



## Examen práctico

**12:15 a 13:45**

- Tiempo máximo para realizar el trabajo de programación propuesto: 90 minutos
- Entrega del trabajo a través de la plataforma Moodle

Se ha de diseñar un programa cuya temática es la misma que la de los programas desarrollados en la práctica 6 y el trabajo obligatorio cuyos enunciados están disponibles en la sección «Material docente» de la página web de la asignatura<sup>1</sup>. En concreto, se va a trabajar con los rangos de edades de los usuarios del sistema Bizi Zaragoza, tal y como se recogen en los dos ficheros de usuarios ofrecidos por el portal de datos abiertos del Ayuntamiento de Zaragoza. La sintaxis de dichos ficheros se reproduce a continuación, dando más detalle en la regla <rango-edad> con respecto al ofrecido en el enunciado del trabajo obligatorio:

```
<fichero-usuarios> ::= <cabecera> { <usuario> }  
<cabecera> ::= "IDUsuario;Genero;Edad" fin_línea  
<usuario> ::= <id-usr> <separador> <sexo> <separador> <rango-edad> fin_línea  
<id-usr> ::= literal-entero  
<separador> ::= ";"  
<sexo> ::= "M" | "F"  
<rango-edad> ::= "<=25" | "26-35" | "36-50" | "51-65" | ">65"
```

El programa pregunta el nombre de un fichero de usuarios del sistema Bizi Zaragoza, que sigue la sintaxis establecida anteriormente y que debe estar previamente ubicado en el directorio «../../Datos/Bizi/», relativo al directorio del proyecto en el que se desarrolle el programa. El usuario escribe simplemente su nombre, y es responsabilidad del programa realizar las acciones necesarias para acceder al mismo en la ubicación especificada.

A continuación, el programa solicita al usuario una cadena de caracteres que representa un rango de edad, según los literales asociados a la regla <rango-edad>.

Seguidamente, el programa muestra en la pantalla el número de usuarios presentes en el fichero indicado por el usuario pertenecientes al rango de edad suministrado por el usuario.

Se muestran a continuación varios ejemplos de ejecución del programa solicitado, independientes los unos de los otros:

```
Escriba el nombre de un fichero de usuarios del sistema Bizi: usuarios-16.csv  
Escriba el rango de edad: <=25  
Número de usuarios del rango "<=25": 2564
```

```
Escriba el nombre de un fichero de usuarios del sistema Bizi: usuarios-16.csv  
Escriba el rango de edad: 26-35  
Número de usuarios del rango "26-35": 3976
```

<sup>1</sup> [http://webdiis.unizar.es/asignaturas/PROG1/?page\\_id=18](http://webdiis.unizar.es/asignaturas/PROG1/?page_id=18)



Escriba el nombre de un fichero de usuarios del sistema Bizi: usuarios-16.csv  
Escriba el rango de edad: >65  
Número de usuarios del rango ">65": 846

Escriba el nombre de un fichero de usuarios del sistema Bizi: usuarios-17.csv  
Escriba el rango de edad: 36-50  
Número de usuarios del rango "36-50": 7723

Escriba el nombre de un fichero de usuarios del sistema Bizi: usuarios-17.csv  
Escriba el rango de edad: 51-65  
Número de usuarios del rango "51-65": 5510

Cuando se introduzca el nombre de un fichero de usos que no exista, el programa mostrará un mensaje de error en la pantalla:

Escriba el nombre de un fichero de usuarios del sistema Bizi: usuarios-19.csv  
Escriba el rango de edad: 51-65  
No se ha podido leer del fichero "../Datos/Bizi/usuarios-19.csv".

Si el rango de edad especificado no es correcto (no es uno de los literales permitidos por la regla <rango-edad>), el programa se limita a indicar que hay 0 usuarios de ese rango de edad.

Escriba el nombre de un fichero de usuarios del sistema Bizi: usuarios-16.csv  
Escriba el rango de edad: jovenes  
Número de usuarios del rango "jovenes": 0

El programa deberá desarrollarse en un único fichero de código fuente denominado «practico.cpp», que contendrá la función main del programa solicitado y todas aquellas funciones auxiliares que se estimen pertinentes. Puede utilizarse código desarrollado para la práctica 6 y el trabajo de la asignatura, pero deberá incluirse obligatoriamente en el fichero «practico.cpp».

## Presentación del trabajo y criterios de evaluación

Cada estudiante presentará, antes de hora que se indique al comienzo del examen, a través de la plataforma Moodle<sup>2</sup> el fichero con el código fuente «practico.cpp». El contenido del fichero vendrá encabezado por un comentario con el nombre y apellidos del estudiante.

