

Trabajo obligatorio de Programación 1.

Curso 2018-19

7.1. Introducción

En este trabajo obligatorio se va a seguir trabajando con datos de estaciones, usos y usuarios del sistema Bizi Zaragoza. Los desarrollados de la práctica 6 relativos a estaciones y usos del sistema Bizi serán integrados en este trabajo, al que se le va a añadir funcionalidad relativa a los usuarios.

Como se ha explicado en el enunciado de la práctica 6, se dispone de un fichero de estaciones y de dos ficheros de usos, uno de usos comprendidos entre octubre de 2016 y marzo de 2017 (al que se ha denominado «usos-16.csv», y otro de usos comprendidos entre marzo de 2017 y agosto de 2017 («usos-17.csv»). Estos dos ficheros de usos vienen acompañados de dos ficheros de usuarios en esos mismos dos periodos, que se encuentran disponibles en la sección «Código C++ y datos» de la web de la asignatura como «usuarios-16.csv» y «usuarios-17.csv»¹.

En el momento de escribir este enunciado, estos son los dos únicos periodos para los que se han publicado datos, pero no se descarta que puedan publicarse otros posteriormente. El programa solicitado en este trabajo debería ser capaz de incluirlos con mínimos cambios en su código.

El programa objeto de este trabajo es un intérprete de órdenes o comandos que, reiteradamente, solicitará una orden al usuario y la ejecutará, hasta que el usuario decida dar por concluida la ejecución del programa.

En la siguiente sección se establecen los requisitos que debe cumplir el programa solicitado. En la sección 7.3 se describe la sintaxis y el contenido de los ficheros de usuarios y en la sección 7.4 se dan algunas pautas y guías para el diseño e implementación de la solución. Finalmente, se indica en que van a consistir los entregables del trabajo y el mecanismo de entrega.

7.2. Requisitos

7.2.1. Inicio del programa

Cuando el programa comienza su ejecución, solicita al usuario que indique el conjunto de datos con el que va a trabajar inicialmente. Como se dispone de dos pares de ficheros («usos-16.csv» con «usuarios-16.csv» por una parte, y «usos-17.csv» con «usuarios-17.csv» por otra), bastará con que el usuario elija entre las opciones «16» o «17». No obstante, el programa indicará cuales son estas opciones, de acuerdo con el siguiente modelo de interacción:

¹Estos ficheros, denominados originalmente «UsuariosOct_16-Mar17.csv» y «UsosMar17_Ago17.zip», han sido obtenidos de <https://www.zaragoza.es/sede/portal/datos-abiertos/servicio/catalogo/70#CSV>

```
Elección de ficheros de usos y usuarios. Opciones disponibles:  
16: octubre 2016 a marzo 2017  
17: marzo 2017 a agosto 2017  
Introduzca una opción: 16  
El fichero "../Datos/Bizi/usos-16.csv" existe y contiene 1024820 utilizaciones.  
El fichero "../Datos/Bizi/usuarios-16.csv" existe y contiene 20751 usuarios.
```

El programa ofrecerá las dos opciones disponibles y solicitará una. Cuando el usuario introduzca su respuesta, el programa comprobará si existe el fichero correspondiente de usos e informará de el número de usos que contiene. Seguidamente, comprobará que existe el de usuarios e informará de cuántos usuarios contiene.

Si el usuario introduce una opción para la que no hay ficheros, informará de que el fichero correspondiente no existe, y volverá a informar sobre las opciones disponibles y a solicitar una nueva hasta que el usuario elija una opción disponible:

```
Elección de ficheros de usos y usuarios. Opciones disponibles:  
16: octubre 2016 a marzo 2017  
17: marzo 2017 a agosto 2017  
Introduzca una opción: 18  
No se ha podido abrir el fichero "../Datos/Bizi/usos-18.csv".  
  
Elección de ficheros de usos y usuarios. Opciones disponibles:  
16: octubre 2016 a marzo 2017  
17: marzo 2017 a agosto 2017  
Introduzca una opción: 15  
No se ha podido abrir el fichero "../Datos/Bizi/usos-15.csv".  
  
Elección de ficheros de usos y usuarios. Opciones disponibles:  
16: octubre 2016 a marzo 2017  
17: marzo 2017 a agosto 2017  
Introduzca una opción: 17  
El fichero "../Datos/Bizi/usos-17.csv" existe y contiene 1016934 utilizaciones.  
El fichero "../Datos/Bizi/usuarios-17.csv" existe y contiene 20817 usuarios.
```

Los ficheros estarán ubicados en el un directorio «Datos/Bizi» ubicado en el directorio «Programacion1». Se debe suponer que el programa se ejecutará en el directorio «Programacion1/Trabajo2018-19/Trabajo» (ver sección 7.4).

