

PROYECTO TRUCO DE LAS 21 CARTAS

CONTENIDO

Introducción	2
Objetivo.....	2
El Truco de las 21 cartas	2
El programa	4
Especificación.....	4
Implementación	5
Generación de letras al azar	6
Algoritmo para tratamiento de las cartas	6
Restricciones de código	7
Se pide	7
Entrega del proyecto	8

INTRODUCCIÓN

"El que aprende y aprende y no practica lo que sabe, es como el que ara y ara y no siembra."

Platón

Aquí está el primer proyecto que planteará un reto bastante grande para ti. El proyecto anterior del **Adivinador** ya era un tanto difícil, pero ahora redoblamos la apuesta. En este caso, como ya habrás visto en el video de la clase, te propondremos realizar el **Truco de las 21 cartas**.

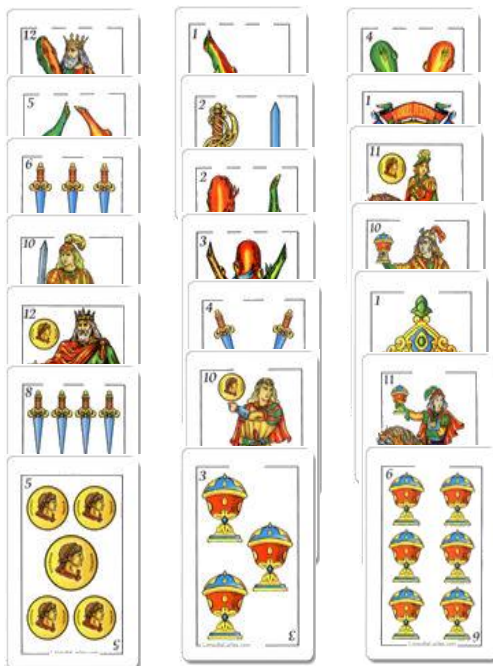
OBJETIVO

Crear un programa en Java que ejecute el **Truco de las 21 cartas** tal como será descrito en este documento, de forma tal que el programa adivine siempre la carta que el usuario está mirando en la pantalla, tal como sucede con el truco de cartas en la vida real.

Este proyecto plantea el uso de arreglos de forma que el/la estudiante entrene el uso de esta herramienta y se prepare para proyectos de mayor envergadura. A través de este proyecto el/la estudiante aplicará todo lo visto hasta ahora en el curso y utilizará su creatividad para encontrar una solución al problema planteado, mediante la creación de un programa de software. Quienes ya realizaron el curso de "Programación profesional desde CERO" tendrán como reto migrar su código Pascal a Java; quienes vienen desde otros cursos o escuelas, tendrán el reto de resolver el problema que supone programar este proyecto, lo cual es un refuerzo al aprendizaje de programación en sí.

EL TRUCO DE LAS 21 CARTAS

El **truco de 21 cartas** (que puede ser hecho con cualquier otro número de cartas que sea múltiplo de 3), funciona de la siguiente manera. El mago, o ejecutor, toma 21 cartas cualesquiera de un mazo común, y las presenta en tres grupos al espectador. Al ser 21 cartas cada grupo queda de 7 cartas:

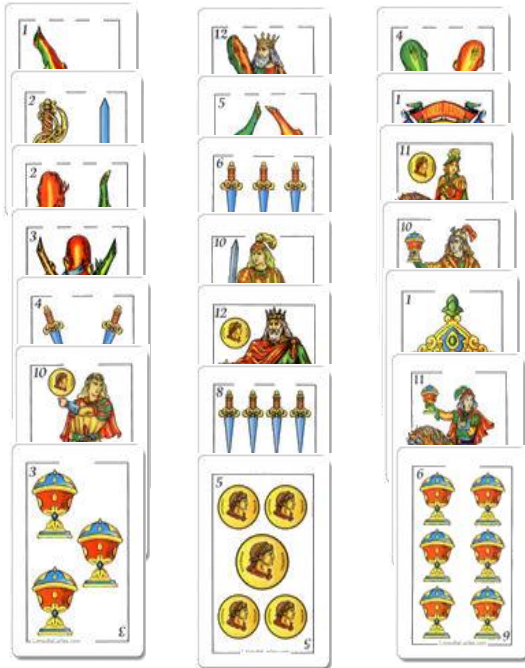


El mago le pedirá al espectador que elija una de esas cartas sin decirle cuál es. Lo único que el espectador tiene que hacer es señalar en cuál de esos tres grupos está la carta que tiene en mente.

