la banca: El diseño de una estrategia de juego efectiva para la banca que tome decisiones realistas y desafiantes puede ser complejo.	Alto	Moderado	Investigar estrategias de juegos, para implementarlas en la parte de la banca, se debe de planificar y diseñar de manera cuidadosa. Realizando pruebas para garantizar que la estrategia fue implementada con éxito.
4.-Registro del puntaje y juego por puntos: No se maneje de forma adecuada el registro y la gestión de puntajes, generando puntajes incorrectos e insatisfacción al usuario.	Moderado	Alto	Realizar pruebas exhaustivas a medida que implementamos funciones y necesidades, dividir las necesidades para asegurar la efectividad.
5.-Dificultad en distribución de cartas de formas aleatoria: Asegurarse de que las cartas se distribuyan de manera aleatoria y justa puede ser un desafío, especialmente si no se implementa correctamente.	Moderado	Medio	Realizar una función con librerías que nos puedan ayudar a generar la repartición de forma aleatoria de nuestra baraja y hacer pruebas de esto.
6.- Falta de experiencia en lenguaje c: El programador no esté del todo familiarizado con el lenguaje.	Moderado	Alto	Capacitación en el lenguaje y proporción de recurso para un entendimiento optimo.
7.-Cambio en los requisitos: El juego está sujeto a cambios que el cliente pueda desear en cualquier momento.	Moderado	Moderado	Utilización de una metodología para el desarrollo, la cual sea adaptativa a los cambios en los requisitos de manera flexible.

<b>UACM</b> Universidad Autónoma de la Ciudad de México <small>NADA HUMANO ME ES AJENO</small>	<b>Introducción a la ingeniería de software</b> <b>Planeación</b>	
<b>Producto</b>	<b>Proyecto de Brisca</b>	
<b>Emitido por</b>	<b>Flores Trejo Victor Ruben</b>	<b>Fecha: 10/09/23</b>

Forma de estrategia	
<b>Nombre: Victor Ruben Flores Trejo</b>	<b>Fecha: 03/09/2023</b>
<b>Semana: 1</b>	<b>Ciclo: 1er ciclo</b>

### GRÁFICAS DE DEPENDENCIA




<b>UACM</b> Universidad Autónoma de la Ciudad de México <small>NADA HUMANO ME ES AJENO</small>	<b>Introducción a la ingeniería de software</b> <b>Planeación</b>	
<b>Producto</b>	<b>Proyecto de Brisca</b>	
<b>Emitido por</b>	<b>Flores Trejo Victor Ruben</b>	<b>Fecha: 10/09/23</b>

<b>JUSTIFICACIÓN DE LA ESTRATEGIA</b>
1.- Diseño de la estructura del programa: Se debe planificar y diseñar la estructura general del programa (representación de cartas y mano de los jugadores) es esencial para tener la base antes de comenzar a implementar (Ciclo 1).
2.- Representación de las cartas y barajas: Representar la baraja española y las cartas, fundamental para poder empezar a manejar el juego (Ciclo1).
3.- Repartición de cartas: Desarrollar la función que reparta de forma aleatoria las cartas a cada jugador, asegurándose que estas sean eliminadas del maso, evitando repeticiones en las partidas (Ciclo 1).
4.- Reglas básicas: Implementación de la lógica que permita ejecutar partidas entre usuario y consola. Por el momento no hemos considerado la lógica de la banca (computadora), solo contamo0s con la mecánica básica (Ciclo 1).
5.- Determinar la mano ganadora: Lógica necesaria para determinar que jugador gana la ronda en función de las reglas del juego, es muy importante para poder seguir avanzando en el juego (Ciclo 1).
6.- Finalización del juego: Implementar las condiciones para finalizar el juego, ya sea que se agota la baraja o cuando se alcance el puntaje objetivo. Permitiendo que las partidas se jueguen de inicio a fin (Ciclo 1).
7,- Llevar el registro de puntos: Agregar la funcionalidad para llevar el registro de puntos de cada jugador a lo largo del juego, manteniendo un seguimiento de quien está ganando (Ciclo 2).
8.- Lógica de la banca (computadora): Implementar estrategia de juego para la computadora, involucra algoritmos para que la computadora seleccione cartas de manera estratégica y sea desafiante (Ciclo 2).
9.- Pulir la interfaz en la terminal: Diseña la interfaz que mostrara las cartas para hacer un poco más intuitivo el juego (Ciclo 2).
10.- Cambio de cartas por parte del usuario: Permitir al jugador cambiar su carta por la carta muestra, esto si cumple con las condiciones adecuadas, como es tener el siete o el dos del palo muestra (Ciclo 2).