En este examen práctico **se valorará esencialmente que el programa pueda ser compilado sin errores y que, al ser ejecutado, proporcione los resultados esperados.**

También se valorará el diseño del programa, la eficiencia de la solución presentada, la adecuada especificación de cada una de las funciones que integren la solución y la legibilidad del código atendiendo a los criterios de la *Guía de estilo para programar en C++* publicada en la web de la asignatura.

---

<sup>2</sup> <https://moodle2.unizar.es/>

## Solución

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cstring>

using namespace std;

const int MAX_LONG_CADENA = 250;
const char PREFIJO_NOMBRE_FICHERO[] = "../../Datos/Bizi/";

/*
 * Pre: «nombreRelativo» es una cadena de caracteres con tamaño suficiente
 * para almacenar la concatenación de «PREFIJO_NOMBRE_FICHERO» y
 * «nombreFichero»
 * Post: «nombreRelativo» almacena la concatenación de
 * «PREFIJO_NOMBRE_FICHERO» y «nombreFichero».
 */
void crearNombreFicheroRelativo(const char nombreFichero[],
                                char nombreRelativo[]) {
    strcpy(nombreRelativo, PREFIJO_NOMBRE_FICHERO);
    strcat(nombreRelativo, nombreFichero);
}

/*
 * Pre: La cadena de caracteres «nombreFicheroUsuarios» representa la ruta de
 * acceso y el nombre de un fichero de texto con el formato de usuarios
 * del sistema Bizi establecido en el enunciado. El vector «rangoEdad»
 * es una cadena de caracteres acabada en el carácter nulo.
 * Post: Si no ha habido problemas de lectura del fichero «rutaFichero», la
 * función ha devuelto «true» tras haber asignado a cada componente del
 * vector «numUsuariosPorRangoEdad» (según la equivalencia mostrada en
 * la postcondición de la función «indiceRangoEdad») el número de
 * usuarios de cada rango de edad presentes en el fichero de nombre
 * «nombreFicheroUsuarios». En caso contrario, se ha limitado a devolver
 * «false».
 */
int contarRangosEdadUsuarios(const char nombreFicheroUsuarios[],
                              const char rangoEdad[]) {
    ifstream f(nombreFicheroUsuarios);
```

```

if (f.is_open()) {
    int numUsuarios = 0;

    f.ignore(MAX_LONG_CADENA, '\n'); // Línea de cabecera

    int idUsuario;
    f >> idUsuario;
    while (!f.eof()) {
        f.ignore(); // ';'
        f.ignore(); // sexo
        f.ignore(); // ';'
        char rangoEdadLeido[MAX_LONG_CADENA];
        f.getline(rangoEdadLeido, MAX_LONG_CADENA);
        if (strcmp(rangoEdadLeido, rangoEdad) == 0) {
            numUsuarios++;
        }
        f >> idUsuario;
    }
    f.close();
    return numUsuarios;
}
else {
    return -1;
}
}

/*
* Pre: ---
* Post: Se ha solicitado al usuario el nombre de un fichero de texto que
* responde sintácticamente <fichero-usuarios> establecida en el
* enunciado y que se encuentra en el directorio ".././Datos/Bizi/"
* relativo al directorio del proyecto Code::Blocks en el que se
* desarrolló el programa. A continuación ha solicitado al usuario una
* cadena de caracteres que representa un rango de edad. Finalmente, si
* el fichero especificado existe y se puede leer, el programa ha
* mostrado en la pantalla el número de usuarios presentes en el fichero
* indicado que pertenecen al rango de edad seleccionado por el usuario,
* de acuerdo con el siguiente ejemplo de formato:
*
* Número de usuarios del rango "<=25": 2564
*
* En caso de que el fichero no exista o no se pueda leer, ha informado
* del error escribiendo un mensaje en la pantalla.

```

```

*/
int main() {
    cout << "Escriba_el_nombre_de_un_fichero_de_usuarios_del_sistema_Bizi:_\n"
          << flush;
    char nombreFichero[MAX_LONG_CADENA];
    cin >> nombreFichero;
    char rutaRelativa[MAX_LONG_CADENA];
    crearNombreFicheroRelativo(nombreFichero, rutaRelativa);

    cout << "Escriba_el_rango_de_edad:_\n" << flush;
    char rangoEdad[MAX_LONG_CADENA];
    cin >> rangoEdad;

    int numUsuarios = contarRangosEdadUsuarios(rutaRelativa, rangoEdad);
    if (numUsuarios >= 0) {
        cout << "Número_de_usuarios_del_rango_\n" << rangoEdad << "\n:_\n"
              << numUsuarios << endl;
        return 0;
    }
    else {
        cerr << "No_se_ha_podido_leer_del_fichero_\n" << rutaRelativa
              << "\n." << endl;
        return 1;
    }
}

```