Imagina por un momento que eres el espectador y el mago te ha presentado las cartas que tienes aquí a la izquierda, tal como se muestran allí. A ti te ha gustado el **5 de oro**, que está al final de la primera columna, así que le dices al mago que tu carta está allí, en la columna 1.

Lo que el mago hará entonces será juntar las cartas nuevamente en tres montones, pero sin desordenarlas. Así que el primer montón tendrá las 7 cartas que ves en la columna 1 en el orden en que están allí: **12 de basto**, **5 de basto**, **6 de espadas**, **10 de espadas**, **12 de oro**, **8 de espadas** y **5 de oro**. Lo mismo con las otras dos columnas.

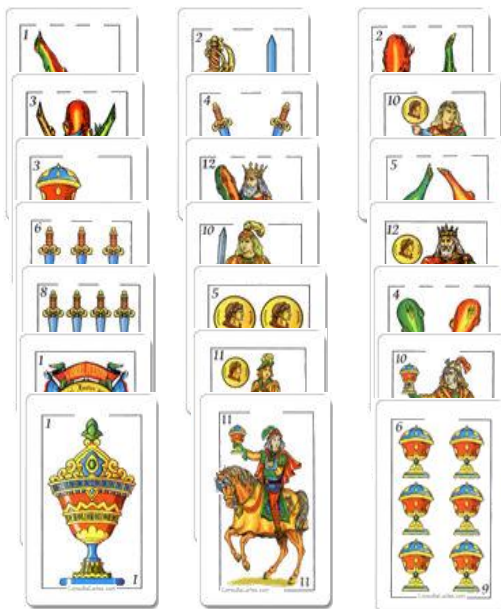
A continuación el mago juntará los montones en un único mazo pero teniendo la precaución de que el montón que tiene a la carta del espectador quede en medio de los otros dos:



En este caso entonces, al juntar las cartas en un único mazo quedarán en el orden que se ve a la izquierda. Como ves, el **5 de oro** estaría en la posición 14, luego del **8 de espadas**. El mago no sabe cuál es nuestra carta, solo hace lo que acabo de decir.

Lo que hará a continuación es volver a mostrar las cartas al espectador en tres montones, pero los repartirá intercalado. Es decir, tomará el mazo y pondrá la primera carta en la columna 1, la segunda en la columna 2, la tercera en la columna 3, luego la cuarta en la columna 1, la quinta en la columna 2, la sexta en la columna 3, y así hasta el final.

Esto dará como resultado un nuevo arreglo de cartas, tal como se muestra a continuación:

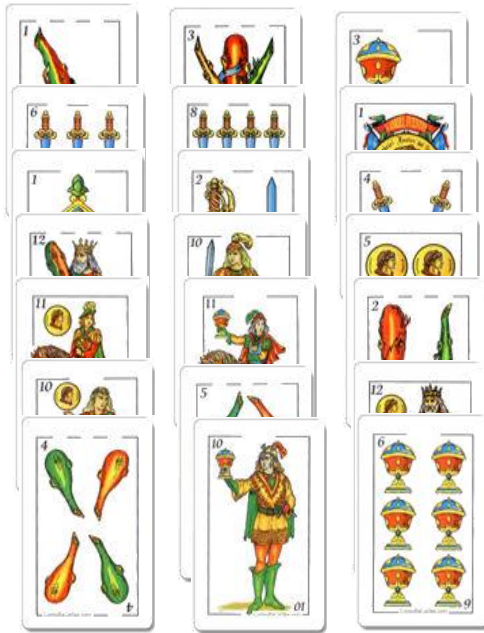


En esta segunda presentación al espectador las cartas han quedado tal como ves a la derecha, siendo esto el resultado del mecanismo en que se ha hecho. Lo que hará el mago ahora será pedirle al espectador que vuelva a indicar en qué columna o grupo está la carta que ha elegido, siendo obligatorio que la carta elegida en este caso sea la misma que en el anterior, de otro modo esto no tendría sentido.