<b>UACM</b> Universidad Autónoma de la Ciudad de México <small>NADA HUMANO ME ES AJENO</small>	<b>Introducción a la ingeniería de software</b> <b>Planeación</b>	
<b>Producto</b>	<b>Proyecto de Brisca</b>	
<b>Emitido por</b>	<b>Flores Trejo Victor Ruben</b>	<b>Fecha: 10/09/23</b>

FUNCIONALIDADES Y NECESIDADES
N1.-Simular juego de brisca con un solo jugador (usuario) contra la banca (computadora). Crear una interfaz de usuario en la terminal para que los jugadores interactúen con el juego
N2.-Implementar el juego en lenguaje c El juego se debe de implementar en el lenguaje de programación c para su ejecución
N3.-Escoger de forma aleatoria una carta muestra Seleccionar al azar una carta que sirva como referencia para la partida
N4.-Repartir 3 cartas de la baraja de forma aleatoria para cada jugador. Distribuir cartas aleatorias a los jugadores al inicio del juego
N5.-Decidir por sorteo qué jugador comienza. Se determina al azar cuál de los dos jugadores (usuario o banca) inicia la partida
N6.-Decidir cuál es la mejor mano. Evalúa las cartas en la mano de cada jugador para determinar quién tiene la mejor combinación de cartas
N7.-Tomar una carta para volver a tener 3 cartas en la mano. Permitir a los jugadores reponer cartas en su mano después de cada jugada
N8.-Cambio de carta muestra. Implementar la lógica para cambiar la carta muestra según las reglas del juego
N9.-Suma de valores de cartas. Calcula la suma de los valores de las cartas en la mano de cada jugador para determinar el resultado de cada ronda
N10.-Manejo de puntaje y registro. Registrar y actualizar los puntajes de los jugadores a lo largo de la partida
N11.-Representación de la baraja. Utilizar una estructura de datos para representar y gestionar las cartas de la baraja
N12.-Estrategia de la banca (computadora). Definir una estrategia de juego para que la banca (computadora) tome decisiones automáticas durante la partida
N13.-Finalización del juego. Establecer las condiciones para determinar cuándo concluye la partida y calcular el resultado final del juego
N14.-Baraja española de 40 cartas. El juego requiere una baraja específica con 40 cartas divididas en cuatro palos: oros, copas, espadas y bastos
N15.-Cartas y sus valores. Es necesario establecer los valores de las cartas en cada palo, asignando puntuaciones a las cartas as, tres, rey, caballo y sota

 <p><b>UACM</b>          Universidad Autónoma          de la Ciudad de México          NADA HUMANO ME ES AJENO</p>	<p><b>Introducción a la ingeniería de software</b>  <b>Planeación</b></p>	
	<p><b>Producto</b></p>	<p><b>Proyecto de Brisca</b></p>
<p><b>Emitido por</b></p>	<p><b>Flores Trejo Victor Ruben</b></p>	<p><b>Fecha: 10/09/23</b></p>

Forma semanal personal	
Nombre: Victor Ruben Flores Trejo	Fecha: 03/09/2023
Semana: 1	Ciclo: 1er ciclo

<b>PRODUCTOS DE DESARROLLO GENERADOS</b>	<b>TIEMPO DEDICADO</b>	<b>TAMAÑO DEL PRODUCTO</b>
<b>REPOSITORIO GITHUB</b>	1 hora	
<b>OBJETIVOS, PLANEACIÓN, RIESGOS Y ESTRATEGIAS</b>	2 horas	120 kb

<b>DATOS GLOBALES DEL PROYECTO EN ESTA SEMANA</b>	<b>TIEMPO</b>
<b>TIEMPO DEDICADO AL PROYECTO EN ESTA SEMANA</b>	4 horas
<b>TIEMPO DEDICADO AL PROYECTO EN ESTE CICLO HASTA ESTA SEMANA</b>	4 horas
<b>PRODUCTOS TERMINADOS DE ESTA FASE EN ESTA SEMANA</b>	2

SEGUIMIENTO DE RIESGOS	ESTADO
INDICACIÓN DE LA ESTRUCTURA DEL REPOSITORIO	COMPLETADO
ESTANDAR DE DOCUMENTACIÓN	COMPLETADO

<b>UACM</b> Universidad Autónoma de la Ciudad de México NADA HUMANO ME ES AJENO	<b>Introducción a la ingeniería de software</b> <b>Planeación</b>	
<b>Producto</b>	<b>Proyecto de Brisca</b>	
<b>Emitido por</b>	<b>Flores Trejo Victor Ruben</b>	<b>Fecha: 10/09/23</b>



Link de repositorio: [https://github.com/VRFT/Proyecto\\_Brisca.git](https://github.com/VRFT/Proyecto_Brisca.git)