



UNIVERSIDAD AUTÓNOMA DE ZACATECAS

Practica 14: Recursión

Estudiante:

José Francisco Hurtado Muro

Profesor:

Dr. Aldonso Becerra Sánchez

May 22, 2025

Tabla de Contenidos

| | | |
|----------|---|----------|
| 1 | Actividades que debe realizar el alumno: | 3 |
| 1.1 | Actividad inicial: | 3 |
| 1.2 | Actividad 1: | 3 |
| 1.3 | Actividad 2: | 3 |
| 1.3.1 | Es binario o no | 3 |
| 1.3.2 | Convertir decimal a hexadecimal | 4 |
| 1.3.3 | Convertir decimal a binario | 5 |
| 1.3.4 | Encriptar | 7 |
| 2 | Código Agregado - UML | 8 |
| 3 | Pre-evaluación del Alumno | 9 |
| 4 | Conclusión | 9 |
| 5 | Referencias: | 9 |

Introducción

La recursividad es una poderosa herramienta que permite simplificar muchos programas que por su naturaleza son recursivos y que su solución iterativa puede resultar más compleja.

1 Actividades que debe realizar el alumno:

1.1 Actividad inicial:

Generar el reporte en formato IDC.

1.2 Actividad 1:

Primero genere la **Introducción**.

1.3 Actividad 2:

1.3.1 Es binario o no

Crea un método recursivo en Java que compruebe si un número es binario. Un número binario está formado únicamente por ceros y unos. Por ejemplo, un número binario es: 101001; un número no binario es 123001. El parámetro proporcionado como argumento por el usuario es un número de tipo de dato long.

Haga el programa (la cual es el **Desarrollo** del programa, junto con la captura de pantalla del programa funcionando).

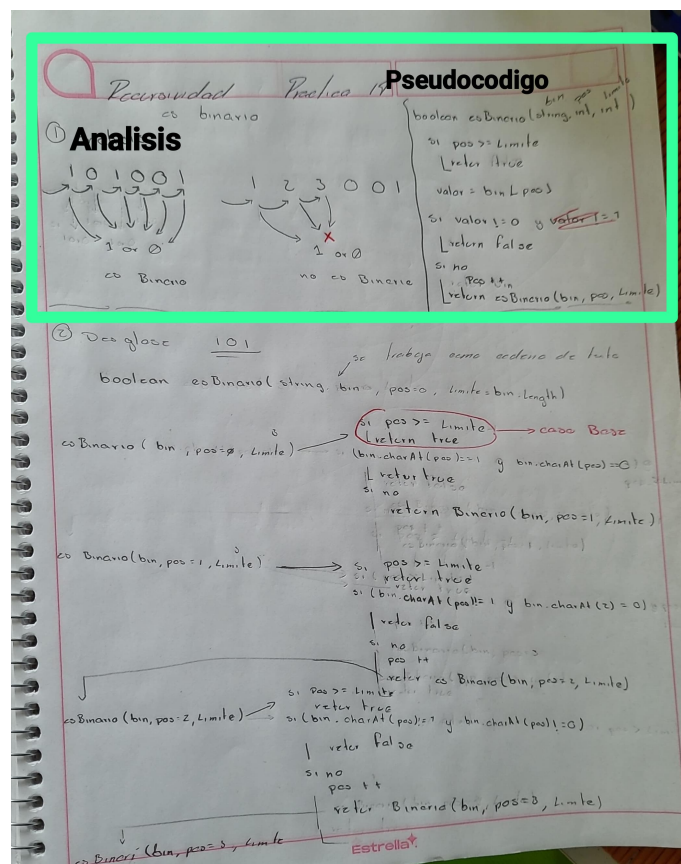


Figure 1: Análisis: Es binario o no

Explicación

Para este caso lo que se realiza fue recibir un cadena de texto el cual seria nuestro numero binario, dos enteros, la posición desde donde se quiere iniciar a revisar y el limite que es hasta donde se quiere revisar que por fines prácticos se trabajara con que el limite es el de cantidad de caracteres tenga la cadena de texto, entonces lo siguiente es revisar que la posición no sea mayor que el limite, en caso de serlo regresa true que en nuestro caso quiere decir que se reviso todas las posiciones, lo cual indica que si es binario, pero en caso de aun no llegar al limite, se estará obteniendo carácter por carácter, y se estará revisando, donde si es diferente a 1 y de 0, este significa que no es binario ya que no es ninguna de esas dos opciones, pero en caso que si sea uno de esos dos valores la posición incrementa en 1 y se manda a llamar así misma la función, repitiendo los mismos paso por cada carácter como vemos en la figura 2