Como habíamos elegido el **5 de oro** pues le señalamos al mago que está en el grupo 2. El mago volverá a hacer exactamente lo mismo que antes: juntará las cartas sin desordenarlas en sus tres respectivos grupos. Luego unirá los grupos en un único mazo teniendo la precaución de que el grupo con la carta elegida quede entre medio de los otros dos. En este caso el **5 de oro** ya está en el grupo de en medio, con lo cual el mago

simplemente junta los montones en ese orden (podría intercambiar los grupos 1 y 3 para confundir más al espectador si así lo quisiera).

Dado que en este caso dejaremos las cartas como están, su orden será ese mismo. Se puede apreciar que el **5 de oro** está en la posición número 12. Al volver a repartir las cartas de forma intercalada (la primera al grupo 1, la segunda al grupo 2, la tercera al grupo 3, la cuarta al grupo 1, y así), tenemos ahora el siguiente arreglo de cartas mostrado al espectador:



Nuevamente, el mago le pedirá al espectador que indique en cuál de los tres montones o grupos está la carta elegida (siendo obviamente la misma que al principio). En nuestro caso podemos ver que el **5 de oro** está en el tercer grupo de cartas, así que le indicamos eso al mago.

Otra vez se repetirá el proceso: el mago unirá las cartas sin desordenarlas en tres montones, uno por grupo. Luego unirá los montones ubicando aquel con nuestra carta entre medio de los otros dos.

Mágicamente, al hacer eso el **5 de oro** quedará en la posición número 11 (justo en medio) del mazo. Eso será así de forma indiscutible y funcionará siempre para cualquier carta que el espectador haya elegido, siempre que se hayan seguido estas sencillas reglas.

Lo único que tienes que hacer para adivinar la carta es contar las cartas y sacar la número 11. Como ves, la matemática es mágica.

EL PROGRAMA

El proyecto consiste en programar este truco de forma tal que el programa creado sea capaz de adivinar la carta que el usuario esté mirando en la pantalla, únicamente recibiendo como información el número de grupo en que está la carta (tarjeta) que el usuario está mirando.

ESPECIFICACIÓN

Se pide implementar un programa en Java que haga exactamente lo que se acaba de describir. En vez de utilizar cartas se utilizarán letras mayúsculas desde la **A** hasta la **U** (21 letras).

El programa ordenará las letras al azar, tal como si barajara un mazo, y luego las mostrará en tres grupos (columnas):

```
Haremos 3 secuencias. Empecemos...
Secuencia 1 :

    E   J   U
    Q   D   K
    I   M   N
    O   S   F
    P   R   B
    G   T   L
    C   H   A

En que grupo esta tu tarjeta [1,2,3]:
```

Como puedes observar, se muestra inicialmente el mensaje `Haremos 3 secuencias. Empecemos...`

Luego se muestra la secuencia actual, en este caso **Secuencia 1**, las cartas ordenadas al azar y se queda a la espera de que el usuario ingrese el número del grupo en que está la letra que está mirando.

Imagina que estás interesado en la letra **A**, que está al final del grupo 3, así que ingresamos 3 y presionamos ENTER. El programa juntará las letras nuevamente siguiendo el método explicado anteriormente y las volverá a mostrar al usuario:

```
En que grupo esta tu tarjeta [1,2,3]: 3
Secuencia 2 :

    E   Q   I
    O   P   G
    C   U   K
    N   F   B
    L   A   J
    D   M   S
    R   T   H

En que grupo esta tu tarjeta [1,2,3]:
```

Si prestas atención, podrás ver cómo se han repartido de forma intercalada las letras respecto al arreglo anterior, habiendo ubicado la tercera columna en el segundo lugar y la segunda en el tercero.

Nuevamente se pide que el usuario indique la columna en que está su letra, que en este caso es la número 2. Ingresamos el 2 y presionamos ENTER:

```
En que grupo esta tu tarjeta [1,2,3]: 2
Secuencia 3 :

    E   O   C
    N   L   D
    R   Q   P
    U   F   A
    M   T   I
    G   K   B
    J   S   H

En que grupo esta tu tarjeta [1,2,3]:
```

El programa realiza nuevamente el procedimiento que hemos explicado anteriormente y vuelve a mostrar nuevamente las letras al usuario.

Por última vez se debe ingresar el número de columna en que está la letra elegida. Como ves, la letra **A** está en la columna 3, así que ingresamos el número 3 y presionamos ENTER.

```
En que grupo esta tu tarjeta [1,2,3]: 3
Obviamente elegiste la A
```

El programa le indicará al usuario cuál es la letra que había elegido.

