Universidad Juárez Autónoma de Tabasco

División Académica de Ciencias y Tecnologías de la Información

Estructura de Datos

Proyecto

Interfaz Gráfica del Juego Uno

Alumno:

David Ricardo Santiago Alcoser

Matrícula:

212H17118

Docente:

José Luis Méndez Ramos

Índice

[Las reglas del juego son 4](#_Toc121959069)

[¿Cómo funciona el programa? 4](#_Toc121959070)

[Representación gráfica del mecanismo de colas 5](#_Toc121959071)

[Cambio de dirección en sentido contrario 5](#_Toc121959072)

[Cambio de dirección al sentido normal 6](#_Toc121959073)

[Cancelación de turno por algún comodín en sentido normal 6](#_Toc121959074)

[Cancelación de turno por algún comodín en sentido contrario 7](#_Toc121959075)

[Archivos y sus funciones 8](#_Toc121959076)

[Tda\_cola\_lineal 8](#_Toc121959077)

[Tda\_cola\_circular\_doble 8](#_Toc121959078)

[Tda\_pila 8](#_Toc121959079)

[banco\_aleatorio 8](#_Toc121959080)

[Colores 9](#_Toc121959081)

[Selec\_color 9](#_Toc121959082)

[escoger\_color 9](#_Toc121959083)

[funciones\_colas 9](#_Toc121959084)

[impresion 10](#_Toc121959085)

[insertar 10](#_Toc121959086)

[Insertar\_inicio 10](#_Toc121959087)

[buscar 11](#_Toc121959088)

[NoEncontrado 11](#_Toc121959089)

[primero 11](#_Toc121959090)

[quitar 11](#_Toc121959091)

[vacio 11](#_Toc121959092)

[reiniciar 11](#_Toc121959093)

[comprobar\_c 12](#_Toc121959094)

[Funciones de com\_comodin 13](#_Toc121959095)

[Encontrar\_cd 13](#_Toc121959096)

[Interfaz gráfica (programa principal) 14](#_Toc121959097)

[Botón guardar 15](#_Toc121959098)

[Botón Jugar 15](#_Toc121959099)

[Repartido de cartas y guardado de barajas en su cola general 16](#_Toc121959100)

[Generación de la última carta puesta 16](#_Toc121959101)

[Funciones de ventanas emergentes 16](#_Toc121959102)

[Botón sacar 16](#_Toc121959103)

[Botón poner 17](#_Toc121959104)

[Comprobar ganador 17](#_Toc121959105)

[Función cambiar color 18](#_Toc121959106)

[Función cambiar 18](#_Toc121959107)

Interfaz Gráfica del Juego Uno

La creación de este proyecto se hizo con el fin de la demostración del uso de las estructuras de datos llamadas colas circulares doblemente enlazadas, colas lineales pilas en un caso muy común que la gran mayoría de jóvenes en algún momento de nuestras vidas hemos visto que es el juego de cartas “uno”. Dicho juego consiste en ir por turnos poniendo cartas sobre una pila de cartas, donde la carta a poner tiene que ser igual en color o número o ser algún comodín especial, de no tener una carta que sea compatible se sacan cartas de una pila de cartas que se le conoce como banco hasta que salga una carta compatible dándole el turno al jugador siguiente y así hasta que un jugador se quede sin cartas, entonces el juego termina; en este juego los principales comodines son: cambiar color, agregarle cartas al jugador que sigue, cancelar turno y cambiar dirección del juego. Según la versión del uno habrá más comodines, pero estos son los principales que hay en todas las versiones.