7.2.2. Segundo paso del programa

A continuación, el programa informará de cuáles son las órdenes que acepta, mostrando la siguiente información:

```
ÓRDENES DISPONIBLES
=====
AYUDA: muestra esta pantalla de ayuda.
FICHERO: permite elegir los ficheros de usos y usuarios.
INFORME <nombre-fichero>: escribe en el fichero especificado un informe con el número
de usos de las estaciones según los ficheros seleccionados actualmente
USUARIO <id-usuarios>: informa acerca del número de usos realizados por el usuario
especificado.
ESTADISTICAS: informa sobre el número de usos de hombres y mujeres según el contenido
de los ficheros seleccionados actualmente.
FIN: termina la ejecución de este programa.
```

7.2.3. Órdenes disponibles

A continuación, el programa solicitará interactivamente una de las órdenes disponibles. El usuario deberá escribir una de ellas, junto con el argumento adecuado en el caso de se requiera (caso de las órdenes «INFORME» y «USUARIO»), la ejecutará y volverá a pedir una nueva orden hasta que el usuario escriba la orden «FIN», en cuyo caso el programa terminará.

Será indiferente que el usuario escriba la orden en mayúsculas, minúsculas o cualquier combinación de ambas.

Se describen a continuación con más detalles en qué consisten las distintas órdenes.

Orden «AYUDA»

Cuando el usuario escribe la orden «AYUDA», se muestran en la pantalla las órdenes disponibles, tal y como se ha especificado en la sección 7.2.2:

```
Orden: ayuda

ÓRDENES DISPONIBLES
=====
AYUDA: muestra esta pantalla de ayuda.
FICHERO: permite elegir los ficheros de usos y usuarios.
INFORME <nombre-fichero>: escribe en el fichero especificado un informe con el número
de usos de las estaciones según los ficheros seleccionados actualmente
USUARIO <id-usuarios>: informa acerca del número de usos realizados por el usuario
especificado.
ESTADISTICAS: informa sobre el número de usos de hombres y mujeres según el contenido
de los ficheros seleccionados actualmente.
FIN: termina la ejecución de este programa.

Orden:
```

Orden «FICHERO»

La orden «FICHERO» permite modificar la selección de ficheros («usos-16.csv» con «usuarios-16.csv» por una parte, y «usos-17.csv» con «usuarios-17.csv» por otra) que el usuario seleccionó inicialmente. El programa se comportará como se especificó en la sección 7.2.1. La orden no se completará hasta que el usuario seleccione una opción correcta.

```
Orden: fichero
Elección de ficheros de usos y usuarios. Opciones disponibles:
16: octubre 2016 a marzo 2017
17: marzo 2017 a agosto 2017
Introduzca una opción: 16
El fichero "../Datos/Bizi/usos-16.csv" existe y contiene 1024820 utilizaciones.
El fichero "../Datos/Bizi/usuarios-16.csv" existe y contiene 20751 usuarios.

Orden:
```

Orden «INFORME»

Cuando el usuario escribe esta opción, debe proporcionar como argumento el nombre del fichero en el que se va a escribir la información acerca de las estaciones Bizi más usadas de acuerdo con el fichero de usos que se encuentre seleccionado en ese momento, ordenadas de mayor a menor número de usos y con el mismo formato que se estableció en el enunciado de la práctica 6. Adicionalmente, el programa mostrará las primeras diez líneas del fichero correspondiente al informe, de forma que muestre las ocho estaciones más utilizadas.

Si los ficheros seleccionados cuando se escribe esta opción fuesen los correspondientes al periodo entre octubre de 2016 y marzo de 2017 (ficheros «usos-16.csv» y «usuarios-16.csv»), el programa mostraría por la pantalla la siguiente información:

```
Orden: informe estaciones16.txt
Puesto  Usos  Id Nombre
-----
 1   47064  16 Plaza España
 2   42306  67 Avda. G. Gómez de Avellaneda - C/ Clara Campoamor
 3   40251  47 Plaza San Francisco - Universidad
 4   39173  34 Plaza Magdalena
 5   36264  10 Pº Echegaray y Caballero - Puente de Santiago
 6   31200  46 Avda. Goya - C/ Baltasar Gracián
 7   30686  45 C/ Gran Vía - Facultad Económicas
 8   28914  42 Avda. Tenor Fleta - C/ Uncastillo
```

El programa, además, habrá generado en el directorio en el que se está ejecutando el programa, un fichero denominado «estaciones16.txt» en el que figurarán todas las estaciones Bizi ordenadas de mayor a menor número de usos según el contenido del fichero actualmente seleccionado («usos-16.csv» en el ejemplo mostrado).

