
Informe del Taller # 02

# Introducción a la Programación Orientada a Objetos

Christian Camilo Taborda Campiño 1632081-3743

Cristian Camilo Vallecilla Cuellar 1628790-3743

Juan Sebastián Paz Velásquez 1626846-3743

# Punto 1: (Letrero)

Para el desarrollo de este programa se diseñó una clase -Letrero-, la cual entre sus métodos tenía tres herramientas que ayudaban a convertir la entrada en la salida deseada. Uno de los métodos pasaba las letras de minúsculas a mayúsculas, otra agregaba la cantidad de espacios necesarios, y la última dividía las líneas de la entrada con la ayuda de la función -substring- y la estructura de iteración -for-. Al realizar una instancia de -Letrero- se usaba el algoritmo con los tres métodos y después se imprimía la salida:

## Implementación:

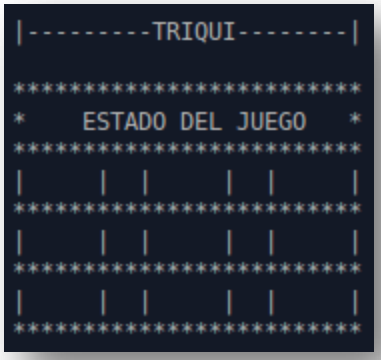
```
Ingrese la frase:
Sere un maestro pokemon

**********
*        S*
*       SE*
*      SER*
*     SERE*
*    SERE *
*   SERE U*
*  SERE UN*
* SERE UN *
*ERE UN M*
*RE UN MA*
*E UN MAE*
* UN MAES*
*UN MAEST*
*N MAESTR*
* MAESTRO*
*MAESTRO *
*AESTRO P*
*ESTRO PO*
*STRO POK*
*TRO POKE*
*RO POKEM*
*O POKEMO*
* POKEMON*
*POKEMON *
*OKEMON  *
*KEMON   *
*EMON    *
*MON     *
*ON      *
*N       *
*        *
**********
```

# Punto 2: (Triqui)

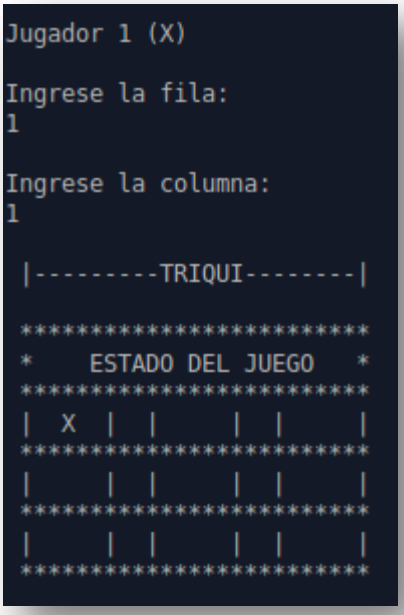
Para este programa se desarrollaron dos clases: -Jugador- y –Tablero-, a través de las cuales se implementó la relación de uso en uno de los métodos de –Jugador-, el cual modificaba uno de los elementos de la matriz que hacía parte de los atributos de –Tablero- por medio de una posición (X,Y). También se implementaron métodos en el -main- y la clase –Tablero- para validar si una entrada era correcta o no, y para decidir cuándo terminar la ejecución; todo lo anterior a través de la estructura condicional –if-.

# Implementación:

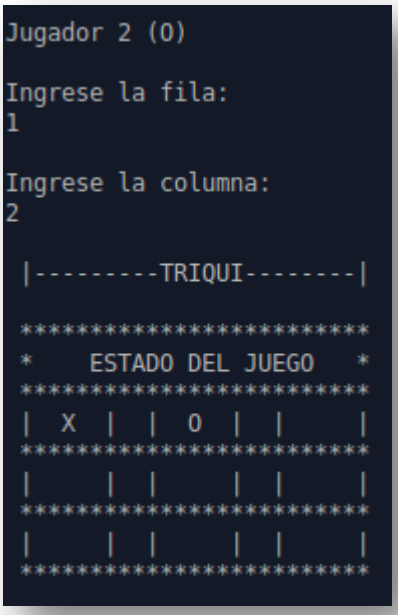
Impresión del tablero inicial:



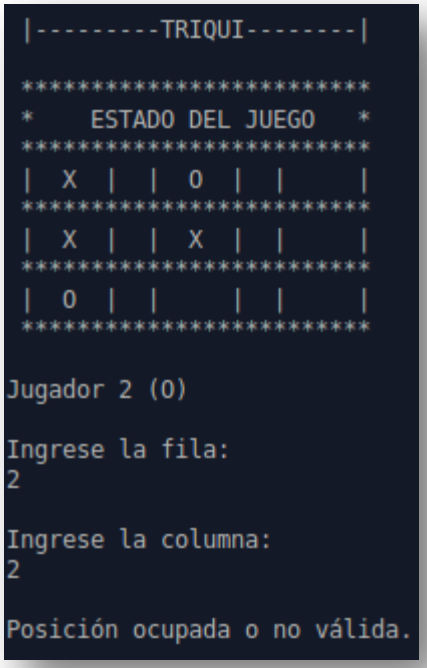
Impresión de una entrada del Jugador 1:



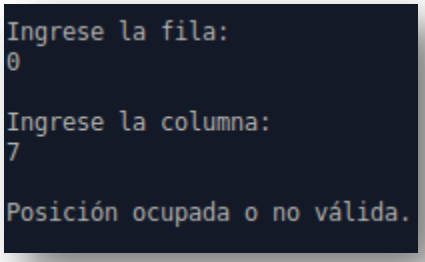

Impresión de una entrada del Jugador 2:



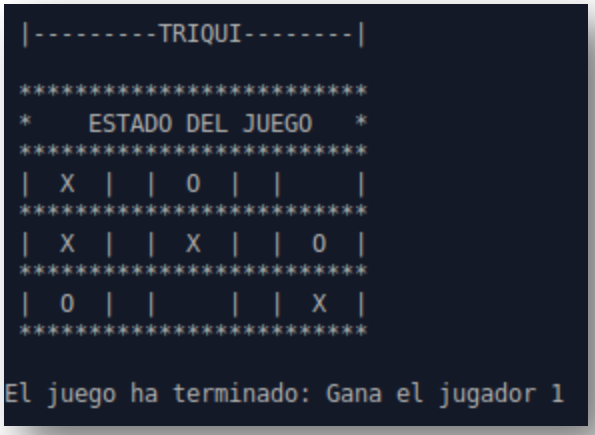
Ingreso de una posición ocupada:




Ingreso de una posición incorrecta:



Finalización del juego:

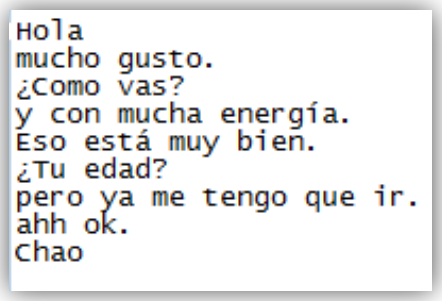


# Punto 3: (Conversación)

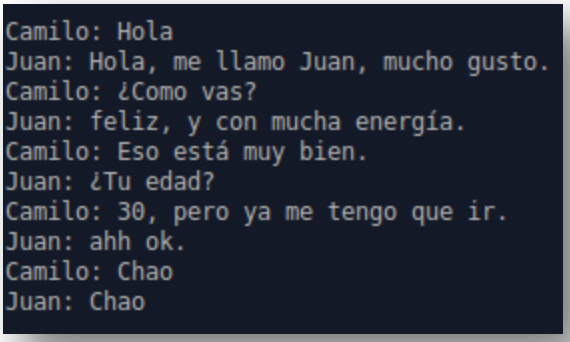
Para este punto se diseñaron dos clases, -Persona- y -Conversacion-. Se leía el archivo de texto ingresado línea a línea hasta encontrar la palabra “chao”, y cada línea se almacenaba en un vector. Una vez listo, el vector se pasaba como atributo para crear un objeto de la clase -Conversacion-. La clase –Persona- contiene los métodos –hablar- y –contestar-, los cuales retornan -strings-, la diferencia está en que –contestar- varía dependiendo de otro –string- de entrada, si esté hace parte del grupo de comandos, retornará el string junto con una parte extra en el mensaje dependiendo del comando que sea. El programa se basa en la estructura iterativa –while- para obtener líneas de la clase –Conversacion- y hacer que los objetos de –Persona- se contesten los mensajes uno a otro hasta encontrar la palabra “chao” en alguno de los mensajes emitidos.

