

DESARROLLO TALLER 1.

Integrantes:

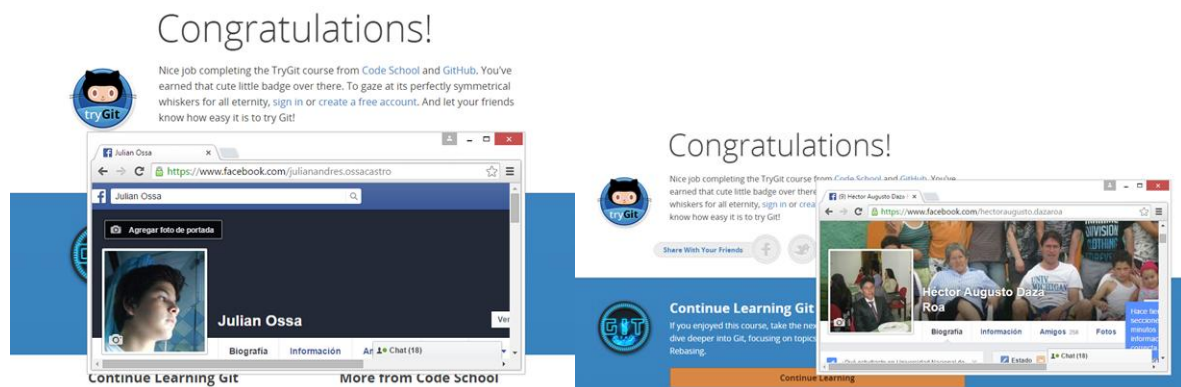
-Héctor Augusto Daza Roa: -Hadazar@unal.edu.co

-Github: Hadazar.

-Julian Andres Ossa Castro: -Jaossacas@unal.edu.co

-Github:Jaossacas.

Repositorio: <https://github.com/Hadazar/Programacion-Orientada-a-Objetos-2015-2>



EXPLICACIÓN DE EJERCICIOS:

Ahorcado Jaossacas: En el ejercicio del ahorcado lo que hice fue definir los objetos que iba a necesitar en el transcurso del ejercicio desde el principio de este, lo demás fueron algoritmos encargados de analizar las respuestas de los usuarios y si estas estaban permitidas o no o si no se repetían o si ya finaliza el juego y alguno ganaba o perdía, finalizando el código ya inicie con la elaboración tanto del diseño del juego como del ahorcado indicador de cuantas oportunidades le quedaban al usuario y una vez hecho eso lo finalice.

Ahorcado Hadazar: El código de este programa lo podemos dividir básicamente en tres etapas: etapa preliminar, desarrollo del juego y fin del juego. En la etapa preliminar, lo primero que se realiza es una instanciación de los objetos “Scanner”, “File”, “FileReader” y “BufferedReader” para la realizar posteriormente la lectura de archivo y de pantalla; luego se genera un número aleatorio entre el 1 y el 450 (cantidad de palabras contenidas en el documento de texto) y a través de un ciclo se realizan ese mismo número de lecturas al archivo, seleccionando así una palabra al azar; posteriormente se crean las principales variables y los principales arreglos con los que trabaja el programa; finalmente se usan ciclos para llenar el arreglo que representa a la palabra oculta con el símbolo “_”, y para llenar al arreglo que contendrá las letra usadas con espacios vacíos. El desarrollo del juego consta de un ciclo cuya continuidad termina si la cantidad de aciertos es igual a la longitud de la palabra o si la cantidad de errores es igual a 7 (cantidad de segmentos que posee el dibujo: q(x_x)p); cada etapa del ciclo consta de una impresión en pantalla de la palabra oculta a través de un ciclo, la impresión del dibujo y de la lista de letras usadas, la comprobación de

que la letra introducida no había sido introducida antes, y en caso de no haberse repetido la letra se compara esta con la palabra (si la palabra contiene la letra introducida, se incrementa el número de aciertos en una unidad por cada posición en donde esté contenida; si la palabra no contiene la letra introducida, se incrementa el número de errores en una unidad y si aumenta un segmento en el dibujo). El fin del juego tiene dos casos posibles: la victoria y la derrota; si el número de aciertos es equivalente a la longitud de la palabra el jugador gana (se imprime en pantalla la palabra y el mensaje “Has ganado!!!”); si el número de errores es equivalente a 7 el dibujo se completa y por lo tanto el jugador pierde (se imprime en pantalla la palabra hasta donde la descubrió el jugador, el dibujo completo y la frase “Has perdido!!!, la palabra secreta era: “ junto con la palabra correcta).

