

**Instituto Nacional de Aprendizaje**

**Curso**

**Programación de Aplicaciones Informáticas.**

**Módulo**

**Lógica computacional**

**Actividad**

**Practica N 8**

**Integrantes**

**Mariel Daniela Rojas Sanchez, Daniel Andrés Vargas Alvarado, Manfred Gerardo Barrantes Torres.**

**Profesor**

**Oscar Pacheco Vásquez**

**2022**

**PLANTILLA PARA LA SOLUCIÓN DE PROBLEMAS**

**Propuesta de***:* Mariel Daniela Rojas Sanchez, Daniel Andrés Vargas Alvarado, Manfred Gerardo Barrantes Torres.

|  |
| --- |
| **Planteamiento del Problema** |
| Realizar un algoritmo que calcule con exactitud toda la suma de los resultados de los múltiplos de 3 o 5 por debajo de 1000 e imprimir los resultados. |

|  |  |
| --- | --- |
| **Identificación del problema** | |
| ***Datos de entrada*** | ***Datos de Salida*** |
| 1. Consulta al usuario, si desea continuar. | 1. Números múltiplos de 5 y 3 y la suma de estos. |
| **Observaciones *(Limitaciones / Excepciones / Suposiciones)*** | |
| Observaciones   1. Se le consulta al usuario si desea continuar, esto con la finalidad de tener interacción con el mismo; Esto no es solicitado, se agregó para interacción.   Limitaciones:   1. No se podrá realizar la operación con números mayores a 999. 2. No se podrá realizar restas y divisiones. 3. No se harán sumas de otros múltiplos que no sean 3 o 5. 4. Si un número es múltiplo de 5 y 3 solo se debe escribir más de una vez. | |
|  | |
| **Planteamiento de alternativas** | |
| ***Alternativa # 1*** | ***Alternativa #2*** |
| 1. Inicio 2. Imprimirle al usuario cual es la función del programa. 3. Consultarle al usuario si desea continuar presione la letra “T/t” con una bandera que empieza en falso y con un “mientras”. 4. Con un ciclo “Para” señalizar un contador hasta 999 y dentro del mismo ciclo crear un ciclo “Si” para determinar si los números múltiplos de 5 y 3. 5. Impresión en pantalla de los múltiplos de 3 y 5. 6. Con un ciclo “Para” recorrer los números múltiplos y sumarlos. 7. Imprimir “El total de todos los números es”. 8. Fin. | 1. Inicio 2. Definir nuestras variables. 3. Inicializar algunas variables con 0. 4. Imprimir “Suma de los múltiplos de 3 y 5” 5. Imprimir en pantalla “Los múltiplos de 3 son” con un ciclo “Para” para recorrer los números múltiplos de 3. 6. Se muestran los resultados obtenidos de la suma de todos los números de múltiplos de 3. 7. Imprimir “Los múltiplos de 5 son” con un ciclo “Para” para recorrer los números múltiplos de 5. 8. Se muestran en pantalla la suma de los múltiplos del 5. 9. Fin |

|  |  |
| --- | --- |
| **Elección de Alternativa** | |
|  | |
| ***Alternativa Seleccionada*** | ***Justificación*** |
| Alternativa #2 | Se seleccionó la número 2 ya que este algoritmo es más práctico de realizar, es más preciso y contundente con lo requerido, además de explicar con coherencia los pasos a seguir. |

**PLANTILLA PARA LA SOLUCIÓN DE PROBLEMAS**

**Propuesta de***:* Mariel Daniela Rojas Sanchez, Daniel Andrés Vargas Alvarado, Manfred Gerardo Barrantes Torres.

|  |
| --- |
| **Planteamiento del Problema** |
| Crear un algoritmo que calcule la diferencia entre la suma de los cuadrados de los primeros cien números naturales. |

|  |  |
| --- | --- |
| **Identificación del problema** | |
| ***Datos de entrada*** | ***Datos de Salida*** |
| 1. No se solicita nada. | 1. Imprime la suma de los cuadrados. 2. Impresión cuadrado de la suma. 3. Impresión la diferencia entre la suma de los cuadrados y el cuadrado de la suma. |
| **Observaciones *(Limitaciones / Excepciones / Suposiciones)*** | |
| Observaciones.   1. Se realizan sumas y elevaciones al cuadrado   Limitaciones:   1. No se realizara otra operación que no se elevación al cuadrado y suma. | |

