

# UNIVERSIDAD TECMILENIO

Mauricio Estrada De la Garza - AL02976904

Igor Sung Min Kim Juliao - AL02829189

14/09/2023

Programación Orientada a Objetos

Reto 4

## Objetivo de la actividad:

El alumno desarrollará un programa en el cual se implemente los conceptos:

1. Uso de Funciones.
2. Uso de Ciclos
3. El correcto uso y asignación de datos.
4. Creación y Uso de arreglos de dos dimensiones.
5. Manejo de Excepciones.

## Requerimientos para la actividad:

Uso Java jdk a través de un IDE, preferentemente Netbeans. ☕

## Instrucciones para el alumno:

En esta actividad, los alumnos crearán un programa en lenguaje Java para realizar las siguientes operaciones matriciales:

1. Suma de Matrices
2. Multiplicación de un entero por la matriz.
3. Impresión de Matriz

El programa debe contar con un menú en el cual se escoja cualquiera de las opciones posibles.

El programa debe preguntar por las dimensiones de la matriz (filas y columnas)

El programa debe introducir los datos a complementar en cada celda de la matriz

## Conceptos a calificar:

### 1. Definición de funciones

1. Función Menú.
2. Función Suma de Matrices
3. Suma de Multiplicación de Matrices.
4. Suma de Display de Matrices.

  Brownie points, por experimentar con diferentes tamaños de matrices.



1. **Uso de Excepciones:** El correcto uso e implementación de excepciones si un dato incorrecto es ingresado y/o un resultado es incorrecto.

Mensajes personalizados de acuerdo al tipo de excepción que se pretenda manejar.

1. **Uso de ciclos:** Continuidad de la ejecución del programa.
2. **Resultados:** Correcta
3. **Reflexión:** Discutir sobre cómo implementaron las matrices, como se realizaron las implementaron las funciones, entendimiento de excepciones y el cómo participan en la continuidad del programa.

## Entregables:

- Código Fuente, archivos con extensión .java (Recuerda, cualquier otro tipo de archivo no es permitido, i.e. .class, .zip, .jar, etc)
- Compilación sin errores.

- Cumplimiento de Especificaciones.
- 
- ||||| Reflexión individual por integrante del equipo. !!!!!

**Importante:** Solo se revisa la última entrega

\* Sin reporte y/o código fuente, ningún porcentaje es considerado ⇒ Calificación NE

**Restricción:** Equipos de no más de tres personas, no menos de dos. 😊

## CÓDIGO:

```

349 while (!opcionvalida)
350 {
351     try
352     {
353         linea();
354         System.out.print("INGRESA UNA OPCION VALIDA: ");
355         opcion = scanner.nextInt();
356         opcionvalida = true;
357     }
358     catch (InputMismatchException e)
359     {
360         espacio();
361         System.out.println("DEBES INGRESAR UN NUMERO");
362         scanner.nextLine();
363     }
364 }
365 return opcion;
366
367
368
369
370
371
372
373
374
375
376
377
378
379

```

reto4.Reto4 > main > try > while (opcion) > switch (opcion) > case 3:

