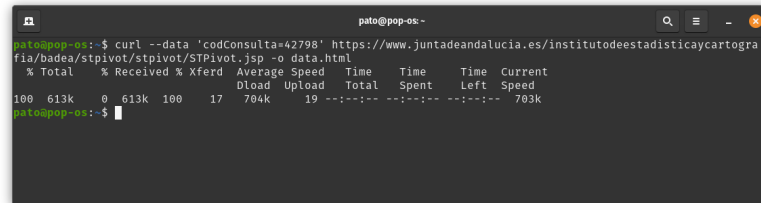




## 2. Procesamiento de la tabla

### 2.1. Obtención de la tabla

Como hemos indicado anteriormente, la tabla es constantemente actualizada por la Junta de Andalucía. Ejecutando la orden *curl* sobre la página web, podremos obtener un fichero html, ya que es la respuesta a una consulta query.



```
pato@pop-os:~$ curl --data 'codConsulta=42798' https://www.juntadeandalucia.es/institutodeestadisticaycartografia/badea/stpivot/stpivot/STPivot.jsp -o data.html
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 613k 0 613k 100 17 704k 19 --:--:-- --:--:-- --:--:-- 703k
pato@pop-os:~$
```

Figura 2: Orden curl ejecutada en un terminal.

### 2.2. Municipios y datos

Una vez tengamos la tabla en html, tendremos que estudiar el formato. De esta manera podremos encontrar las distintas localidades y los datos de la pandemia.

Habiendo encontrado donde están los distritos sanitarios, ya tendremos como está organizada la información que aparece en esa misma fila.



```
<row-heading colspan="1" indent="3" rowspan="1" style="odd">
  <caption caption="San Fernando"/>
</row-heading>
<cell style="odd" val="95001.0" value="95.001"/>
<cell style="odd" val="9599.0" value="9.599"/>
<cell style="odd" val="10104.104167324555" value="10.104,1"/>
<cell style="odd" val="1076.0" value="1.076"/>
<cell style="odd" val="1132.6196566351932" value="1.132,6"/>
<cell style="odd" val="567.0" value="567"/>
<cell style="odd" val="596.8358227808128" value="596,8"/>
<cell style="odd" val="9646.0" value="9.646"/>
<cell style="odd" val="10153.57733076494" value="10.153,6"/>
<cell style="odd" val="8231.0" value="8.231"/>
<cell style="odd" val="110.0" value="110"/>
</row>
```

Figura 3: Foto de como es el formato de cada distrito.

Vemos que el nombre del distrito aparece con el siguiente formato *caption caption="Nombre del Distrito"* y luego los valores de cada una de las columnas aparecen de la forma *val="numero"*. Cabe destacar un tipo concreto de nombre que vamos a evitar a lo largo del trabajo. Este será el de *Municipios de...* ya que tiene columnas vacías y no aporta información importante.

Municipios de Cádiz sin especificar	1.796	-	28	-	4	-	1.800	-	1.716	26
-------------------------------------	-------	---	----	---	---	---	-------	---	-------	----

Figura 4: Ejemplo de este último tipo de distrito.

### 3. Explicación del programa

#### 3.1. Sección de Declaraciones

Lo primero que encontraremos en nuestro archivo flex será la inclusión de todas las bibliotecas necesarias. Tras ella se encuentra la sección de declaraciones, que tiene la siguiente forma:

```
using namespace std;

vector<string> Lugar;
vector<string> Poblacion;
vector<string> Confirmados_PDIA;
vector<string> Confirmados_PDIA_14dias;
vector<string> Tasa_PDIA_14dias;
vector<string> Confirmados_PDIA_7dias;
vector<string> Tasa_PDIA_7dias;
vector<string> Total_Confirmados;
vector<string> Curados;
vector<string> Fallecidos;

string cadena;
ifstream yyin;
int cont = 0;
int aux = 0;
bool es_municipio = false;
```

Figura 5: Sección de declaraciones de nuestro programa.

Como podremos observar hay varias declaraciones de vectores tipo *string* donde dinámicamente se almacenarán, en un futuro, los datos que indican en su propio nombre.