|  |  |
| --- | --- |
| **Planteamiento de alternativas** | |
| ***Alternativa # 1*** | ***Alternativa #2*** |
| 1. Inicio 2. Se definen las variables que vayamos a ocupar. 3. Señalizamos una variable con 0, con un ciclo “Para” suma1 = suma1 + (i \* i) 4. Imprimir la suma del cuadrado 5. Señalizamos una variable con 0, con un ciclo “Para” suma2 = suma2 + i 6. Imprimir la suma del cuadrado elevado a la potencia y la diferencia de los cuadrados suma1 ^ 2 - suma2. | 1. Inicio 2. Definir nuestras variables 3. Inicializar las variables a usar con 0. 4. Para la suma de los cuadrados; 5. Utilizar un ciclo “Para” para sumar de los cuadrados, he imprimir el valor de este, hacer lo mismo sumar a i + 1. 6. Para el cuadrado de la suma; 7. Realizar una suma de los 100 primeros números con la función para. 8. Imprimir en pantalla el resultado de este. 9. Fin. |

|  |  |
| --- | --- |
| **Elección de Alternativa** | |
| ***Alternativa Seleccionada*** | ***Justificación*** |
| Alternativa #1 | Se optó por la alternativa 1 ya que los pasos son sumamente congruentes y fáciles de interpretar para realizar el pseudocódigo, asimismo define paso a paso las funciones a utilizar y sus respectivas especificaciones. |

**PLANTILLA PARA LA SOLUCIÓN DE PROBLEMAS**

**Propuesta de***:* Mariel Daniela Rojas Sanchez, Daniel Andrés Vargas Alvarado, Manfred Gerardo Barrantes Torres.

|  |
| --- |
| **Planteamiento del Problema** |
| Realizar un pseudocódigo que solicite al usuario 8 números para que después se ingresen a arreglo unidimensional y luego imprimir dichos números de forma inversa a lo ingresado. |

|  |  |
| --- | --- |
| **Identificación del problema** | |
| ***Datos de entrada*** | ***Datos de Salida*** |
| 1. Solicitar 8 números. | 1. Impresión de un vector con los números 8 ingresados. 2. Impresión del vector inverso. |
| **Observaciones *(Limitaciones / Excepciones / Suposiciones)*** | |
| 1. No se mostraran otros números que no haya ingresado el usuario. | |

|  |  |
| --- | --- |
| **Planteamiento de alternativas** | |
| ***Alternativa # 1*** | ***Alternativa #2*** |
| 1. Inicio. 2. Definir variables. 3. Dimensionamos nuestro vector con 8 filas con un “Repetir”; recorremos que el vector para almacenar los números ingresados, definimos una variable “final = 8” para que termine el “Repetir”. 4. Imprimimos el vector con un “Repetir” escribir el vector de “n”.   Imprimir el vector de forma inversa escribimos el vector = n -1.   1. Fin. | 1. Inicio. 2. Dimensionar un vector de 8 campos. 3. Solicitar al usuario que ingrese 8 números. 4. Almacenar los datos solicitados en forma ingresada, utilizando la función para con la intención de moverse por el vector. 5. Se muestra los números ingresados esto con la función para, de esta manera mostramos los números guardados en el vector mientras nos movemos en forma ordenada dentro del vector. 6. Con la función repetir de manera inversa mostraremos en pantalla los números antes mostrados pero en esta ocasión con en forma contraria. 7. Fin. |

|  |  |
| --- | --- |
| **Elección de Alternativa** | |
| ***Alternativa Seleccionada*** | ***Justificación*** |
| Alternativa # 1 | Ambas alternativas son realizables, pero la alternativa 1 marca mejor el paso a paso para realizarlo en seudocódigo, por ende la solución es más entendible. |

**PLANTILLA PARA LA SOLUCIÓN DE PROBLEMAS**

**Propuesta de***:* Mariel Daniela Rojas Sanchez, Daniel Andrés Vargas Alvarado, Manfred Gerardo Barrantes Torres.

|  |
| --- |
| **Planteamiento del Problema** |
| Efectuar un algoritmo de sistema decimal que determinar y permita mostrar su composición de números entre 0 y 9999. |

|  |  |
| --- | --- |
| **Identificación del problema** | |
| ***Datos de entrada*** | ***Datos de Salida*** |
| 1. Números entre el 0 y 9999. | 1. Composición del número |
| **Observaciones *(Limitaciones / Excepciones / Suposiciones)*** | |
| Observaciones:   1. Acá se utilizan decimales en algunas o la mayoría de cifras.   Limitaciones:   1. Solo se mostrará los números entre en 0 y 9999. | |