Output - reto4 (run) x

```

FUN:
RETO 4 - POO

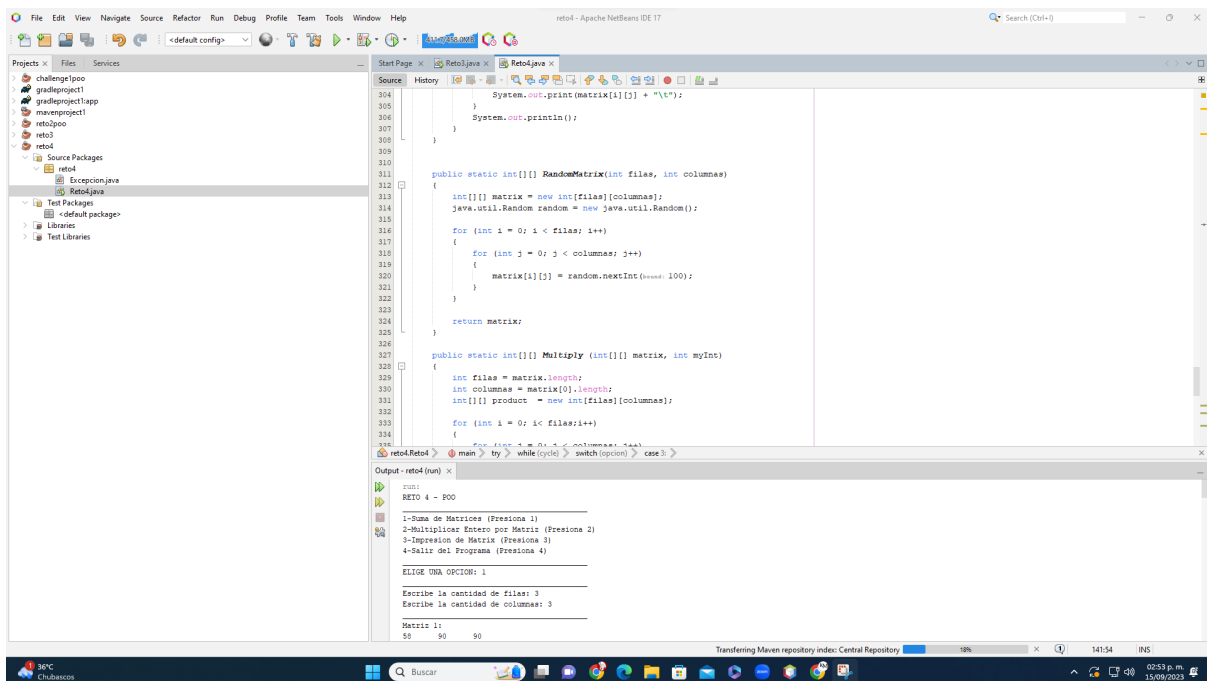
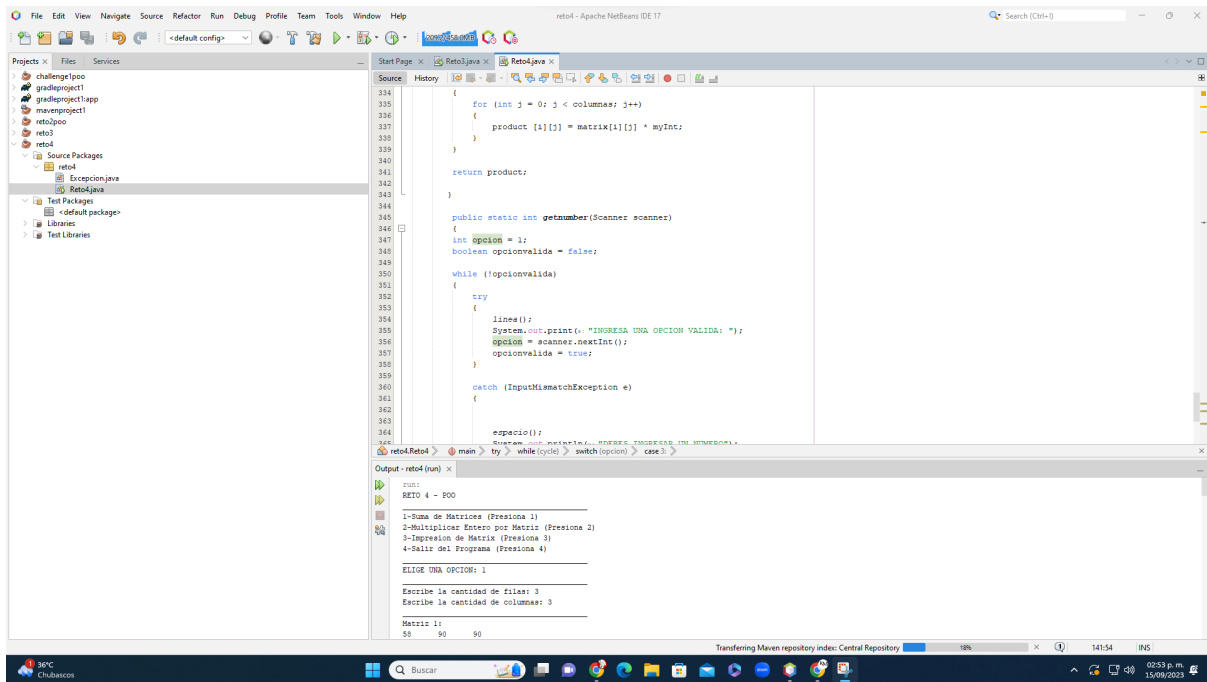
1-Suma de Matrices (Presiona 1)