Orden «USUARIO»

Cuando el usuario escribe esta opción, debe proporcionar como argumento el identificador de un usuario del sistema Bizi. El programa informará acerca del número de utilizaciones del sistema que dicho usuario haya realizado de acuerdo con el fichero de usos seleccionado actualmente.

Si los ficheros seleccionados cuando se escribe esta opción fuesen los correspondientes al periodo entre octubre de 2016 y marzo de 2017 (ficheros «usos-16.csv» y «usuarios-16.csv»), el programa mostraría por la pantalla la siguiente información:

```
Orden: usuario 55381  
El/la usuario/a 55381 ha realizado 18 viajes.  
  
Orden:
```

Orden «ESTADISTICAS»

Cuando el usuario escribe esta opción, el programa informa del número de utilizaciones del sistema que han realizado hombres y mujeres de acuerdo con el fichero de usos seleccionado actualmente.

Si los ficheros seleccionados cuando se escribe esta opción fuesen los correspondientes al periodo entre octubre de 2016 y marzo de 2017 (ficheros «usos-16.csv» y «usuarios-16.csv»), el programa mostraría por la pantalla la siguiente información:

```
Orden: estadisticas  
Usos de Bizi por hombres: 635513  
Usos de Bizi por mujeres: 389307  
  
Orden:
```

Orden «FIN»

Cuando el usuario selecciona esta opción, el programa termina (y, por lo tanto, no vuelve a solicitar más órdenes):

```
Orden: fin
```

7.2.4. Órdenes no válidas

Cuando el usuario seleccione una orden no disponible, deberá informar sobre el error y solicitar una nueva orden:

```
Orden: acabar  
Orden "ACABAR" desconocida.  
  
Orden:
```

7.3. Ficheros de usuarios

Los dos ficheros de usuarios ofrecidos por el portal de datos abiertos del Ayuntamiento de Zaragoza responden a la siguiente sintaxis:

```
<fichero-usuarios> ::= <cabecera> { <usuario> }  
<cabecera> ::= "IDUsuario;Genero;Edad" fin_línea  
<usuario> ::= <id-usuario> <separador> <sexo> <separador> <rango-edad> fin_línea  
<id-usuario> ::= literal-entero  
<separador> ::= ";"  
<sexo> ::= "M" | "F"  
<rango-edad> ::= literal-cadena
```

Cada fichero comienza con una línea de cabecera a la que le sigue la información sobre los usuarios del sistema Bizi en un determinado periodo de tiempo. Estos usuarios figuran cada uno en una línea distinta y contienen información sobre su identificador, su sexo (masculino –‘M’– o femenino –‘F’–) y sobre su rango de edad. En este trabajo no se va a utilizar la información acerca del rango de edad al que pertenece cada usuario, por lo que se va a considerar que este último dado es una mera cadena de caracteres que puede ser ignorada cuando se lea el contenido del fichero.

Los dos ficheros proporcionados por el portal de datos abiertos del Ayuntamiento de Zaragoza están ordenados por identificadores de usuario crecientes.

7.4. Diseño e implementación del programa

El programa correspondiente a este trabajo debe desarrollarse en un área de trabajo denominada «Trabajo 2018-19», ubicada en el directorio «Programacion1/Trabajo2018-19». Dentro de la misma, deberá crearse un proyecto Code::Blocks denominado «Trabajo», que deberá estar ubicado en el directorio «Programacion1/Trabajo2018-19/Trabajo».

Debe reutilizarse todo el código que sea posible de los módulos bizi-usos y bizi-estacion desarrollados en la práctica 6, que deberán añadirse al proyecto correspondiente a este trabajo. A dichos módulos se les pueden añadir todas aquellas funciones relacionadas con la gestión de datos de tipo UsoBizi o Estacion que se identifiquen de forma adicional durante el desarrollo del programa.

Adicionalmente, se sugiere crear en el directorio «Programacion1/Biblioteca» un módulo denominado bizi-usuarios en el que declarar un tipo registro denominado Usuario en el que representar la información sobre usuarios Bizi con la que se va a trabajar en este trabajo.