## Implementación:

Este es el archivo (.txt) de entrada:



Esta es la salida:

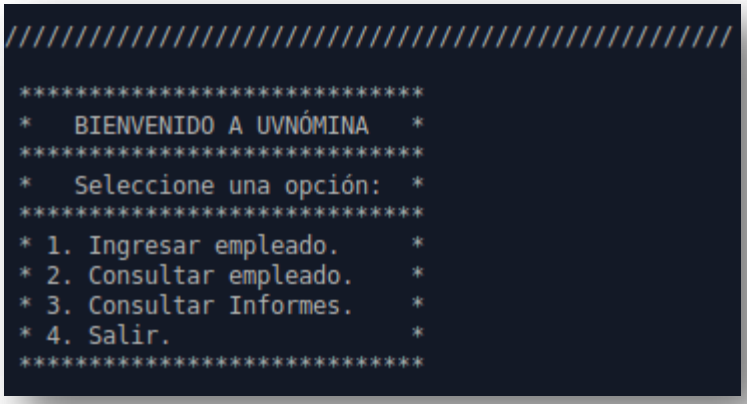



## Punto 4: (Uvnómina)

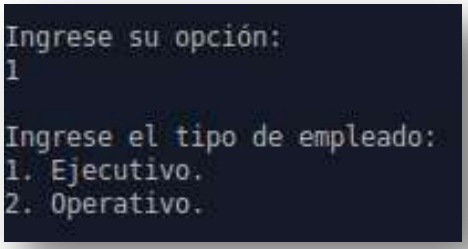
Para este punto se crearon varias clases. –Cargo- y –Salario-, las cuales con relación de composición complementaban a –Empleado-. Luego con relación de herencia se diseñaron las clases –Empleado_Ejecutivo- y –Empleado_Operativo-, por último se creó la clase –Consolidado-. El programa funciona a través de un vector dinámico para almacenar los objetos de empleados que se van ingresando. Luego sólo es usar los métodos -get- correspondientes a cada clase para mostrar todos los datos que el programa va almacenando. Para decidir qué actividad realizar el programa utiliza la estructura condicional –switch-.

## Implementación:

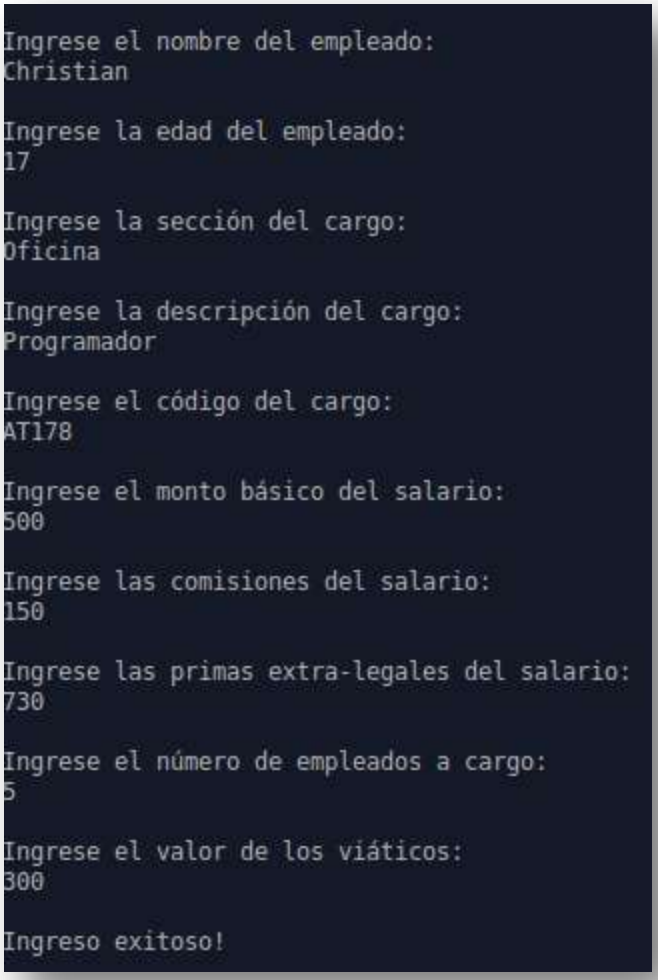
Impresión del menú principal del programa:



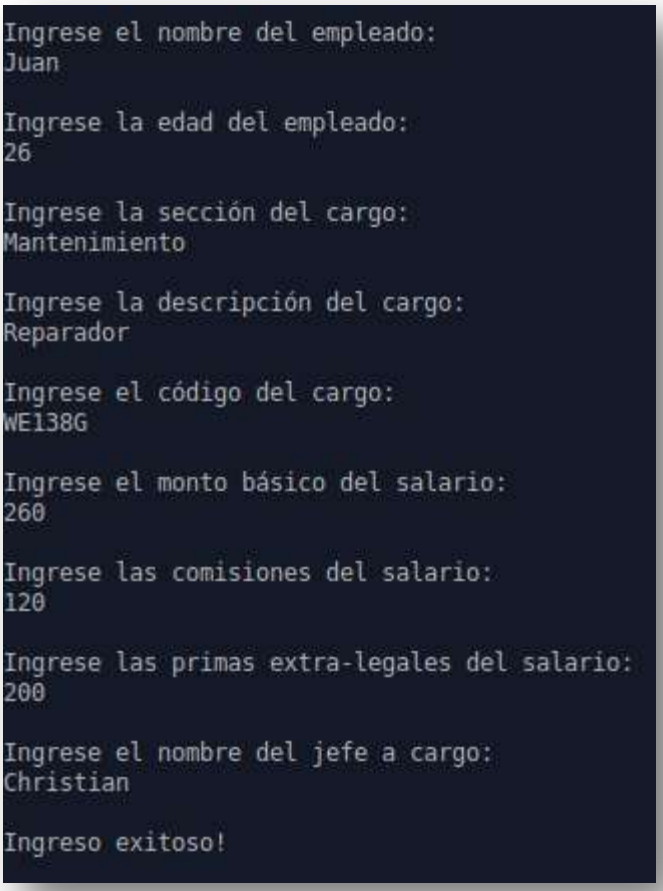

Impresión de opciones al elegir la opción 1 del menú principal:



Recolección de datos para el ingreso de un empleado ejecutivo:

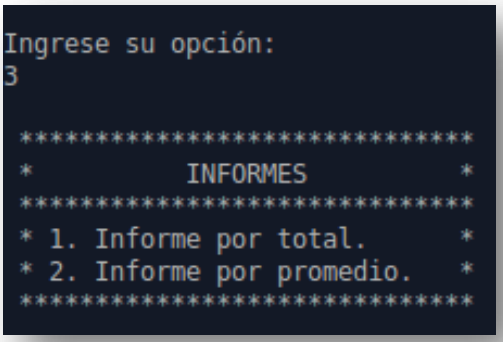


Recolección de datos para el ingreso de un empleado operativo:

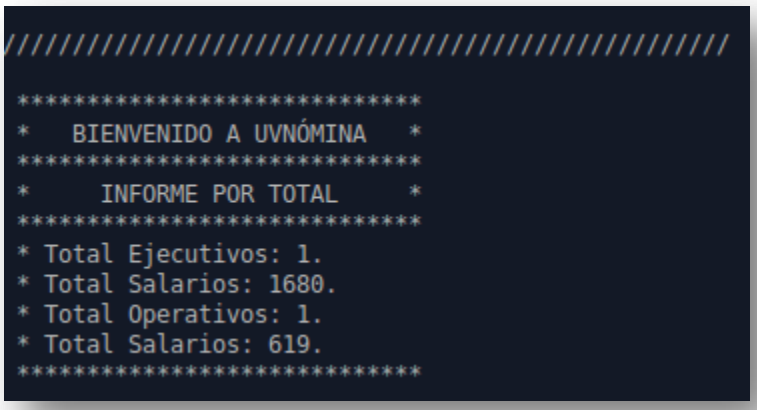


Consulta de un empleado (opción 2 del menú principal):

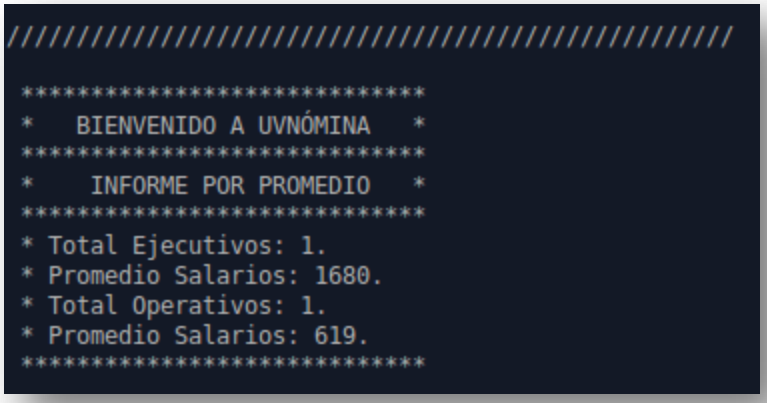

Impresión del mini-menú al seleccionar la opción 3 del menú principal:



Consulta de informes por total:




Consulta de informes por promedio:




Salir del programa (opción 4 del menú principal):