Las reglas del juego son:

* Poner cartas que sean igual en número o color
* El comodín cambiar dirección hace que los turnos vayan en sentido contrario al que ya se estaba jugando
* El comodín cancelar turno hace que el jugador siguiente del último que puso su carta, pierda su turno.
* El comodín +2 le agrega 2 cartas al jugador siguiente del turno anterior, y el jugador que recibe las cartas pierde su turno.
* El comodín cambia color es compatible con cualquier última carta puesta y esta cambia el color que será compatible con las cartas a pones
* El comodín +4 cambia el color compatible con las partas a poner, le añade 4 cartas al jugador siguiente y hace que este pierda su turno.
* Gana el primer jugador en quedarse sin cartas

# ¿Cómo funciona el programa?

Todo el juego depende principalmente de 3 colas circulares doblemente enlazadas, en una se guardan los nombres de los jugadores, otra cola guarda toda la baraja del jugador es decir hay una cola de baraja por cada jugador (máximo se pueden 4 jugadores), y hay una cola general de barajas la cual guarda todas las barajas de jugadores. El turno siempre será el jugador que este como primer dato en la cola de jugadores de igual forma su baraja será la primer baraja que este en la cola general de barajas, para seguir de turno solo se quita el jugador de la primera parte y se pasa al final de la cola, entonces el turno es del primer jugador actual en la cola, lo mismo ocurre con la cola general de barajas, se quita la baraja del principio y se

pone al final de la cola, así la baraja del jugador en turno es la baraja que se encuentro en la primera posición de la cola general de barajas. Este algoritmo se repite hasta que la baraja de algún jugador quede vacía.

## Representación gráfica del mecanismo de colas

jugadores Barajas Baraja individual

Jugador

en turno

Baraja 3

Baraja 2

Baraja 1

Jugador 3

Jugador 2

Jugador 1

Carta 1

Carta 2

}}

Carta 3

Siguiente turno

jugadores Barajas

Jugador

en turno

Baraja 1

Baraja 3

Baraja 2

Jugador 1

Jugador 3

Jugador 2

El sentido normal del juego es sacar el primer dato y pasarlo abajo

El sentido anormal o contrario es sacar el último dato y pasarlo al principio

### Cambio de dirección en sentido contrario

jugadores Barajas

Jugador

en turno

Baraja 3

Baraja 2

Baraja 1

Jugador 3

Jugador 2

Jugador 1

jugadores Barajas

Jugador

en turno

Baraja 2

Baraja 1

Baraja 3

Jugador 2

Jugador 1

Jugador 3

### Cambio de dirección al sentido normal

jugadores Barajas

Jugador

en turno

Baraja 3

Baraja 2

Baraja 1

Jugador 3

Jugador 2

Jugador 1

jugadores Barajas

Jugador

en turno

Baraja 1

Baraja 3

Baraja 2

Jugador 1

Jugador 3

Jugador 2

### Cancelación de turno por algún comodín en sentido normal

jugadores Barajas

Jugador

en turno

Baraja 1

Baraja 3

Baraja 2

Jugador 1

Jugador 3

Jugador 2

jugadores Barajas

Jugador

en turno

Baraja 3

Baraja 2

Baraja 1

Jugador 3

Jugador 2

Jugador 1

### Cancelación de turno por algún comodín en sentido contrario

jugadores Barajas

Jugador

en turno

Baraja 3

Baraja 2

Baraja 1

Jugador 3

Jugador 2

Jugador 1

jugadores Barajas

Jugador

en turno

Baraja 1

Baraja 3

Baraja 2

Jugador 1

Jugador 3

Jugador 2

# Archivos y sus funciones

El proyecto usa varias importaciones de programas externos, esto para facilitar la implementación de las funciones y métodos de clases usadas para realizar el proyecto.

## Tda\_cola\_lineal

Las estructuras de datos se manejaron como objetos, en este archivo se encuentra la clase objeto “Cola Lineal” el cual me permite acomodar mis datos en dicha estructura respetando sus propiedades (estructura de tipo fifo, es decir lo que entra primero sale primero). Entre los métodos de la clase se encuentran: imprimir la cola, quitar datos de la cola, agregar datos a la cola, ver el primer dato de la cola, reiniciar la cola y verificar si la cola está vacía.

## Tda\_cola\_circular\_doble

En este archivo se encuentra la clase objeto “Cola Circular Doblemente Enlazada” la cual permite sacar y agregar datos tanto del final como del inicio de la cola, es decir se comporta como una cola lineal y como una pila de datos. Entre los métodos de la clase se encuentran: imprimir la cola de manera normal (como una cola lineal), imprimir la cola de manera inversa (como una pila de datos), métodos de agregar y quitar datos tanto al inicio como al final, métodos que retornan el primero o el último dato, reiniciar la cola y por último verificar si la cola está vacía.