#### 3.2. Sección de Reglas

Tras haber declarado todas las variables necesarias, nos encontraremos con la sección de reglas donde encontraremos los distintos acrónimos que usaremos para encontrar las expresiones regulares indicadas.

```
digito      [0-9]
mayus       [A-Z]
letra       [a-z]
tilde       .
numero      ({digito}+)
lugar       (({mayus}+|{tilde}){letra}*{tilde}*{letra}*)
lugarespa   ({lugar}\b{lugar})
sitio       ({lugar}|{lugarespa}+)
simbolo     [\<\>\|]
salto       [\t\n]+
exclusion    [^\t\n]+
TarLug      ("<caption caption=")
TarFin      (">")
TarNum      ("val=")
Punto       (",")
comilla     ("")
```

Figura 6: Sección de reglas de nuestro programa.

Entre ellas debemos destacar:

- **tilde.** Las tildes vienen en la tabla codificadas con símbolos extraños y distintos (a pesar de estar en el formato UTF-8). Así que para nuestra facilidad buscaremos cualquier carácter con la expresión regular ".".
- **lugar.** Es una palabra que empieza por mayúscula o por tilde, seguida de una concatenación de letras, pudiendo estar intercaladas o finalizar en tilde.

- **lugarespa** y **sitio**. Nos ayudarán a encontrar posibles concatenaciones de lugares. Ya sean con un único espacio, como San Fernando, o varios, como Conil de la Frontera.
- Aquellas que empiezan por **Tar**. Como hemos comentado cuando hablamos del formato de la tabla, estas expresiones regulares nos ayudarán a encontrar los datos necesarios.

### 3.3. Almacenamiento de los datos

Lo siguiente que podremos observar en el código es la forma que tenemos de encontrar y almacenar los distintos distritos. Como bien sabemos, podremos encontrar los lugares en el documento html de la siguiente forma: *caption caption="Poblacion"*. De esta forma, buscaremos cadenas de con ese formato, y ahora lo que debemos hacer es quedarnos únicamente con la población. Para ello, localizamos el '=' y para quedarnos con el recortado, será la subcadena que empieza dos posiciones más allá del '=' ya que evitamos el propio símbolo y las comillas, hasta el final de la cadena eliminando los dos símbolos ya comentados y los últimos tres símbolos, que equivaldrían a *TarFin*.

Además, como ya comentamos en una sección anterior, en el propio bucle comprobaremos si la cadena 'Municipios de' está incluida, en caso de estarlo, decidimos no incluirla en nuestro vector.

```
{TarLug}{comilla}{sitio}{comilla}{TarFin} {
    if(aux>=13){
        cadena = string(VYText());
        size_t first = cadena.find_last_of("=");
        string recortado = cadena.substr(first+2,cadena.size()-first-2-3);
        if(recortado.find(string("Municipios de"))!=string::npos)
            es_municipio = true;
        else
            es_municipio = false;
        if(!es_municipio)
            Lugar.push_back(recortado);
    }
    aux++;
}
```

Figura 7: Almacenamiento de las poblaciones.

Ahora veremos como insertar los distintos valores. Como vimos anteriormente, estos datos se encuentran en el fichero html tras la expresión *"val="*. Entonces, de la misma forma, nos volvemos a quedar con la subcadena que nos interesa eliminando caracteres delante y detrás de la cadena leída con expresiones regulares. Cabe destacar que existen tantos datos como columnas tenga la tabla. Por tanto, con un *switch* y un contador que indique la columna en la que estemos situados, podremos almacenar cada dato en su correspondiente vector. Cabe mencionar que el caso en el que el contador sea igual a 3, ese dato no nos interesa así que lo evitaremos.

### 3.4. Función main

Lo primero que haremos en el main será primordial. Intentaremos abrir el fichero *data.html* donde nosotros hemos almacenado nuestra tabla en formato html. De esta forma, tenemos dos posibles situaciones: (a) tenerla descargada, en este caso podemos poner nuestro programa a funcionar. (b) no tenerla descargada, en este caso, usaremos la orden *system* para ejecutar el comando *curl* del que hablamos al principio y poder así descargarla.