A. En este programa se crearon dos variables de tipo “double” que almacena los dos valores que ingresa el usuario. El programa está compuesto básicamente por condicionales; el primero de ellos, abarca la mayor parte del código y verifica que los números introducidos por el usuario sean enteros; en caso ser enteros, el programa procede a los demás condicionales, los cuales a su vez definen si los números son pares o impares, cuál es el mayor y si el primero es múltiplo del segundo; en total son cinco condicionales.

B. En el taller el ejercicio de separar los dígitos de un entero solicitado al usuario se elaboró de manera sencilla y concisa manejando códigos simples como lo son el uso de Scanner y otros términos que se complementan de algoritmos fáciles de comprender y manejar que lo que hacen es encargarse de manipular cada valor impreso por el usuario y que los tome por separado y se repite el proceso hasta tomar todos los valores en el número completo e imprimir uno por uno con un determinado espacio.

C. Este programa se vale de cinco variables y un ciclo para cumplir con su función. La primera variable, llamada “Tester”, define la continuidad del ciclo; la segunda variable, llamada “Número”, almacena el valor introducido por el usuario en cada etapa del ciclo; la tercera variable, llamada “acumulado”, almacena la sumatoria de todos los valores que han sido introducidos por el usuario; la cuarta variable, llamada “Cantidad_números”, almacena la cantidad de veces que ha sido repetido el ciclo (cantidad que es equivalente a la cantidad de números que ha sido ingresada por el usuario); la quinta variable, llamada “promedio”, es el promedio de los números que han sido ingresados por el usuario (cociente entre “acumulado” y “Cantidad_números”). Cada etapa del ciclo consiste en una inicialización de la variable “número” por parte del usuario, y de un condicional que permite al usuario romper el ciclo al cambiar el valor de la variable “tester” (si ingresa el número 9999), en caso contrario se agrega el valor introducido en la sumatoria de la variable “Acumulado” y se incrementa la variable “Cantidad_números” en una unidad.

D. Este programa se vale de cuatro variables, un ciclo, y cinco condicionales para cumplir con su función. La primera variable, llamada “monto”, almacena la cantidad de dinero que el usuario desea retirar; la segunda variable, llamada “Billete_moneda”, almacena el valor de la denominación con la que se está trabajando, esta comienza con un valor de “50000” y al final de cada etapa del ciclo disminuye su denominación (50000 → 20000 → 10000...); la

tercera variable, denominada “Repeticiones_ciclo”, almacena la cantidad de veces que el ciclo se ha repetido, esta variable permite determinar el proceso con el cual se debe disminuir la denominación de “Billete_moneda”, si su valor es múltiplo de tres la denominación se multiplica por “ $\frac{2}{5}$ ” en caso contrario se divide en dos ((0)50000($\frac{2}{5}$) → (1)20000($\frac{1}{2}$) → (2)10000($\frac{1}{2}$) → (3)5000($\frac{2}{5}$) → (4)2000($\frac{1}{2}$) → (5)1000($\frac{1}{2}$) → (6)500($\frac{2}{5}$) → (7)200($\frac{1}{2}$) → (8)100($\frac{1}{2}$) → (9)50); La cuarta variable, llamada “Cantidad_Billete”, almacena la cantidad de billetes o monedas de la denominación con la que se esté trabajando. Cada etapa del ciclo consta de la inicialización de la variable “Cantidad_Billete”, la impresión en pantalla de esta cantidad en caso de ser mayor a cero, la disminución del monto en la cantidad de dinero que está representada por la cantidad de billetes calculada, la disminución de la denominación de “Billete_moneda” y un incremento de la variable “Repeticiones_ciclo” en una unidad.

E. En este ejercicio del taller se elaboró algo semejante al ejercicio b, que fue solicitarle un valor al usuario y con respecto a este se fueran imprimiendo uno a uno hacia abajo cada uno de los números primos existentes entre este valor y el 0, lo cual se logra solicitando la impresión de un resultado dado por el algoritmo elaborado en el código, y que este se repita hasta hallar todos los valores.

F. Este código fue complejo de elaborar ya que aunque logramos hacerlo competitivo entre dos servidores humanos no lo logramos entre humano y computadora, esto lo logramos gracias al uso de códigos encargados de leer operaciones complejas como lo son el booleanos y otros semejantes, además de usar un diseño elaborado por impresión en el cual se diseña el tablero de juego y basados en esto colocamos, condicionales, operaciones de lectura, restricciones para los usuarios para que estos siguieran las reglas del juego sin salirse del esquema y la temática de este.