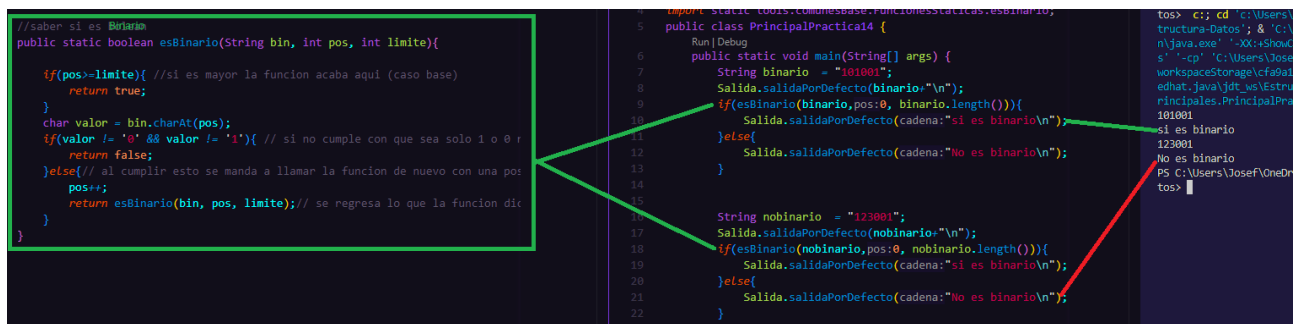


Figure 2: Funcionamiento: Es binario o no

1.3.2 Convertir decimal a hexadecimal

Construya una función en Java que convierta un número decimal en una cadena que represente el valor del número en hexadecimal (base 16).

Recordatorio: El cambio de base se realiza mediante divisiones sucesivas por 16 en las cuales los restos determinan los dígitos hexadecimales del número según la siguiente correspondencia:

| | | | | | | | | | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Resto | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Dígito | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |

Por ejemplo:

$$\begin{array}{rcl}
 65029 & \boxed{16} & \\
 5 & 4064 & \boxed{16} \\
 \downarrow & \downarrow & \downarrow \\
 5 & 0 & E \rightarrow FE05
 \end{array}$$

Haga el programa (la cual es el **Desarrollo** del programa, junto con la captura de pantalla del programa funcionando).