## Tda\_pila

Aquí se encuentra la clase objeto “Pila la cual tiene un almacenamiento de tipo lifo (lo primero que entra sale a lo último), es la estructura más simple de todas, sus métodos son: reiniciar la pila, insertar un dato, quitar un dato, verificar si la pila está vacía, ver el tope de la pila (el último dato) y finalmente imprimir la pila.

## banco\_aleatorio

En este archivo de encuentra la función b\_aleatorio, la cual se encarga de generar de generar una carta de manera aleatoria con valores entre 1 y 15, valores entre 1 y 10 son cartas normales, y los valores que se generen entre 11 y 15 serán comodines de tal modo que el número 11 sería el +2, el 12 el +4, el 13 el cambiar de dirección, el 14 el cancelar turno, y el 15 el cambiar color. Dicha función se llama dentro de un ciclo n veces según la cantidad de cartas que se necesiten para repartir entre los jugadores.

En este archivo se importa la función selec\_color, la cual igual de manera aleatoria asigna un color a las cartas normales y a los comodines cambiar dirección y cancelar turno, mientras que a los comodines cambiar color y +4 les asigna el color de negro, para finalmente retornar la carta con su color asignado.

## Colores

En este archivo se encuentran las funciones selec\_color y escoger\_color

### Selec\_color

Está función recibe como parámetro el valor que retorna la función banco\_aleatorio el cual puede ser un número entre 1 y 10 (carta normal) o un string que son los comodines ya mencionados anteriormente. Está función genera números aleatorios entre el 1 y el 4 así dependiendo el número se le asignará un color, si el número es 1 el color es rojo, si es 2 el número es azul, si es 3 el color es amarillo y si es 4 el color es verde. Posterior si el valor recibido es un comodín +4 o cc (cambia color) retorna ese valor concatenado con un espacio y la palabra negro, si es cualquiera de los otros comodines retorna dicho valor concatenado con un espacio y el color generado aleatoriamente; si el valor recibido es el número 10 retorna dicho valor, pero vuelto cadena, concatenado con un espacio y el color generado aleatorio. Finalmente, si el dato recibido es un número entre 1 y 9 retorna un 0 como cadena concatenada del dato recibido hecho string, más un espacio seguido del color generado aleatoriamente.

### escoger\_color

Está función es llamada cuando se puso un +4 o un cambia color (cc) y sirve para cambiar el color de juego ya que eso es lo que permiten estos comodines. Recibe como parámetro un string el cual es una palabra que diga el color a cambiar, dicho color puede ser verde, azul, amarillo o rojo. Retorna una cadena compuesta por un 00 seguido de un espacio seguido del color ingresado.

Por ejemplo, si se escribió el color “azul” retornará “00 azul”, esto para que se tome como una carta y se ponga al final de la pila de cartas puestas y así la carta que siga sea del color que se escogió.

## funciones\_colas

Como se explicó anteriormente la cola baraja que es una cola por jugador, se mete dentro de la cola general de barajas, es decir tenemos colas dentro de una cola, tal como se muestra en la imagen:

Baraja 2 Barajas Baraja 1

Baraja 2

Baraja 1

Carta 3

Carta 2

Carta 1

Carta 1

Carta 2

}}

Carta 3

Entonces si por ejemplo si se necesitaba imprimir el primer dato que sería la baraja 1 el programa no imprimiría las cartas que conforman la baraja 1, si no que imprimiría la dirección de memoria donde se guarda la baraja 1 dentro de la cola general de barajas; así se presentarían varios casos similares donde se necesita acceder o quitar algún dato en específico de la baraja que este en primera posición.

Para darle solución a estos casos se crearon las funciones de colas, las cuales reciben como parámetro principal un dato de tipo cola, es decir una estructura, en dichas funciones se importa el tipo “tda” requerido para así poder acceder a los métodos propios de la baraja en primera posición y trabajar con las acciones requeridas en el programa.