|  |  |
| --- | --- |
| **Planteamiento de alternativas** | |
| ***Alternativa # 1*** | ***Alternativa #2*** |
| 1. Inicio. 2. Definir variables. 3. Pedir al usuario que ingrese un número menor a 9999. 4. Con “Si” determinar cada una de las unidades, centenas, decenas, unidades de millar e imprimir cada una de ellas. | 1. Inicio. 2. Se definen las variables. 3. Se solicita al usuario que ingrese un número menor a 9999. 4. Se ingresa en la variable determinada. 5. Se realiza la descomposición al número ingresado y se guardan sus decimales, esto para guardarlo en la casilla correspondiente. 6. Se evalúa que el número y según se ordena, si esta unidades se mutra el número y en que casilla esta y así sucesivamente. 7. Se imprime el resultado en la pantalla. 8. Fin. |

|  |  |
| --- | --- |
| **Elección de Alternativa** | |
| ***Alternativa Seleccionada*** | ***Justificación*** |
| Alternativa # 2 | Se seleccionó la alternativa 2 ya que esta es de mayor comprensión y esclarece paso a paso lo solicitado. |

**PLANTILLA PARA LA SOLUCIÓN DE PROBLEMAS**

**Propuesta de***:* Mariel Daniela Rojas Sanchez, Daniel Andrés Vargas Alvarado, Manfred Gerardo Barrantes Torres.

|  |
| --- |
| **Planteamiento del Problema** |
| Crear un algoritmo que reciba una fecha y hora y calcule la fecha y la hora resultante al sumar las horas ingresadas por el usuario. |

|  |  |
| --- | --- |
| **Identificación del problema** | |
| ***Datos de entrada*** | ***Datos de Salida*** |
| 1. Fecha y hora ingresada por el usuario. 2. Horas para calcular la siguiente fecha. | 1. Fecha y hora. 2. Fecha y hora nuevas. |
| **Observaciones *(Limitaciones / Excepciones / Suposiciones)*** | |
| 1. Se utilizará la hora militar. 2. Limitaciones: El año debe ser mayor a 1000 y menor a 2050. 3. No se podrá realizar para fechas hacia atrás. | |

|  |  |
| --- | --- |
| **Planteamiento de alternativas** | |
| ***Alternativa # 1*** | ***Alternativa #2*** |
| 1. Inicio 2. Definir nuestras variables, diaMax, mesMax, añoMax, añoBic, horasMax, segundosMax, minutosMax, dia0, mes0, año0, hora0. 3. Imprimir al usuario que ingrese el año, 1000 al 2050, ingresar el mes del 1 al 12, con un “Repetir”. 4. Con un “Si” determinar si es bisiesto, ingresar el día 1 al 29. 5. Ingresar las horas para el siguiente día. 6. Con un “Si” calcular la hora, el día y el año. 7. Imprimir resultado. 8. Fin. | 1. Inicio. 2. Definir las variables. 3. Solicitar al usuario que ingrese año, mes, día y hora en formato de 24 horas. 4. Se guarda en la variable. 5. Se muestra lo guardado. 6. Con un “Si” para determinar si es bisiesto. 7. Obtención de cuantos días tiene el mes, ya que, si es el 4,6,9 y 11 mes estos contaran con 30 días de lo contrario con 31. 8. Solicitar las horas a sumar. 9. Con un “Si” calcular la hora, el día y el año. 10. Determinar si la fecha cambia ya sea día o mes. 11. Si año es trece a año le sumaremos 1 y a mes también esto con la finalidad de realizar el cambio de año y también mes. 12. Mostrar el resultado con fecha y hora correspondiente. 13. Fin. |

|  |  |
| --- | --- |
| **Elección de Alternativa** | |
| ***Alternativa Seleccionada*** | ***Justificación*** |
| Alternativa # 2 | Se seleccionó la alternativa número 2 ya que es de más fácil de comprensión esto llevándonos a realizar el seudocódigo más eficiente y rápido |