2-Multiplicar Entero por Matriz (Presiona 2)
3-Imprimir de Matriz (Presiona 3)
4-Salir del Programa (Presiona 4)

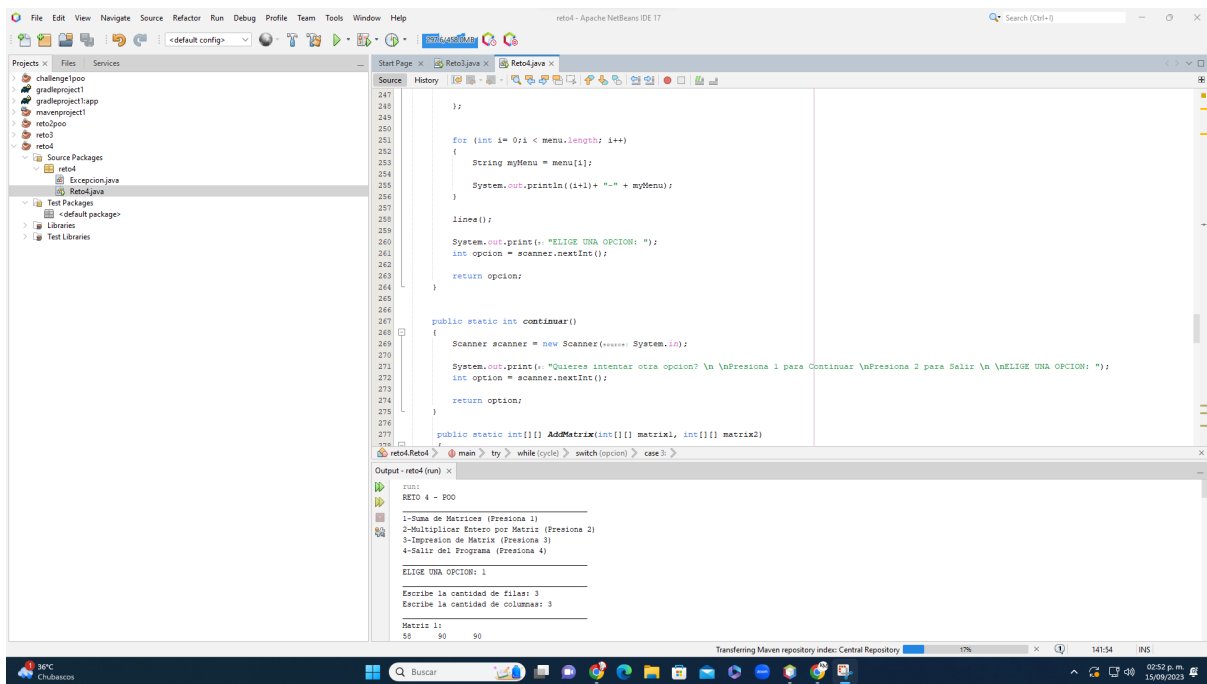
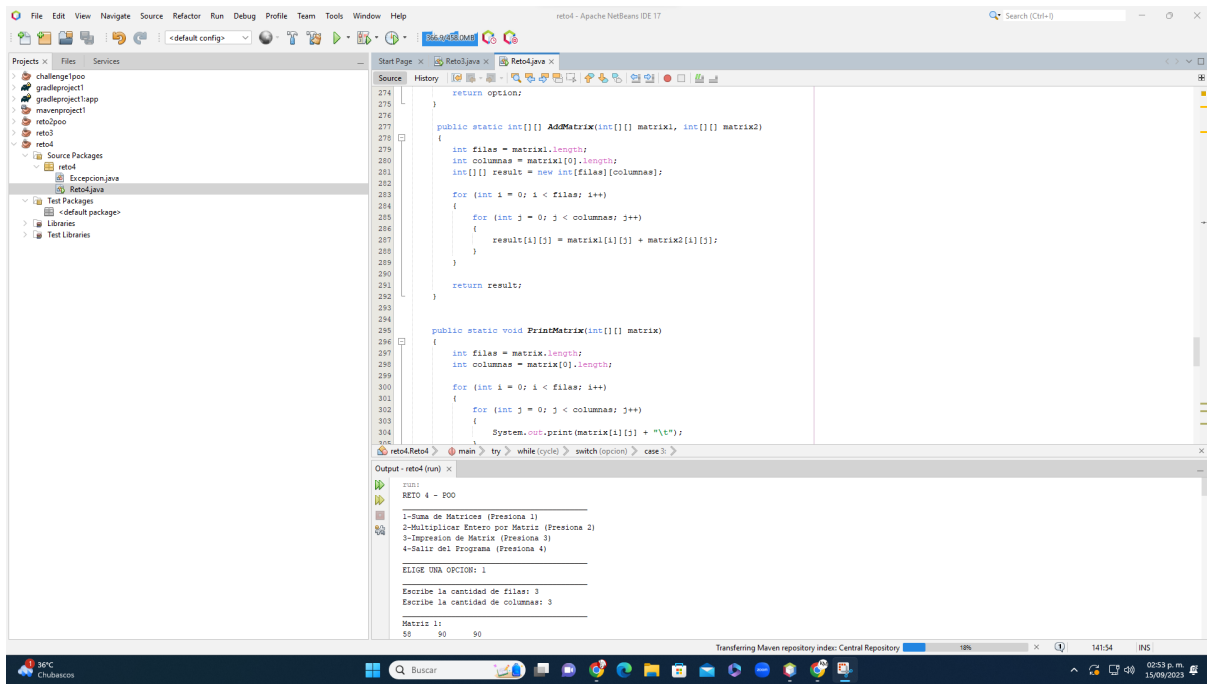