## impresion

Este método retorna el método de imprimir de la cola recibida como parámetro, se usa para imprimir la baraja del jugador que está en turno, es decir la impresión de la baraja en primera posición.

## insertar

Recibe una cola de datos que es la baraja en primera posición, junto con un dato de tipo string el cual sería una carta generada dentro del programa principal de manera aleatoria. El programa retorna la cola recibida con el método insertar al final actuando sobre el dato string, en otras palabras esta función sirve para agregar más cartas al mazo del jugador.

## Insertar\_inicio

Esta función recibe una cola y una carta aleatoria, retornando la cola con su método de insertar al inicio actuando sobre la carta aleatoria. Sirve y hace lo mismo que la función anterior solo que en vez de insertar al final inserta al inicio.

Insertar al inicio Inserta al final

Carta aleatoria

Carta 3

Carta 2

Carta 1

Carta 1

Carta 2

Carta 3

Carta aleatoria

## buscar

La función buscar es usada cuando el usuario desea ingresar una carta, así esta función recibe como parámetro la cola de la baraja en primera posición y el nombre de la carta para buscar. Lo primero que se hace es vaciar la cola de barajas en una cola auxiliar para ir contando cuantas cartas tiene el jugador, después se regresan las cartas a la cola original de la baraja. Posterior a eso se quita la carta del principio de la cola y se manda al final bajo la condición de que el primer dato sea diferente a la carta que se desea buscar y que la cantidad de veces que se haga ese ciclo sea menor o igual a la cantidad de cartas registradas al inicio de la función.

Si se termina el ciclo y la cantidad de veces que se repitió es menor o igual a la cantidad de cartas registradas y la primera carta sea igual a la que se busca se retorna el primer dato de la cola, de no ser así se retorna un falso.

## NoEncontrado

Está función no interactúa con colas. Es llamada desde el programa principal si la función “buscar” retorno un falso, la acción que hace es abrir una ventana con un mensaje que indica que la carta que se busca no existe en la baraja

## primero

Recibe la baraja en primera posición y la retorna con su método de primer dato, se usa para ver la primera carta de la baraja del jugador.

## quitar

recibe la baraja enviada y sirve para quitar la carta del principio de dicha baraja, retorna la cola recibida con su método de quitar del principio. En el programa principal se usa para quitar la carta del mazo una vez que se haya evaluado si es compatible con la última carta puesta.

## vacio

Esta función sirve para verificar si un jugador se quedó sin cartas. Recibe como parámetro la baraja en turno y mediante el método de cola vacía determina si la cola esta vacía o no, esto sería como decir si hay cartas o no, si es verdad retorna verdadero, si no está vacía retorna un falso.

## reiniciar

Está función fue creada para comprobar que el programa funciona correctamente, en el archivo principal esta función se encuentra como comentario, lo que hace es que deja al jugador sin cartas en su baraja. El método reiniciar propio de la cola definido en su “tda” lo que hace es que iguala los datos a “nulo”, por lo tanto, la estructura ya no tiene datos. Retorna la baraja recibida reiniciada.

Entonces se puede decir que en el programa principal se usa para forzar a un jugador a quedarse sin cartas y terminar el juego, y así verificar que termina de manera correcta.

## comprobar\_c

Está es la función que válida si la carta a poner es compatible con la última carta puesta, recibe la última carta puesta que sería el tope de la pila donde se ponen las cartas y la carta que el jugador desea poner extraída ya de su baraja.

Primero comprueba si la carta a poner es un cambia color o un +4 pues ambos son compatibles con cualquier color de la última carta puesta, de ser así retorna un verdadero.