NOTA: El programa debe controlar que se ingresen opciones válidas y pedir las hasta que así sea.

IMPLEMENTACIÓN

Para llevar adelante este proyecto deberás utilizar las siguientes constantes y variables ya definidas:

```
//Constantes para estandarizar los valores del programa
final short MAX_TARJETAS_GRUPO= 7; //Tarjetas por grupo
final short MAX_GRUPOS= 3; //Cantidad de grupos
final short MAX_TARJETAS= MAX_TARJETAS_GRUPO*MAX_GRUPOS; //Total de tarjetas.
final short MIN_TARJETA_VALOR= 'A'; //Tarjeta inicial, en este caso letra A.
final short MAX_TARJETA_VALOR= (char)(MAX_TARJETAS+(int)('A')-1); //Tarjeta final.

//Variables que deben usarse para resolver el problema.
char[] grupo1= new char[MAX_TARJETAS_GRUPO],
      grupo2= new char[MAX_TARJETAS_GRUPO],
      grupo3= new char[MAX_TARJETAS_GRUPO];
char[] deck= new char[MAX_TARJETAS];
```

Las variables **grupo1**, **grupo2** y **grupo3** son simples arreglos de 7 celdas (0 a 6) de tipo **char**.

La variable **deck** es un arreglo de 21 celdas (de 0 a 20) de tipo **char**.

Las **constantes** definidas son las siguientes:

- **MAX_TARJETAS_GRUPO**: Define la cantidad de tarjetas por grupo de cartas. En este caso son 7 cartas por cada grupo.
- **MAX_GRUPOS**: Define la cantidad de grupos a utilizar, en este caso 3.
- **MAX_TARJETAS**: Define la cantidad de tarjetas totales en el mazo, en este caso 21.
- **MIN_TARJETA_VALOR**: Define la primera letra disponible, obviamente la **A**.
- **MAX_TARJETA_VALOR**: Define la última letra disponible, en este caso la **U**.

GENERACIÓN DE LETRAS AL AZAR

La función `nextInt(n)` de la clase `Random` sirve para obtener números aleatorios entre `0` y un `n-1` tal como ya hemos visto a lo largo del curso, pero no sirve para generar caracteres al azar. Sin embargo, sabiendo que los caracteres están ordenados en la tabla ASCII según un valor numérico (su índice), es posible generar un número al azar que coincida con la letra que queremos obtener y luego utilizar el casteo (`char`) para obtener la letra en cuestión.

En concreto nuestras letras van desde la **A** hasta la **U**, es decir desde el índice **65** de la tabla hasta el índice **85** de la tabla. La gracia está entonces en poder generar números al azar entre 65 y 85 inclusive. Esto ya lo hemos visto en clases anteriores. En concreto en este caso sería:

```
numero= generador.nextInt(85-65+1)+65;
```

En este ejemplo, `generador` es un objeto (variable) de tipo `Random`, tal como ya se ha hecho en clases previas a este proyecto, con la cual se utiliza la función `nextInt` para obtener un número aleatorio. De forma genérica, si tenemos un número X y otro número Y que es mayor, y queremos generar números aleatorios entre X e Y, se debería utilizar `nextInt` de la siguiente manera:

```
numero= generador.nextInt(Y-X+1)+X;
```

Ahora bien, tal como hemos enseñado en el curso, es aconsejable utilizar constantes para nuestros programas, y como ya hemos definido 5 en este proyecto, las usaremos para esta tarea. Por tanto, la fórmula final queda así:

```
indice= generador.nextInt((int)MAX_TARJETA_VALOR-(int)MIN_TARJETA_VALOR)+1+(int)(MIN_TARJETA_VALOR);
```

De este modo ya tienes la fórmula que te da como resultado un índice ASCII entre 65 y 85 pero utilizando las constantes definidas. Así, al cambiar algún valor de estas constantes el programa igualmente funcionará de manera correcta.

Lo que te queda por hacer es simplemente obtener el carácter del índice generado utilizando (`char`).

ALGORITMO PARA TRATAMIENTO DE LAS CARTAS

Un algoritmo no es otra cosa que una serie de pasos o instrucciones a seguir para hacer algo o resolver un problema. Una receta de cocina, por ejemplo, es un algoritmo. La forma aburrida de definirlo, según Wikipedia, es:

En matemáticas, lógica, ciencias de la computación y disciplinas relacionadas, un es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permiten llevar a cabo una actividad mediante pasos sucesivos que no generen dudas a quien deba hacer dicha actividad.