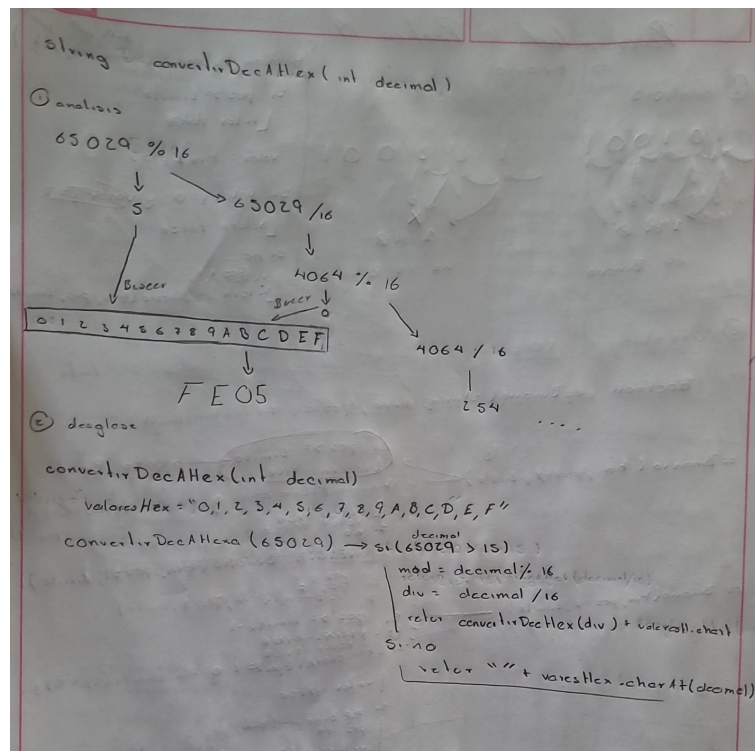


Figure 3: Análisis: Convertir decimal a hexadecimal

Explicación

En este método lo que se hizo fue que si el valor era mayor que 15, que es la base del hexadecimal, se le aplicara el mod para saber cual es el residuo y con este sacar un indice entre 0 y 15 que son los posibles valores en hexadecimal, para esto se retorna la una cadena compuesta por un valor que desconocemos que se saca con el auto llamado de la función, que lleva el mismo decimal dividido entre 16, mas el valor indicado por el mod, pero cuando el decimal alcance estar en un rango de 0 o 16, este regresara en valor que indique su valor como indice en los posibles valores de Hexadecimal, tal y como se ve en la figura 4.

```
//convertir decimal a hexadecimal
public static String convertirDecAHex(int decimal)
{
    String valoresHex = "0123456789ABCDEF"; //valores de hexadecimal
    if(decimal > 15){ //mientras el valor decimal se mayor
        int mod = decimal%16; //se le saca el mod que es el indice del valor que buscamos
        return convertirDecAHex(decimal/16) + valoresHex.charAt(mod); //cadena compuesta por salidas de la funcion
    }else{
        return "" + valoresHex.charAt(decimal); //el valor ya esta dentro del rango de 0 a 16
    }
}
```

Figure 4: Funcionamiento: Convertir decimal a hexadecimal

1.3.3 Convertir decimal a binario

Hacer un programa recursivo en Java que convierta un número decimal en binario. Consiste en un método parecido a la factorización en números primos. Es relativamente fácil dividir cualquier número entre 2. Este método consiste básicamente en divisiones sucesivas. Dependiendo de si el número es par o impar, colocaremos un cero o un uno en la columna de la derecha.

Si es impar, le restaremos uno y seguiremos dividiendo entre dos (y podremos un 1 en el lado derecho como anteriormente expongo), hasta llegar al resultado final que debe ser siempre 1.

Después, sólo nos queda tomar los resultados de la columna de la derecha y ordenar los dígitos de abajo para arriba, y tendremos nuestro número convertido en binario.

Ejemplo:

```
150|0
75|1*
37|1
18|0
9|1
4|0
2|0
1|1
```

El resultado para 150 en base decimal es: 10010110 en base binaria.

*Aquí ponemos 1 al lado derecho y restamos 1 de 75 para poder seguir dividiéndolo entre 2, el resultado lo ponemos debajo, y así sucesivamente.

Haga el programa (la cual es el **Desarrollo** del programa, junto con la captura de pantalla del programa funcionando).