En caso de que las cadenas que serían las cartas sean diferentes, por ejemplo, la última carta puesta es “03 rojo” y la carta a poner es un “03 azul”, ambas cadenas son diferentes por lo tanto se procede a lo siguiente:

* Se declaran 4 variables
* La primera variable sería igual al carácter en la posición 1 de la cadena de la carta que se desea poner
* La segunda variable sería igual al carácter en la posición 1 de la cadena de última carta puesta
* La tercera variable es igual al carácter en la posición 3 de la cadena de la carta que se desea poner
* La cuarta variable es igual al carácter en la posición 3 de la cadena de la última carta puesta

Se usan las posición 1 y 3 de las cadenas porque recordemos que las posiciones en una cadena se inician desde 0, las cadenas son como la siguiente expresión “03 rojo”, por lo tanto la posición 0 en todas las cartas tiene el valor de 0, en la posición 1 cada carta tiene su valor numérico diferente y en la posición 3 se encuentra una letra distintiva de cada calor.

03 rojo

04 amarillo

03 verde

08 azul

Como se puede ver con la evaluación de números no hay problema es simple, pero con la evaluación de colores no se puede evaluar la primera letra de cada color ya que el azul y el amarillo empiezan con la misma letra, entonces serían compatibles cartas azules con cartas amarillos y esto no está permitido dentro del juego, por eso se usa la segunda letra de cada color pues la segunda letra es irrepetible haciendo más clara y única la diferenciación de cada color.

Siguiendo con el procedimiento de la función, esta va a retornar un verdadero si la posición 1 de la última carta es igual a la posición 1 de la carta que se desea poner o si la posición 3 de la última carta es igual a la posición 3 de la carta que se desea poner, de no cumplirse ninguna de las condiciones se retorna un falso.

## Funciones de com\_comodin

Las funciones de com\_comodin son 4 a pesar de que son 5 comodines, sirven para evaluar si la última carta puesta es un comodín, se evalúa que las cartas sean igual a +4, cambiar color, cancelar turno o +2 (se toman en cuenta también los colores del +2 y cancelar turno puesto que existe un comodín de cada una por color). En el caso del cambiar dirección se hace su propia evaluación en el programa principal tomando en cuenta que sean validos cambiar de dirección de cualquier color.

Estas funciones retornan un verdadero si la última carta puesta es el comodín que evalúan, de no ser así retornan un falso.

## Encontrar\_cd

Esta función se creo para evaluar el número de veces que se puso un cd (cambiar dirección) en la pila de cartas a poner. Se manda un contador desde el programa principal y si ese contador dividido entre 2 da como residuo diferente de 0 retorna un verdadero, si da como residuo 0 retorna un falso.

Esto se hace porque si la cantidad de veces que se pone un cambiar dirección es par se juega a sentido normal, pero si es impar se juega a sentido contrario.

# Interfaz gráfica (programa principal)

La interfaz gráfica está compuesta por:

Etiquetas:

1. Nombre
2. Carta
3. Jugador en turno
4. Ganador
5. Última carta puesta
6. Baraja del jugador
7. Jugadores
8. Color para cambiar

Entradas de texto:

* + 1. Para nombres
    2. Para la carta a poner
    3. Para el color a cambiar

Salidas de texto para:

1. El jugador en turno
2. La baraja del jugador
3. Los jugadores participando
4. Ganador
5. Última carta puesta

Lo único habilitado hasta este momento para el usuario es la entrada de texto donde se ingresa el nombre y el botón guardar, así como los botones jugar y poner. El primero depende que de hayan 2 o más jugadores, si solo hay uno se habré una ventana que indica que con un solo jugador no se puede, el segundo depende de la entrada de texto de la carta a poner la cual al estar deshabilitada no se le puede ingresar ninguna palabra por lo tanto al intentar poner una carta abrirá una ventana la cual indica que se registre la carta.

Solo si hay 2 o más jugadores registrados el botón jugar actúa, el botón sacar se habilita y la entrada de datos de carta se habilita, lo cual permite el ingresar cartas para su búsqueda y ponerlas.

El usuario solo podrá interactuar con los botones de poner y sacar, así como con la entrada de datos de cartas. Y solo si pone un cambiar color o un +4 se habilita la entrada de cambiar color y el botón que cambia color, después de que se cambie el color esta última entrada y botón se deshabilitan.