Cada secuencia de instrucciones que has programado y que has visto en los videos es un algoritmo. Dado que una computadora solo puede seguir instrucciones precisas paso por paso para hacer algo, un programa no es otra cosa que muchos algoritmos que sirven para resolver un problema.

El algoritmo que sugerimos para resolver el problema de las cartas en este programa es el que describiremos a continuación. Tú como estudiante puedes utilizar este algoritmo o bien crear uno propio, sin embargo esto último no es obligatorio, y por eso es que dejamos una guía aquí.

Los pasos a seguir para la ejecución del programa serían los siguientes:

1. **Generar letras al azar sin repetir en deck:** Dedicamos una clase completa a mostrar cómo generar números al azar sin repetir en un arreglo. Esto es exactamente lo mismo.
2. **Asignar a los 3 grupos sus 7 cartas:** Esto es simple, se recorre el arreglo **deck** hasta el final. Las primeras 7 celdas se copian a **grupo1**, las otras 7 a **grupo2** y las últimas 7 a **grupo3**.
3. **Se entra en un bucle repetitivo:**
 - I. Se muestran los 3 grupos en pantalla intercalando las letras de modo que cada grupo aparezca en una columna. Esto se logra simplemente imprimiendo los tres arreglos a la vez, primero la celda de **grupo1**, luego la de **grupo2** y finalmente la de **grupo3**.
 - II. Se entra en un bucle para pedir el número de grupo.
 - i. Si la opción ingresada no es correcta (**1, 2 o 3**) se repite este bucle.
 - III. Según la opción ingresada se vuelven a poner los grupos en **deck** manteniendo el orden de sus letras, de forma que el grupo en que está la letra elegida quede en medio. Primero se pone un grupo de la celda 1 a la 7, luego el otro de la celda 8 a la 14 (en este está la letra elegida) y finalmente el último grupo de la celda 15 a la 21.
 - IV. Se vuelven a mover las letras a los grupos de forma intercalada.
 - V. Se repite el bucle desde el punto **3** hasta agotar la cantidad de repeticiones (3).
4. Se muestra al usuario la letra seleccionada, que es la celda central del arreglo **deck**.

RESTRICCIONES DE CÓDIGO

Para este proyecto tienes que utilizar las constantes y variables definidas arriba tal cual se te han indicado (básicamente copia y pega). Además tienes las siguientes restricciones:

- No puedes declararte ningún otro tipo de datos, clase ni utilizar librerías de Java que no se hayan visto en el curso.
- No puedes declarar nuevas constantes.
- Debes utilizar las constantes ya definidas para tus bloques **FOR**, **WHILE** o **REPEAT** que utilices para recorrer arreglos.
- Todo tu código debe estar dentro del procedimiento **main** de tu clase principal. No puedes declarar nada fuera del procedimiento **main**.

Puedes declararte tantas variables como quieras y utilizar el algoritmo que quieras. Los pasos dados en la sección anterior simplemente son una sugerencia para ayudarte, pero no es algo a lo que te tengas que atener de forma restrictiva.

Cualquier otra facilidad de Java no será admitida en este proyecto.

SE PIDE

Escribir un programa Java que cumpla con todo lo descrito en el documento y funcione exactamente igual que los ejemplos. Se te proveerá del archivo **.jar** ya compilados para que puedas probarlos en funcionamiento y ver lo que tienes que lograr.

ENTREGA DEL PROYECTO

Este proyecto es obligatorio y debe ser entregado ya que será evaluado por un docente. Lo que debes entregar es el archivo con el código fuente. Dependiendo de la plataforma en que hagas el curso será el medio por el cual enviarás el archivo:

- **KA EduSoft:** En la lección donde se te presenta este proyecto tendrás la opción para subir el archivo del código fuente.
- **Udemy:** Escribe un correo electrónico a bedelia@kaedusoft.edu.uy con el asunto **PROYECTO JAVA TRUCO21 [NOMBRE] [APELLIDO]** en el cual adjuntarás el archivo con el código fuente de tu trabajo.

Si tienes alguna duda escribe a bedelia@kaedusoft.edu.uy.