ELIGE UNA OPCION: 1

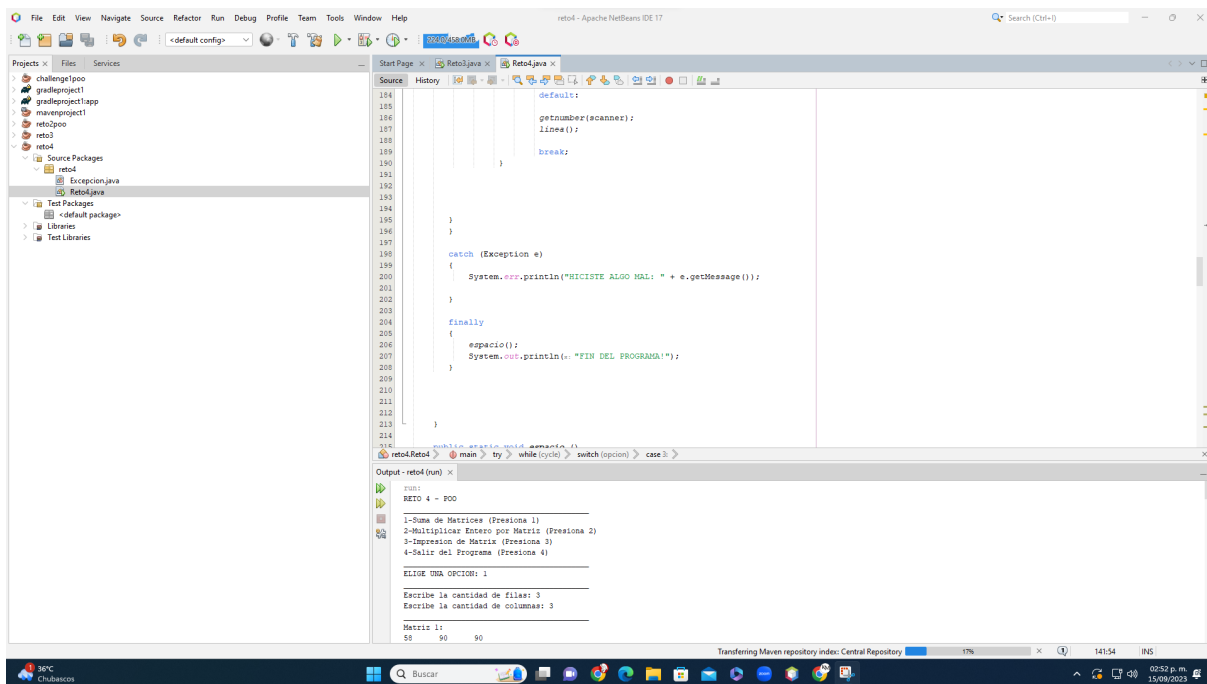
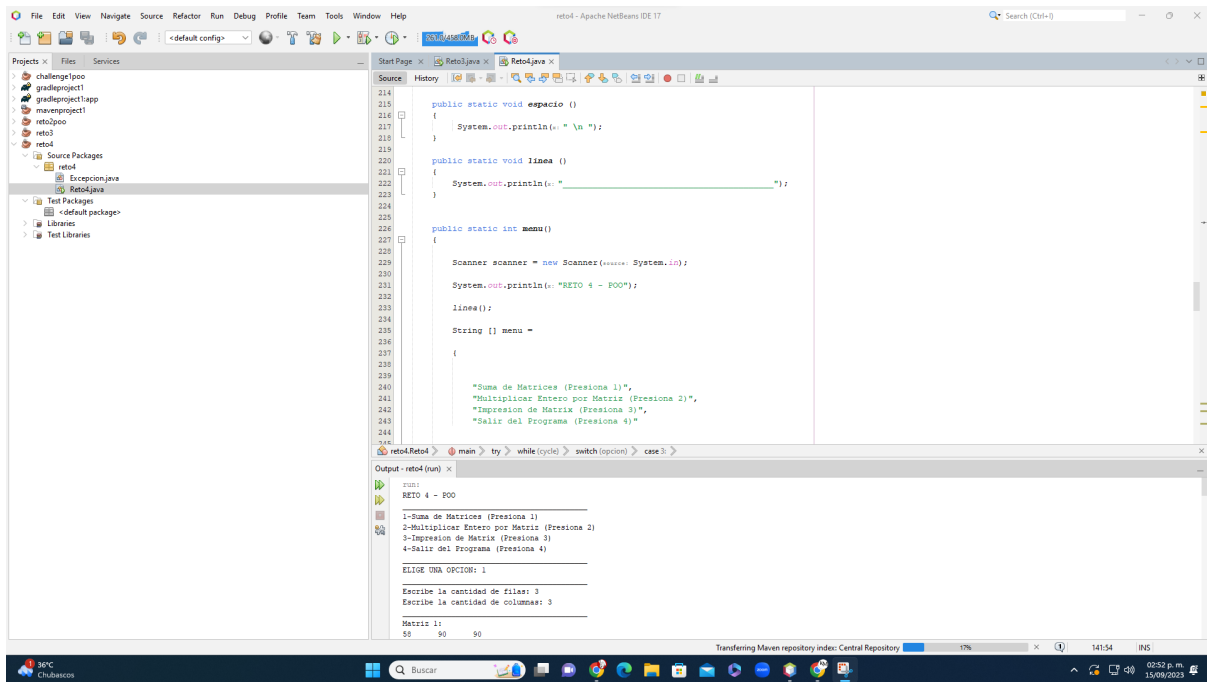
Escribe la cantidad de filas: 3
Escribe la cantidad de columnas: 3