Cabe aclarar que en cada proceso hay momentos donde se habilitan y deshabilitan las salidas de datos para poder interactuar con ellas y enviar los datos que el usuario necesita ver de tal modo que el usuario puede ver los contenidos que necesita sin poder alterar a estos.

## Botón guardar

Para iniciar el juego lo primero que hay que hacer es registrar los nombres de los jugadores se pueden guardar máximo hasta 4 jugadores, una vez registrados los 4 jugadores se inicia el juego en automático eliminando todos los elementos involucrados para el registro de nombres que son la etiqueta “Nombres:”, la entrada de texto de nombres y el botón guardar. Si se deja el espacio en blanco y se le da a guarda el programa no lo permite y habré una ventana indicando que no es válido el nombre.

## Botón Jugar

El botón jugar es el que inicializa el juego como tal, el cual solo si hay más de 2 jugadores actúa, de lo contrario se activa una ventana que avisa que un solo jugador no es válido para iniciar.

Lo primero que hace es contar cuantos jugadores se registraron, esto por medio de un ciclo el cual va insertando los jugadores en una cola circular mientras esos datos son extraídos de la cola original, esto para ir incrementando un contador. Después devuelve los jugadores a la cola original.

Evalúa si el contador es mayor a 1, si no es así llama a su ventana emergente respectiva por medio de una función. Por otra parte, si es mayor a 1 se destruye la etiqueta de nombres, la entra de nombres, se habilita la entrada de cartas, y se activa el botón sacar.

### Repartido de cartas y guardado de barajas en su cola general

Se multiplica la cantidad de jugadores registradas por 5 para generar un total de cartas que al dividirla entre los jugadores le toque 5 cartas a cada uno. Dichas cartas se insertan en la cola de banco de cartas (b\_cartas) ya sean tanto números y/o comodines aleatorios. Posterior a eso se vacía el banco de cartas original y se envían a una cola auxiliar (b\_cartasaux), pero lo que recibe esta cola auxiliar es la función selec\_color que recibe la carta que se quita del banco original; esto para que al momento de insertar las cartas al banco auxiliar se inserten las cadenas de datos ya con sus colores asignados. Para finalizar esta primera parte las cartas se sacan del banco auxiliar y se mandan al banco original.

Para el repartido de cartas se reparten entre los jugadores según la cantidad de jugadores que se hayan registrados, si se registraron 2 se reparte la cantidad de cartas total entre las 2 colas, la cola de baraja 1 y la cola de baraja 2, y luego se insertan la cola de barajas general la baraja 1 y 2. Si se registraron 3 jugadores se reparte la cantidad de cartas entre las 3 colas, la cola de baraja 1, la cola de baraja 2 y la cola de baraja3 y se insertan cada cola de baraja en la cola general de barajas, ocurre lo mismo con 4 jugadores solo que aquí se reparte entre 4 colas evidentemente y posteriormente se anexa cada cola de baraja en la cola general de barajas.

### Generación de la última carta puesta

Para iniciar la partida se requiere de una carta puesta, entonces en la pila de cartas (poner\_cartas) se inserta una carta aleatoria que no contempla comodines para generar, por medio de la función selec\_color y una variable llamada carta que puede ser cualquier número entre el 1 y el 10.

## Funciones de ventanas emergentes

Las funciones de ventanas emergentes que se llaman como tal ventana seguidas de un número o una palabra con guión, son funciones que son llamadas cuando el usuario intenta ingresar un dato incorrecto o accionar un botón sin haber ingresado correctamente el dato.

## Botón sacar

El botón sacar llama a la función a la función “insertar” la cual recibe la baraja en primera posición de la cola general y recibe la función de banco aleatorio “b\_aleatorio” para insertar una carta de manera aleatoria en el mazo del jugador en turno, para posterior imprimir nuevamente la baraja del jugador en la salida de barajas, pero ahora con la carta sacada aleatoria agregada.

## Botón poner

El botón poner declara una variable booleana llamada “winner” la cual servirá para comprobar si hay un ganador o no. Lo primero que evalúa es que en la entrada de cartas se haya ingresado una carta, de no haber ingresado una carta manda a llamar una función de ventana emergente la cual imprime un aviso de que ingrese una carta en donde corresponde.