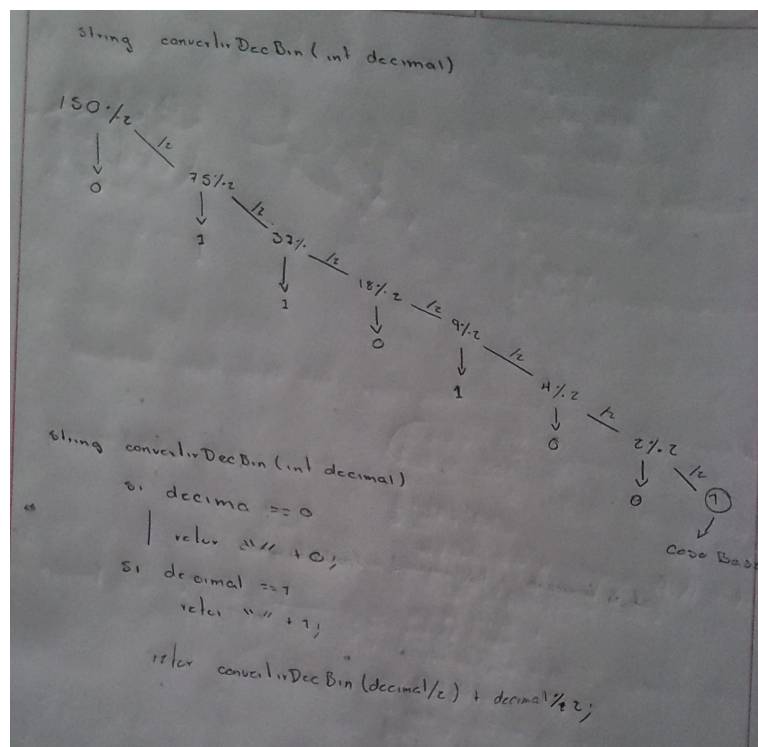


Figure 5: Análisis: Convertir decimal a binario

Explicación

Para este caso lo que se realizó es recibir un decimal, el cual lo que hace es que si es mayor o diferente de 1 lo que haces es aplicarle un mod para saber su residuo, que por general solo regresa 0 o 1, y estos valores se van agregando en una cadena de texto, hasta que el decimal llega a 1, que este es el caso base, y se vería como en la figura 6

```
//convertir a binario
public static String convertirDecBin(int decimal){
    if(decimal != 1){ //revisa cuando el valor decimal no haya lle
        return convertirDecBin(decimal/2) + decimal%2;
    }else{
        return ""+ 1; //Caso base: se llega a 1
    }
}
```

123001
No es binario
valor hexadecimal de
FF05
valor binario 150:
10010110
PS C:\Users\joser\On

Figure 6: Funcionamiento: Convertir decimal a binario

1.3.4 Encriptar

Utilizando la actividad 3 de la práctica 9 (el enunciado de encriptar y des-encriptar cadenas). Haga un programa recursivo que lea un archivo de texto (usando los programas que ya tenemos que leen un archivo de texto plano) en un arreglo y encripte cada posición del arreglo (el cual es una cadena, equivalente a una línea de texto de un archivo) y lo guarde en otro arreglo paralelo, el cual deberá ser grabado en otro archivo de texto. Posteriormente haga el programa que des-encripte siguiendo la misma lógica, grabando el resultado en un archivo.

Haga el programa (la cual es el **Desarrollo** del programa, junto con la captura de pantalla del programa funcionando).

Figure 7: Análisis:

Explicación

Figure 8: Funcionamiento:

2 Código Agregado - UML

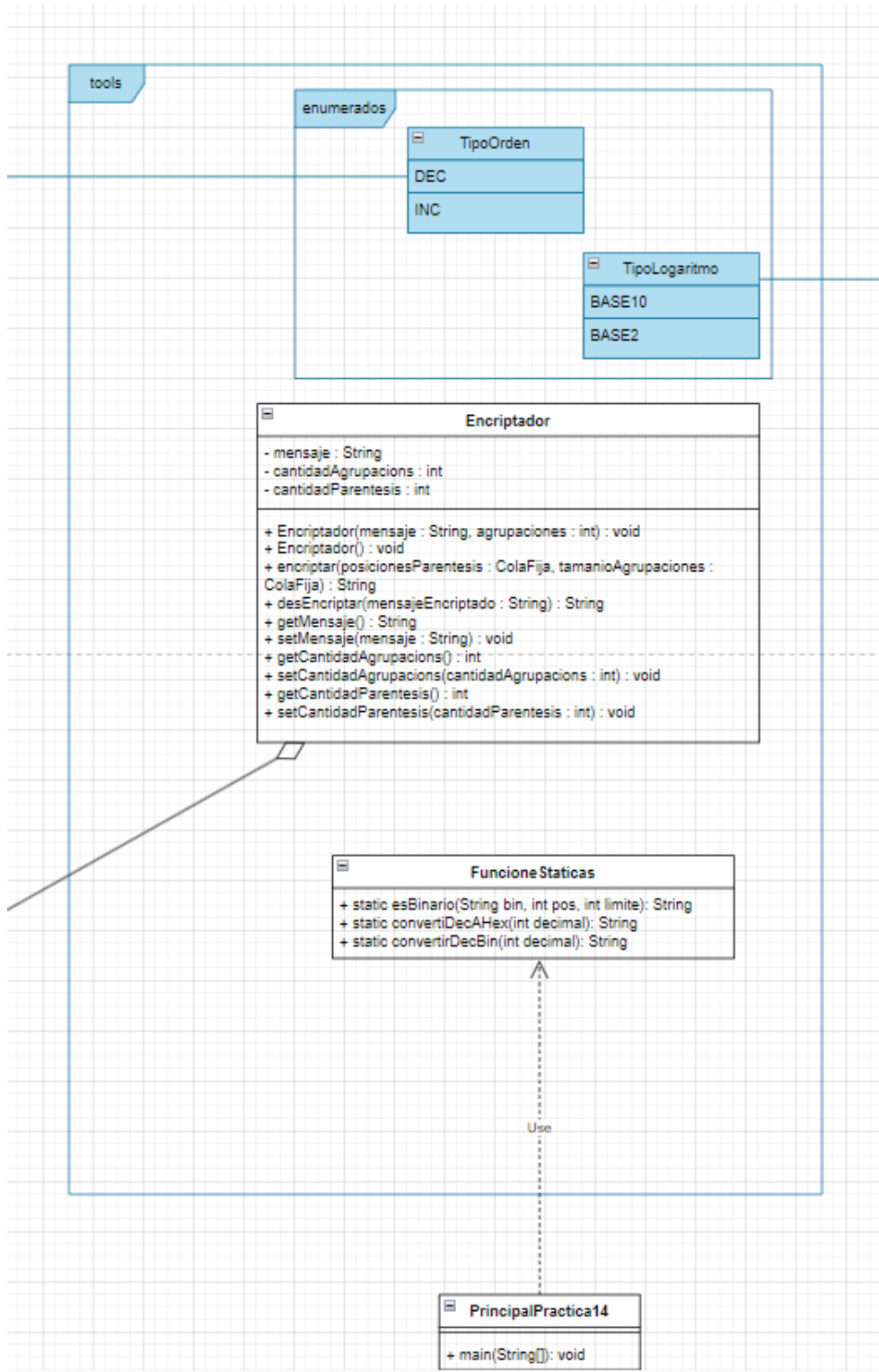


Figure 9: UML

3 Pre-evaluación del Alumno

| Criterio | Evaluación |
|--|------------|
| Cumple con la funcionalidad solicitada | Sí |
| Dispone de código auto-documentado | Sí |
| Dispone de código documentado a nivel de clase y método | Sí |
| Dispone de indentación correcta | Sí |
| Cumple la POO | Sí |
| Dispone de una forma fácil de utilizar el programa para el usuario | Sí |
| Dispone de un reporte con formato IDC | Sí |
| La información del reporte está libre de errores de ortografía | Sí |
| Se entregó en tiempo y forma la práctica | Si |
| Incluye el código agregado en formato UML | Sí |
| Incluye las capturas de pantalla del programa funcionando | Sí |
| La práctica está totalmente realizada (especifique el porcentaje completado) | 80% |

Table 1: Evaluación de la práctica

4 Conclusión

Se entendió de manera correcta el funcionamiento, uso y creación de metodos recursivos y su posible uso en diferentes cosas.

5 Referencias:

- Cairo, Osvaldo; Guardati, Silvia. *Estructura de Datos, Tercera Edición*. McGraw-Hill, México, Tercera Edición, 2006.
- Mark Allen Weiss. *Estructura de datos en Java*. Ed. Addison Wesley.
- Joyanes Aguilar, Luis. *Fundamentos de Programación. Algoritmos y Estructuras de Datos*. Tercera Edición, 2003. McGraw-Hill.