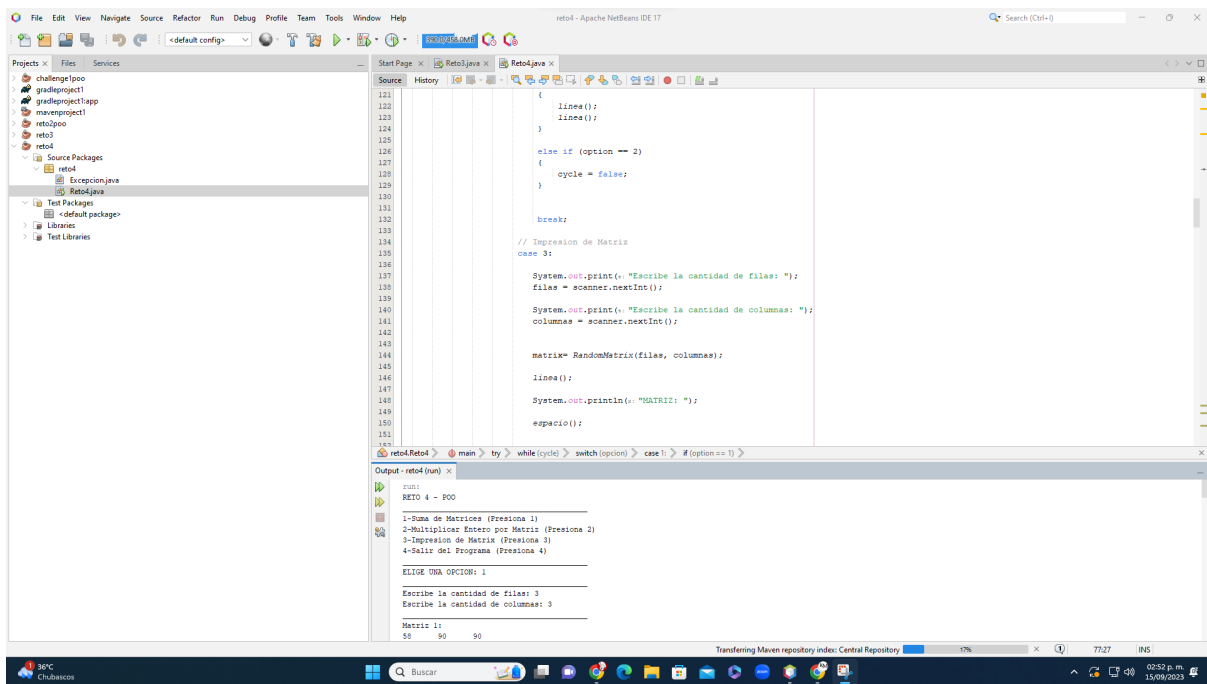
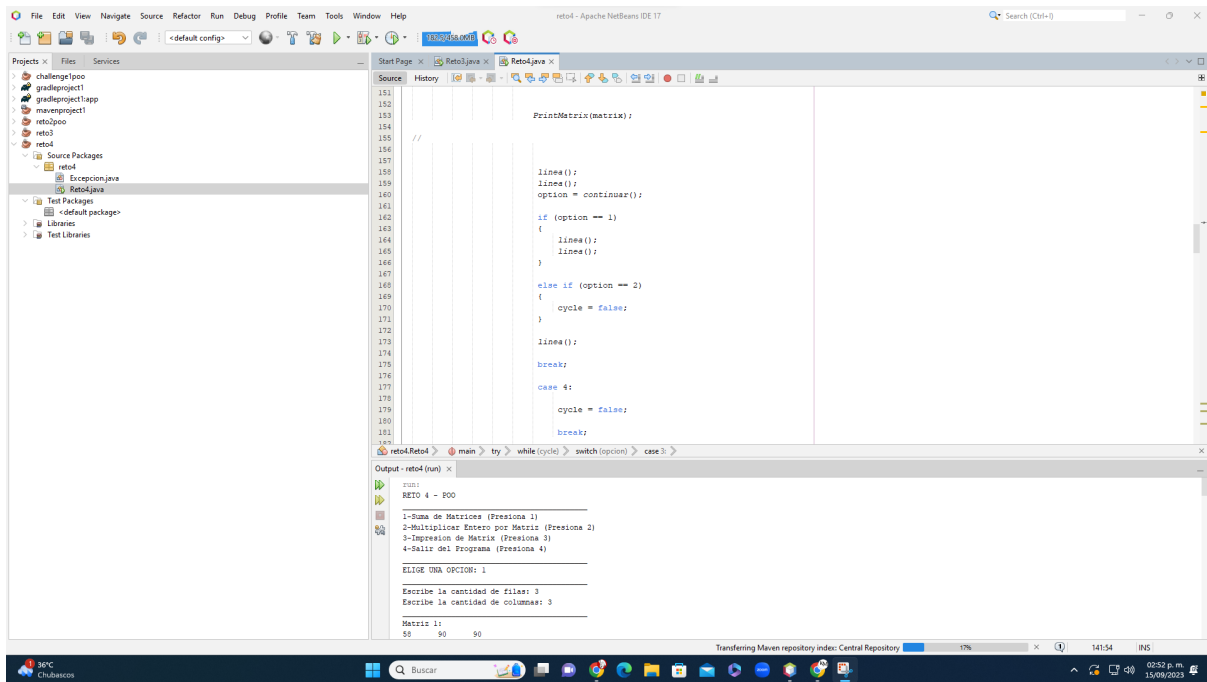
Matriz 1:
58  90  90

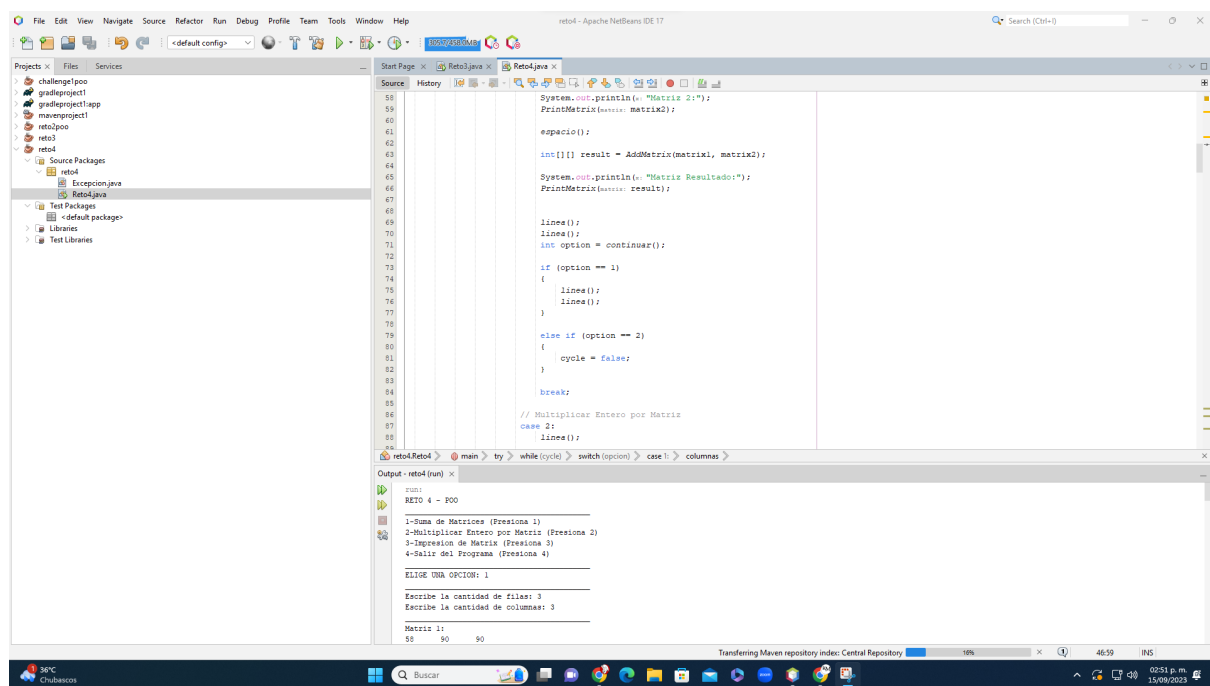
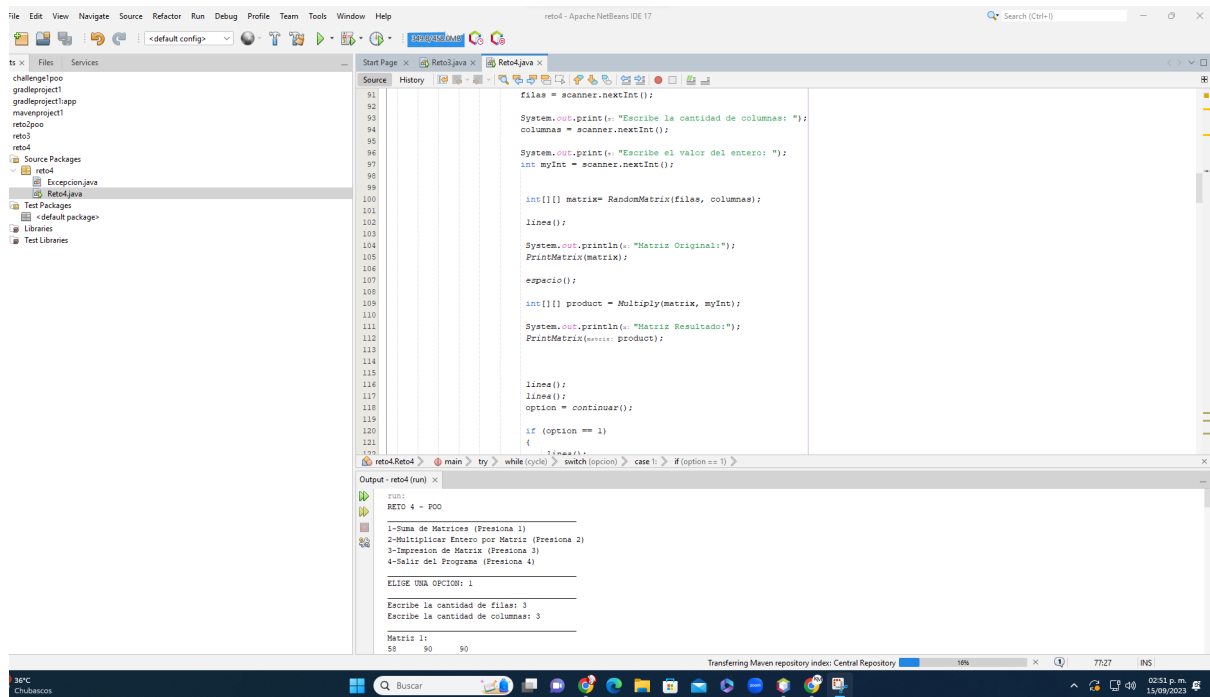
```