**PLANTILLA PARA LA SOLUCIÓN DE PROBLEMAS**

**Propuesta de***:* Mariel Daniela Rojas Sanchez, Daniel Andrés Vargas Alvarado, Manfred Gerardo Barrantes Torres.

|  |
| --- |
| **Planteamiento del Problema** |
| Realizar un algoritmo que generé números al azar y que de estos números almacene en una matriz de 6 x 5 los 30 primeros números primos. |

|  |  |
| --- | --- |
| **Identificación del problema** | |
| ***Datos de entrada*** | ***Datos de Salida*** |
| 1. No se solicita nada. | 1. Impresión de la matriz 6x5 con los 30 primeros números primos. |
| **Observaciones *(Limitaciones / Excepciones / Suposiciones)*** | |
| Limitaciones:   1. No almacenara más de 30 números primos. 2. No procesara números pares. | |

|  |  |
| --- | --- |
| **Planteamiento de alternativas** | |
| ***Alternativa # 1*** | ***Alternativa #2*** |
| 1. Inicio. 2. Se definen las variables. 3. Se usa un ciclo de 1 al 115 de donde se obtendrán los 30 números primos. 4. Se inicializa a num. 5. Si el residuo de a=0 entonces el contador aumenta una unidad y así sucesivamente hasta que sea igual a 115. 6. Se guardan en una matris de 6 x 5. 7. Se muestra la matriz con los resultados. 8. Fin. | 1. Inicio. 2. Se definen variables. 3. Dimensionar una matriz de 6x5. 4. Inicializar contadores desde 0. 5. Utilizar un siclo de 1 hasta 115 ya que acá se contienen los 30 números primos. 6. Se recorre la matriz con repetir y se verifica los números primos. 7. Con la opción mod nos daremos cuenta que el resido da como 0 este se pasa a guardar y se +1 para ir incrementando la variable. 8. Se verifican los números primos y nos movemos en la matriz con un “para” para irlos guardando. 9. Se muestra en pantalla los números guardados; esto los hacemos moviéndonos en la matriz y para lograrlo nos movemos por las filas y columnas con “para”. 10. Fin. |

|  |  |
| --- | --- |
| **Elección de Alternativa** | |
| ***Alternativa Seleccionada*** | ***Justificación*** |
| Alternativa #2 | Se selecciona la numero 2 ya que esta alternativa aunque sea más larga nos define pasa a paso lo necesario para hacer y ejecutar el algoritmo. |

**PLANTILLA PARA LA SOLUCIÓN DE PROBLEMAS**

**Propuesta de***:* Mariel Daniela Rojas Sanchez, Daniel Andrés Vargas Alvarado, Manfred Gerardo Barrantes Torres.

|  |
| --- |
| **Planteamiento del Problema** |
| Crear un algoritmo donde exista 52 números de acuerdo con la baraja de cartas (Naipes), mezclar los números, mostrar el primer número y de acuerdo con el número se eliminaran esa cantidad de cartas son sus respectivos número así sucesivamente hasta cuando el número de cartas restantes no es suficiente para desecharlas. |

|  |  |
| --- | --- |
| **Identificación del problema** | |
| ***Datos de entrada*** | ***Datos de Salida*** |
| 1. No se solicita nada. | 1. Cartas que fueron mostradas, desechadas y las sobrantes. |
| **Observaciones *(Limitaciones / Excepciones / Suposiciones)*** | |
| Observaciones:  Se selecciona la carta y el número de carta, ambas se cuentan y se restan, ejemplo “se selecciona una carta la cual es un 7, por este se resta la carta que se voltio y se suman a esta resta el número de cartas que son “7” lo cual me mostraría de resultado 8 cartas que hay que quitar de 52 cartas de naipe.”  Limitaciones:  No se podrá con más de 52 cartas | |

|  |  |
| --- | --- |
| **Planteamiento de alternativas** | |
| ***Alternativa # 1*** | ***Alternativa #2*** |
| 1. Inicio. 2. Se declaran las variables. 3. Se inicializan los contadores desde 0, estos serán los capases de llevar la cuanta de las cartas que se muestran y las desechadas. 4. Con la opción azar seleccionamos de las13 cartas 1 carta, la cual nos arrojara un número. 5. Guardamos el número en una variable. 6. Según el número que se guardó se realiza la resta de cartas, se toma en cuenta la carta que se seleccionó más el número que tiene dentro esa carta. 7. Luego a contador le sumamos lo que lleva contador +1 estos para saber la cantidad de cartas mostradas. 8. En el contador de cartas desechadas lo que se realiza es sumar todas las cartas que se han desechado durante el juego.   Se muestra en pantalla los resultados obtenidos.  Fin. | 1. Inicio 2. Se declaran variables. 3. Inicializar contadores en 0. 4. Con la opción azar seleccionamos de las13 cartas 1 carta, la cual nos arrojara un número. 5. Utilizamos un mientras para evaluar el numero seleccionado y de esta manera comparar si el número es igual a la opción se le envía dicha opción de lo contrario se busca la opción que sea igual. 6. Según la opción seleccionada se hace los siguientes: Se resta la carta y el número de carta, se le suma 1 a contador y las cartas desechadas se suman en el contador. 7. Se guardan es sus respectivas variables y contadores. 8. Según lo guardado se imprime en la consola los resultados obtenidos. 9. Fin. |