En dicho módulo se deben añadir todas aquellas funciones que gestionen datos de tipo Usuario que se identifiquen durante el desarrollo del programa. En particular, se considera que serán muy útiles las siguientes funciones:

- Una función que, dado el nombre de un fichero de usuarios, lea su contenido y almacene sus datos en un vector de datos de tipo Usuario, indicando también cuantos datos de dicho tipo hay en el fichero.
- Una función que, dados un identificador de usuario, un vector de datos de tipo Usuario y el número de componentes ocupadas de dicho vector, busque dicho identificador entre las componentes ocupadas del vector y devuelva bien el dato de tipo Usuario, bien el índice que ocupa el dicho

vector. Dado que los datos del vector estarán previsiblemente ordenados por identificadores de usuario crecientes (se habrán leído de un fichero en el que así lo estaban), una búsqueda dicotómica será capaz de reducir sensiblemente el tiempo de ejecución de la orden «ESTADISTICAS».

En cuanto al módulo principal, se sugiere aplicar un diseño modular en el que cada orden ofrecida por el programa sea implementada por una función específica.

La función `main`, por tanto, puede comenzar invocando en primer lugar a la función que permite seleccionar los ficheros con datos con los que trabajar (como en el caso de que, posteriormente, se seleccione la orden «FICHERO») y, a continuación, a la función que muestra las órdenes disponibles (como en el caso de que, posteriormente, se escriba la orden «AYUDA»). El programa, a continuación, deberá entrar en un bucle en el que se irá solicitando una orden y se ejecutará, hasta que se escriba la orden «FIN».

La función que permita seleccionar los ficheros con los que trabajar inicialmente y la que ejecute la orden «FICHERO» tiene que leer del teclado una opción de las ofrecidas y generar los nombres de los ficheros con los que se va a trabajar, que tendrán la siguiente forma: «`../..../Datos/Bizi/usos-nn.csv`» y «`../..../Datos/Bizi/usuarios-nn.csv`», donde *nn* es la opción elegida por el usuario. La lectura de la opción como una cadena de caracteres facilitará la generación de los nombres de los ficheros involucrados. Comprobar si el fichero de usos existe y cuántos usos almacena puede hacerse fácilmente con la función `contarUsos` del módulo `bizi-usos` de la práctica 6. Para comprobar si el fichero de usuarios existe y cuántos usuarios almacena podría utilizarse la primera de las funciones sugeridas para el módulo `bizi-usuarios`.

7.5. Entrega del trabajo

El trabajo deberá desarrollarse de forma individual y entregarse a través de la plataforma Moodle antes de las 20:00 horas del viernes 18 de enero de 2019.

Deberá entregarse todo el código necesario para compilar correctamente el trabajo. Ello incluye, al menos, los siguientes ficheros:

- «`bizi-estacion.h`»
- «`bizi-estacion.cpp`»
- «`bizi-usos.h`»
- «`bizi-usos.cpp`»
- «`bizi-usuario.h`»
- «`bizi-usuario.cpp`»
- El módulo principal del programa desarrollado

Si en el desarrollo se han utilizado otros módulos definidos por el estudiante, deben entregarse igualmente.

Cada uno de los ficheros entregados estará encabezado por un comentario en el que consten el nombre y apellidos del autor o autora del trabajo.

No se admitirán trabajos presentados mediante otros medios ni presentados fuera de plazo.

Este trabajo debe ser desarrollado por cada estudiante de forma individual. Se hará un análisis de plagios entre trabajos. En caso de identificarse algún caso de plagio, de una parte o de la totalidad del mismo, todos los trabajos implicados serán calificados con un cero.

Al corregir, se compilará el código de los ficheros entregados. La existencia de errores sintácticos en los mismos supondrá un cero en la calificación del trabajo. Los errores debidos al desarrollo del

módulo principal y de los módulos de biblioteca propuestos en directorios distintos a los indicados en este enunciado y en el de la práctica 6 serán penalizados.

Se comprobará el comportamiento del programa entregado, considerando que ha sido desarrollado en la estructura de directorios indicada en este enunciado y en el de la práctica 6. La no localización de los ficheros de datos en el directorio «Programacion1/Datos/Bizi» será penalizada.

Para calificar el trabajo se tendrán en cuenta los siguientes aspectos:

- Fidelidad del comportamiento del programa a lo especificado y descrito en este documento.
- Calidad del diseño algorítmico del código (datos y funciones), atendiendo los diferentes criterios de calidad explicados en la asignatura (diseño descendente, eficiencia, reusabilidad del código, etc.).
- Presentación y legibilidad del código, teniendo muy presentes las reglas publicadas en la *Guía de estilo para programar en C++* accesible desde la web de la asignatura.