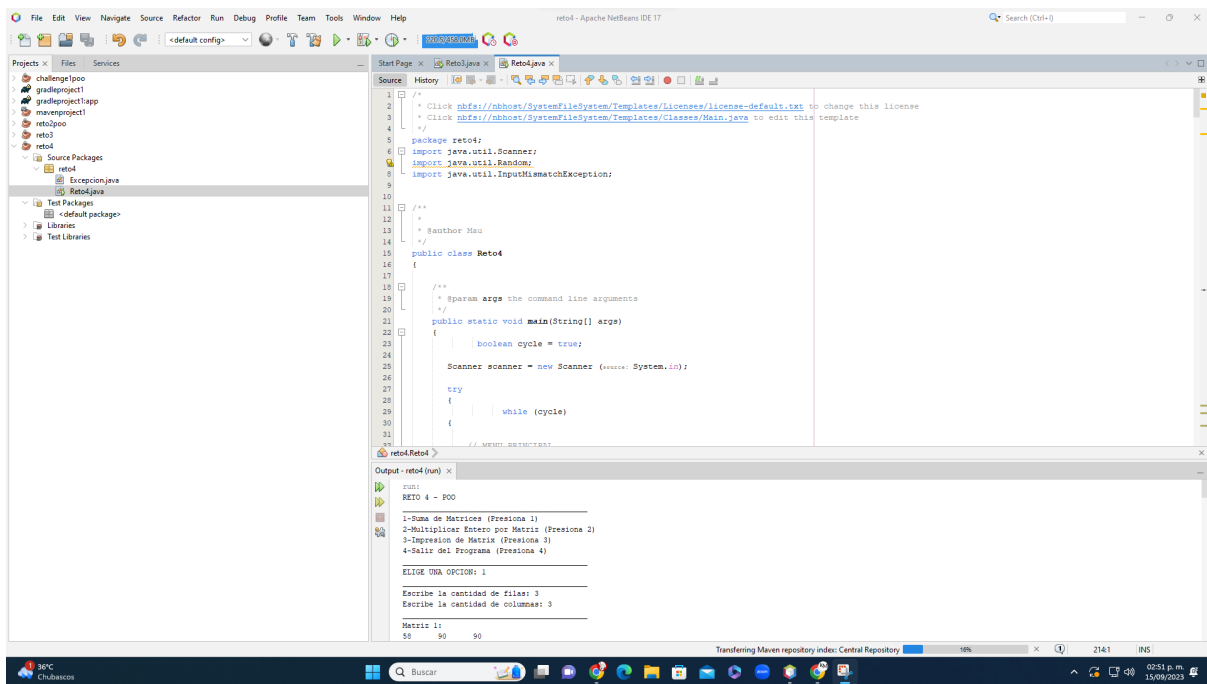
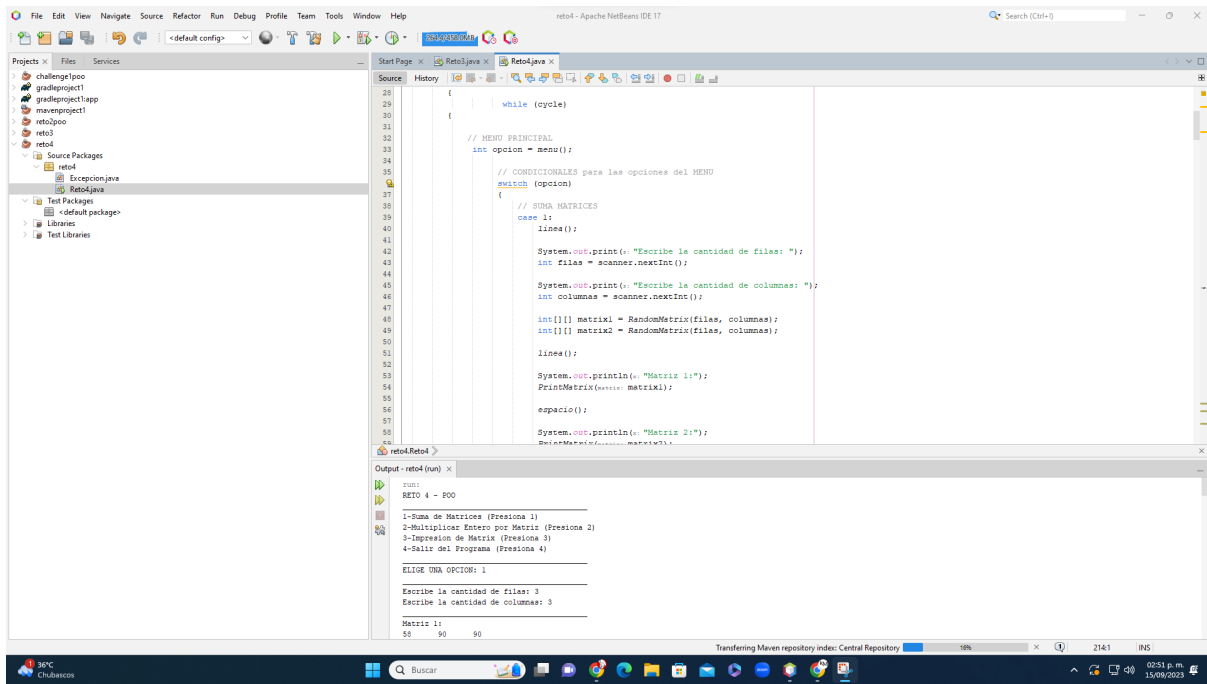