|  |  |
| --- | --- |
| **Elección de Alternativa** | |
| ***Alternativa Seleccionada*** | ***Justificación*** |
| Alternativa #1 | Se ha seleccionado la alternativa 2 ya que nos conduce al resultado con una respuesta más rápida, debido a que su estructura de procesamiento es menos compleja tanto de interpretar como de realizar. |

**PLANTILLA PARA LA SOLUCIÓN DE PROBLEMAS**

**Propuesta de***:* Mariel Daniela Rojas Sanchez, Daniel Andres Vargas Alvarado, Manfred Gerardo Barrantes Torres.

|  |
| --- |
| **Planteamiento del Problema** |
| Crear un algoritmo que se capaz de elaborar un cartón de bingo y pueda mostrase en pantalla dicho cartón. |

|  |  |
| --- | --- |
| **Identificación del problema** | |
| ***Datos de entrada*** | ***Datos de Salida*** |
| 1. No hay datos de entrada. | 1. Cartón completo de números para jugar bingo. |
| **Observaciones *(Limitaciones / Excepciones / Suposiciones)*** | |
| 1. Observaciones.   Se tendrá crea una matriz de (6x5) para poder colocar los números en este, no solo los números si no también la palabra “Bingo”.  No se pude repetir los números, además los números tendrán que estar impresos con 2 cifras. | |

|  |  |
| --- | --- |
| **Planteamiento de alternativas** | |
| ***Alternativa # 1*** | ***Alternativa #2*** |
| 1. Inicio 2. Definimos nuestras variables, fila, columna, i, numero, j, bingo, l Como Entero. 3. Dimensionar para la palabra bingo [5,5] y se dimensiona un vector de 25 espacios. 4. Inicializamos i = 0 5. Dentro de un “Repetir” hasta que i = 5, utilizaremos un ciclo “Para”, para que la estructura condicional “Si” determine la posición de cada uno. 6. Con un ciclo “Para” y una estructura condicional “Si” colocamos los números sin repetir. 7. Con una estructura condicional colocar a los números que solo tengan una cifra colocarle un 0. 8. Asignamos a cada posición una letra para imprimir la palabra Bingo. 9. Imprimimos el cartón para el bingo con dos ciclos “Para”. 10. Fin. | 1. Inicio 2. Definimos nuestras variables, fila, columna, i, numero, j, bingo, l Como Entero. 3. Dimensionar para la palabra bingo [6,5] y dimensionamos vector con 25 espacios. 4. Inicializamos i = 0 5. Dentro de un “Repetir” hasta que i = 5, utilizaremos un ciclo “Para”, para que la estructura condicional “Si” determine la posición de cada uno. 6. Con un ciclo “Para” y una estructura condicional “Si” colocamos los números sin repetir. 7. Con una estructura condicional colocar a los números que solo tengan una cifra colocarle un 0. 8. Verificar con el recorrido de l vector que los números no se repitan. 9. Imprime la palabra BINGO antes de la matriz. 10. Imprimimos el cartón para el bingo con dos ciclos “Para”. 11. Fin. |

|  |  |
| --- | --- |
| **Elección de Alternativa** | |
| ***Alternativa Seleccionada*** | ***Justificación*** |
| Alternativa #2 | 1. Se optó por esta opción ya es las eficiente y más rápida en desarrollar en seudocódigo, además nos deja entender los pasos que se deben realizar. |