Posterior a eso hace una comparación por medio de la función “buscar” la cual regresará un verdadero si la carta que se desea poner existe o un falso si la carta que se desea poner no existe en la baraja. Si es un falso llama a una función de ventana emergente para avisar que la carta no se encontró; pero si es un verdadero, por medio de la función “comprobar\_c” se comprueba si la carta a poner es compatible con la última carta puesta, para poder leer la carta a poner se usa la función de cola “primero”, si la primer función retorna un verdadero por medio del método de “insertar” de la pila de cartas puestas envía la función de cola “quitar” para que en la pila se inserte la carta que se retira del mazo del jugador. Pero si retorna un falso se irán generando cartas aleatorias e insertando en el mazo del jugador hasta que se generé una compatible con la última carta puesta. Sería el equivalente a sacar cartas del banco en el juego real.

### Comprobar ganador

Se evalúa si la baraja del jugador está vacía por medio de la función de cola vacío, si es así winner será verdadero si no está vacía winner será falso.

Si dicha variable es verdadera se manda jugador que puso la última carta a la salida de “ganador” y se borran todas las salidas menos la de jugadores, así como se deshabilitan todos los botones y se llama a una función de ventana emergente la cual muestra el mensaje de fin del juego.

En su caso contrario que sea falso se hace primero la comprobación del cambio de dirección el cual va sacando las cartas de su pila y las va metiendo a una pila auxiliar y cada que encuentra un “cd” incremente un contador. Después regresa las cartas a la pila original y llama a la función “encontrar cd” para determinar si el contador es par o impar, si es par se juega en sentido normal, si es impar se juega en sentido contrario. El sentido normal es que el jugador del principio pasa al final así como su baraja, el sentido contrario es que el jugador del final pasa al principio junto con su baraja.

#### Comprobación de comodines

Se evalúan los comodines con sus respectivas funciones, todos menos los cambios de dirección

##### Comodín +2

Si el comodín es un +2 de cualquier color le inserta 2 cartas aleatorias al jugador que seguía del último y este pierde su turno.

##### Comodín cancela turno

Se salta el turno del jugador que sigue, dependiendo el cambio de dirección la cancelación de turno será de principio a fin o de fin a principio

##### Comodín +4

Retrocede una posición para dar a entender que el que cambia el color fue el último en poner la carta +4, aunque la salida de turno sale el jugador que sigue del siguiente del último que puso la carta con su respectiva baraja, se inhabilita el botón sacar y la entrada de datos de cartas para que el juego no siga hasta que se cambie el color, esto último se hace por medio del botón cambiar para que se habilite la entrada de texto donde se escribe el color que se desea cambiar así como el botón que activa el cambio de color, e inhabilita el botón sacar y la entrada de cartas.

Para que el botón cambiar se habilite se llama antes a la función “cambiar color”

Avanza al turno del siguiente que puso la última carta que fue el +4 y por medio de un ciclo de longitud 4 le agrega 4 cartas a dicho jugador por medio de la función de cola “insertar”, después avanza al siguiente jugador que dependerá del sentido en el que se este jugando, ya que el que recibe las 4 cartas pierde su turno.

##### Comodín cambia color

Llama la función “Cambiar color” para habilitar y deshabilitar los mismos botones y entradas que cuando se pone un comodín +4.

## Función cambiar color

Esta función deshabilita el botón sacar y la entrada de cartas y habilita el botón “cambiar” para cambiar de color y la entrada para escribir el color que se desea cambiar.

## Función cambiar

Evalúa que se haya escrito un color si no se escribió llama una función de ventana para enviar el mensaje que indica que escriba el color, si se escribió un color se evalúa que sea uno de los 4 colores del uno, verde, rojo, azul o amarillo, si no es así llama una función de ventana que indica que el color no es válido, si es un color válido, se pone una carta con valor numérico de 00 en la pila de cartas puestas con el color que se ingresó. Por ejemplo: “00 azul”.

Vuelve a habilitar la entrada y botones deshabilitados