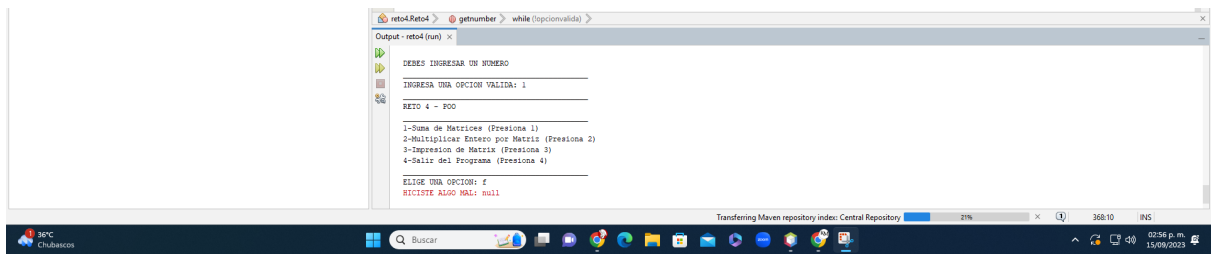




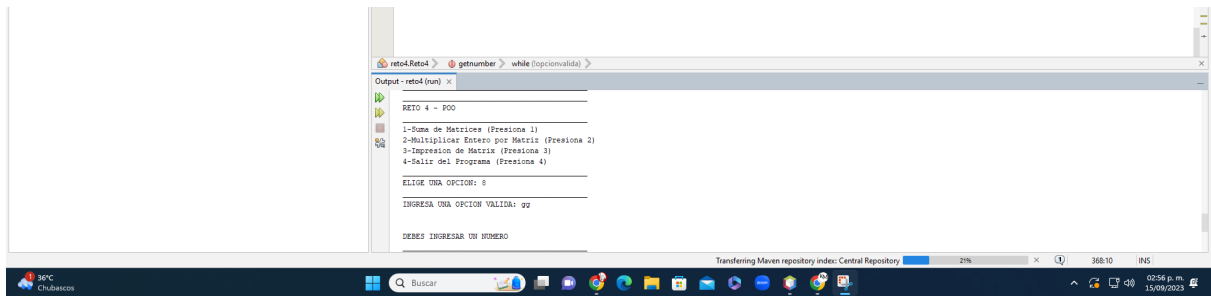




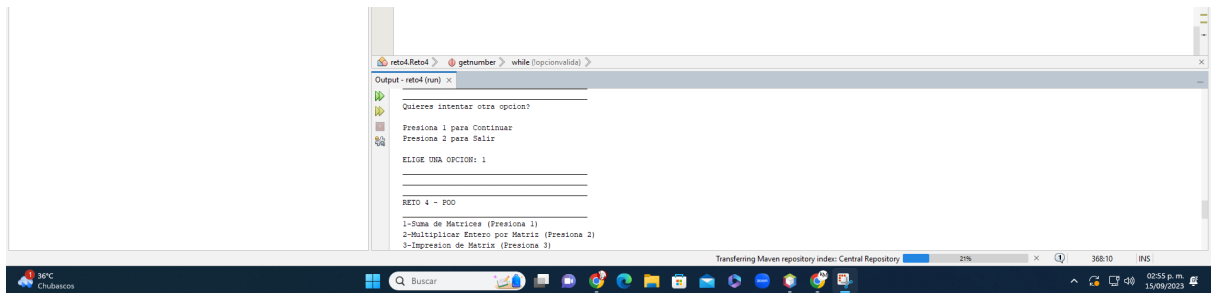
## PRUEBAS:



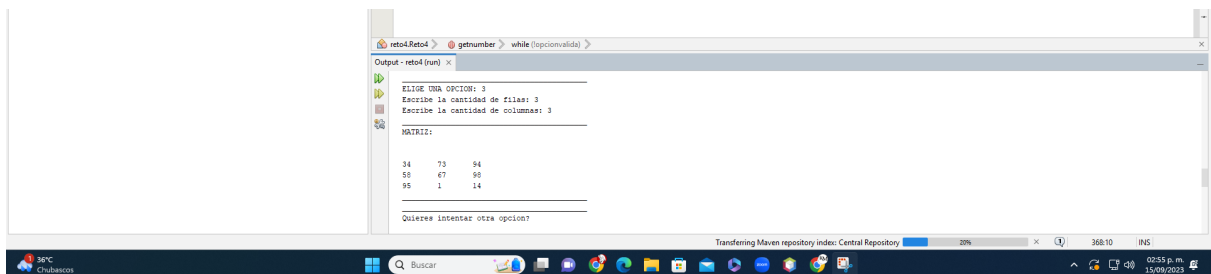
```
reto4.Reto4 > getnumber > while (opcionvalida) >
Output - reto4 (run) x
DEBES INGRESAR UN NUMERO
INGRESA UNA OPCION VALIDA: 1
RETO 4 - P00
1-Suma de Matrices (Presiona 1)
2-Multiplicar Entero por Matriz (Presiona 2)
3-Imprimir de Matriz (Presiona 3)
4-Salir del Programa (Presiona 4)
ELIGE UNA OPCION: 1
EXISTE ALGO MAS: null
```



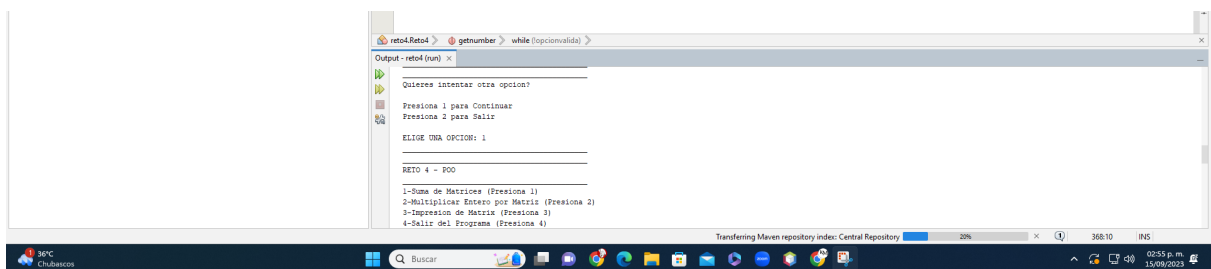
```
reto4.Reto4 > getnumber > while (opcionvalida) >
Output - reto4 (run) x
RETO 4 - P00
1-Suma de Matrices (Presiona 1)
2-Multiplicar Entero por Matriz (Presiona 2)
3-Imprimir de Matriz (Presiona 3)
4-Salir del Programa (Presiona 4)
ELIGE UNA OPCION: 2
INGRESA UNA OPCION VALIDA: 22
DEBES INGRESAR UN NUMERO
```



```
reto4.Reto4 > getnumber > while (opcionvalida) >
Output - reto4 (run) x
Quieres intentar otra opcion?
Presiona 1 para Continuar
Presiona 2 para Salir
ELIGE UNA OPCION: 1
RETO 4 - P00
1-Suma de Matrices (Presiona 1)
2-Multiplicar Entero por Matriz (Presiona 2)
3-Imprimir de Matriz (Presiona 3)
```



```
reto4.Reto4 > getnumber > while (opcionvalida) >
Output - reto4 (run) x
ELIGE UNA OPCION: 3
Escribe la cantidad de filas: 3
Escribe la cantidad de columnas: 3
MATRIZ:
34    73    94
58    67    96
95     1    14
Quieres intentar otra opcion?
```



```
reto4.Reto4 > getnumber > while (opcionvalida) >
Output - reto4 (run) x
Quieres intentar otra opcion?
Presiona 1 para Continuar
Presiona 2 para Salir
ELIGE UNA OPCION: 1
RETO 4 - P00
1-Suma de Matrices (Presiona 1)
2-Multiplicar Entero por Matriz (Presiona 2)
3-Imprimir de Matriz (Presiona 3)
4-Salir del Programa (Presiona 4)
```

```
reto4.Reto4 > getnumber > while (opcionvalida) >
Output - reto4 (run) x
Matriz Original:
58 72 9
52 67 24
95 44 4

Matriz Resultado:
114 216 24
156 201 72
205 132 12

Quieres intentar otra opcion?
```

```
reto4.Reto4 > getnumber > while (opcionvalida) >
Output - reto4 (run) x
ELIGE UNA OPCION: 1
RETO 4 - P00
1-Suma de Matrices (Presiona 1)
2-Multiplica Entero por Matriz (Presiona 2)
3-Imprimir de Matriz (Presiona 3)
4-Salir del Programa (Presiona 4)
ELIGE UNA OPCION: 2
Escribe la cantidad de filas: 3
Escribe la cantidad de columnas: 3
Escribe el valor del entero: 3
```

```
reto4.Reto4 > getnumber > while (opcionvalida) >
Output - reto4 (run) x
Matriz Resultado:
75 169 120
59 60 14
77 70 72

Quieres intentar otra opcion?
Presiona 1 para Continuar
Presiona 2 para Salir
ELIGE UNA OPCION: 1
```

```
reto4.Reto4 > getnumber > while (opcionvalida) >
Output - reto4 (run) x
Escribe la cantidad de filas: 3
Escribe la cantidad de columnas: 3
Matriz 1:
58 90 90
25 39 13
32 5 34
Matriz 2:
17 79 38
74 21 9
45 65 38
```

```
reto4.Reto4 > getnumber > while (opcionvalida) >
Output - reto4 (run) x
RETO 4 - P00
1-Suma de Matrices (Presiona 1)
2-Multiplica Entero por Matriz (Presiona 2)
3-Imprimir de Matriz (Presiona 3)
4-Salir del Programa (Presiona 4)
ELIGE UNA OPCION: 1
Escribe la cantidad de filas: 3
Escribe la cantidad de columnas: 3
Matriz 1:
58 90 90
```