**PLANTILLA PARA LA SOLUCIÓN DE PROBLEMAS**

**Propuesta de***:* Mariel Daniela Rojas Sanchez, Daniel Andrés Vargas Alvarado, Manfred Gerardo Barrantes Torres.

|  |
| --- |
| **Planteamiento del Problema** |
| Crear dos matrices con valores aleatorios de 1 a 99, la primera matriz se ordena e izquierda a derecha y de arriba hacia abajo y la segunda matriz ordenada de abajo hacia arriba y de derecha a izquierda. |

|  |  |
| --- | --- |
| **Identificación del problema** | |
| ***Datos de entrada*** | ***Datos de Salida*** |
| 1. Opción para ver la matriz 1, la matriz 2 o para salir del menú. | 1. Menú que muestra las dos matrices ordenadas. |
| **Observaciones *(Limitaciones / Excepciones / Suposiciones)*** | |
| El usuario únicamente podrá salir del programa cuando digite la opción 3 del menú. | |

|  |  |
| --- | --- |
| **Planteamiento de alternativas** | |
| ***Alternativa # 1*** | ***Alternativa #2*** |
| 1. Inicio 2. Se declaran las variables necesarias. 3. Se dimensiona la matriz 6x6. 4. Se llena la matriz con números al azar de 1 al 99. 5. Se crea un vector de 36 espacios que es el que va a ir ordenando los elementos de la matriz. 6. En un repetir donde según lo que el usuario desea visualizar, se va a imprimir en pantalla, si elige la opción 1 el usuario podrá ver la matriz ordenada de derecha a izquierda y de arriba abajo, esta matriz se ordenó por el método de burbuja, donde un vector recorre la matriz para ordenarla y luego esta toma los valores del vector de forma ordenada, si el usuario digita la opción 2 sucederá lo mismo que en la opción 1 pero en este caso se va a imprimir la matriz con los valores aleatorios de 1 a 99 pero en este caso se observaran ordenados de derecha a izquierda y de arriba abajo. 7. La opción 3 es para finalizar el programa, donde imprime un mensaje indicando que ha salido del programa. 8. Fin. | 1. Inicio 2. Se declaran las variables necesarias. 3. Se dimensiona la matriz 6x6. 4. Se llena la matriz con números al azar de 1 al 99. 5. Se crea un vector de 36 espacios que es el que va a ir ordenando los elementos de la matriz. 6. Se va a imprimir directamente en pantalla, una matriz ordenada de derecha a izquierda y de arriba abajo, esta matriz se ordenó por el método de burbuja, donde un vector recorre la matriz para ordenarla y luego esta toma los valores del vector de forma ordenada y también se imprime seguidamente la misma matriz, pero en este caso se va a imprimir la matriz con los valores iguales a la primera matriz, pero en este caso se observarán ordenados de derecha a izquierda y de arriba abajo. 7. Fin. |

|  |  |
| --- | --- |
| **Elección de Alternativa** | |
| ***Alternativa Seleccionada*** | ***Justificación*** |
| Alternativa #1 | La alternativa 1 es mucho más eficaz y existe más interacción con el usuario ya que esta la opción de un menú donde el usuario puede elegir de qué manera quiere ver ordenada la matriz y si desea salir, por lo que el algoritmo no se termina hasta que el usuario así lo requiera. |

**PLANTILLA PARA LA SOLUCIÓN DE PROBLEMAS**

**Propuesta de***:* Mariel Daniela Rojas Sanchez, Daniel Andrés Vargas Alvarado, Manfred Gerardo Barrantes Torres.

|  |
| --- |
| **Planteamiento del Problema** |
| Diseñar un algoritmo para ayudarle al Mago Paco a realizar un truco de “mentalismo” con cartas de naipe, donde el mago Paco tiene que adivinar 1 carta únicamente guiándose con los valores de otras 4 cartas mostradas. |

|  |  |
| --- | --- |
| **Identificación del problema** | |
| ***Datos de entrada*** | ***Datos de Salida*** |
| 1. Valor de la quinta carta que se debe adivinar. | 1. Mensaje con los valores de las 4 cartas seleccionadas al azar y ordenadas. 2. Mensaje indicando si adivino el numero o no adivino. |
| **Observaciones *(Limitaciones / Excepciones / Suposiciones)*** | |
| 1. Si hay números repetidos, el palo de la carta determinara cuál es mayor: en este caso, el orden sería Tréboles (♣) Diamantes (♦) Corazones (♥) Picas (♠). (Si aparecen dos 9 entonces el palo determina cuál de los dos es mayor). 2. La primera carta mostrada, indica el palo de la carta por adivinar. 3. Las otras 3 cartas indican el numero de la carta por adivinar, dependiendo de como estas 3 cartas estén acomodadas. | |

|  |  |
| --- | --- |
| **Planteamiento de alternativas** | |
| ***Alternativa # 1*** | ***Alternativa #2*** |
| Inicio  Fin. | Inicio  Fin. |

|  |  |
| --- | --- |
| **Elección de Alternativa** | |
| ***Alternativa Seleccionada*** | ***Justificación*** |
| Alternativa #1 |  |