## REPORTE:

-Primero empezaremos a explicar los métodos del código, el primer método que codificamos fue uno de tipo void llamado “espacio” que contiene solamente un dato de salida que muestra la función de “\n” con el propósito de poder ver mejor las matrices y separarlas en la consola al ejecutar el código, luego hicimos un método tipo void de líneas

para que aparezcan líneas en la consola y no sea tan confusa la información, hicimos un método para el menú del programa, básicamente hicimos un arreglo de tipo string para que guardar en las posiciones las opciones que debe elegir el usuario en el programa e hicimos un ciclo FOR para mostrar como dato de salida los elementos del arreglo. al final solamente le agregamos el dato de entrada para después en el Main proceder con las condicionales e incluimos al último la función de "return" para el dato de entrada que en este caso es un tipo Int y se llama opción. Luego hicimos un método de tipo Int llamado "continuar" para manejar el ciclo while principal del Main que nos retorna la variable "option". Ahora pasemos con los métodos de los procedimientos, el primero se llama "Add Matrix" y es la función para la primera opción del programa que es sumar 2 matrices, agregamos como parámetros de entrada 2 matrices, la matriz 1 y la matriz 2. ro del scope declaramos 3 variables int, la primera es para las filas que fueran igual a la longitud de la matriz 1, luego la segunda es para las columnas que fuera igual a la longitud de la posición 0 de la matriz 1 y la tercer variable es una matriz de tipo int para el resultado de la suma de ambas matrices básicamente, luego agregamos los ciclos FOR para que se despliegue la matriz correctamente como dato de salida y dentro del scope del segundo ciclo, declaramos que la matriz del resultado fuese igual a la matriz 1 + la matriz 2 y que se muestren todas las filas y columnas que hayamos proporcionando como datos de entrada al programa. ya solamente al final del método agregamos que nos regresara el resultado. El siguiente método es para mostrar la matriz como dato de salida, es del tipo Void y declaramos las filas y columnas con la misma equivalencia que el método anterior e hicimos los ciclos FOR necesarios para las matrices al ser llamadas en Main. Después hicimos un método del tipo Int para agregar valores aleatorios en las posiciones de las matrices, los parámetros de entrada son las variables para pedir la cantidad de filas y columnas, luego hicimos una matriz llamada "matrix" donde guardará como valores la cantidad de filas y columnas que queremos, importamos la libreria para hacer funciones random en Java y después declaramos una función de tipo random dentro del scope, dentro de los ciclos FOR agregamos que en los valores de la matriz se reflejaran numeros aleatorias del 0 al 99 y al final le pusimos que nos regresará como parámetro de salida la matriz "matrix". Después hicimos un método para multiplicar una matriz por una variable de tipo Int, agregamos como parámetros de entrada la variable del entero llamada "myInt" y la matriz "matrix". dentro del algoritmo hicimos una matriz donde guardamos como valores los producto del entero y los valores de la matriz, además de todo lo básico para hacer una función de una matriz, agregamos como parámetro de salida la variable del producto de la multiplicación. El ultimo metodo que realizamos fue para agregar una excepción extra al programa, la excepción es de agregar la opción correcta en el menú del programa básicamente, declaramos la variable de opción igual a 1 y una variable booleana como falsa, luego pusimos un ciclo while que mientras fuera verdadero con un se realizará lo siguiente en estas excepciones: En el TRY las instrucciones son básicamente de ingresar los números válidos del menú (1,2,3 y 4) para que el programa te regrese al menú principal, y declaramos la variable booleana como true, el algoritmo del CATCH solamente desplegará cuando el usuario escriba un char o string como dato de entrada en el parámetro del TRY. En el main, básicamente declaramos la variable booleana para el ciclo WHILE del menú y que siga operando el programa y agregamos más excepciones, en el TRY agregamos el algoritmo principal, en el catch pusimos un dato de salida que dice "HICISTE ALGO MAL" y en el finally solamente escribimos "FIN DEL PROGRAMA".

El algoritmo principal es básicamente llamar la función del menú y juntarla con la variable opción, hacer condicionales de switch y en el default mandamos a llamar a la función de que el usuario proporcione correctamente los datos de entrada.

## **CONCLUSIÓN:**

Habíamos aprendido sobre excepciones y funciones el semestre pasado con C#, únicamente lo que no sabíamos es que puedes declarar una variable booleana y ponerla en un ciclo WHILE para poder usar el programa que codificamos las veces que queramos, tampoco sabíamos que había más tipos de Excepciones en Java, aprendimos a usarlas más frecuentemente, el semestre pasado únicamente habíamos visto excepciones junto la manipulación de archivos en el último tema y lo aplicamos más que todo para la evidencia 3. En conclusión, entre mas experiencia tengamos codificando programas con nuevas funciones, mejores profesionales seremos en el futuro.