Tras ello, mostraremos el menú, que dispone de las siguientes opciones:

- Poder mostrar con un índice todas las localidades de las que tenemos datos.
- Mostrar los datos asociados a una localidad en concreto.
- Eliminar el fichero en caso de querer una actualización.
- Salir.

En caso de querer mostrar los datos sobre la COVID-19 de alguna población lo debemos hacer buscando su nombre y escribiendo el índice por pantalla. Esto se debe a que existen localidades con tildes, que se traducen a símbolos extraños, imposibles de introducir por el teclado.

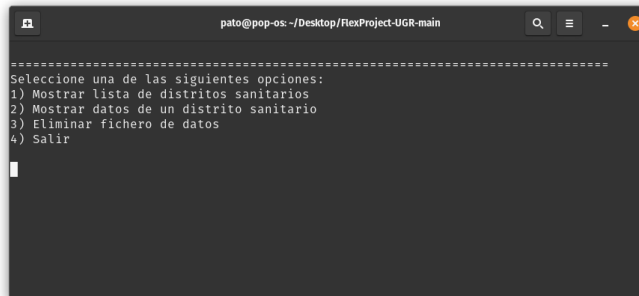
## 4. Ejemplo de funcionamiento

Veamos que una vez listo para ejecutar y sin haber descargado el fichero obtendremos la siguiente salida por pantalla.



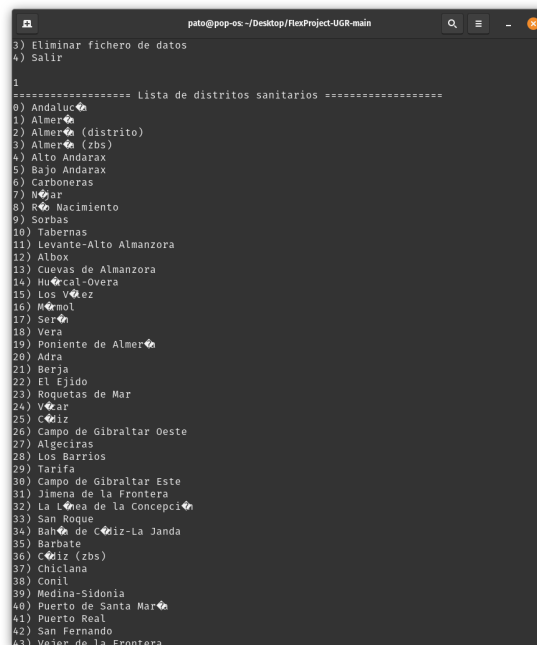
```
pato@pop-os: ~/Desktop/FlexProject-UGR-main
pato@pop-os:~/Desktop/FlexProject-UGR-main$ ./CoronaVirusBase
Fichero data.html NO encontrado
¿Desea descargar el archivo(s/n)?
```

Como sabemos que debemos descargarlo, lo hacemos y ya nos debe aparecer el menú. En este caso



```
pato@pop-os: ~/Desktop/FlexProject-UGR-main
=====
Seleccione una de las siguientes opciones:
1) Mostrar lista de distritos sanitarios
2) Mostrar datos de un distrito sanitario
3) Eliminar fichero de datos
4) Salir
```

elegiremos la primera opción y veremos la lista numerada de distritos sanitarios disponibles.



```
pato@pop-os: ~/Desktop/FlexProject-UGR-main
3) Eliminar fichero de datos
4) Salir
1
===== Lista de distritos sanitarios =====
0) Andalucía
1) Almería
2) Almería (distrito)
3) Almería (zbs)
4) Alto Andarax
5) Bajo Andarax
6) Carboneras
7) Níjar
8) Río Nacimiento
9) Sorbas
10) Tabernas
11) Levante-Alto Almanzora
12) Albox
13) Cuevas de Almanzora
14) Huércal-Overa
15) Los Vélez
16) Múrmol
17) Serón
18) Vera
19) Poniente de Almería
20) Adra
21) Berja
22) El Ejido
23) Roquetas de Mar
24) Vélez
25) Cádiz
26) Campo de Gibraltar Oeste
27) Algeciras
28) Los Barrios
29) Tarifa
30) Campo de Gibraltar Este
31) Jimena de la Frontera
32) La Línea de la Concepción
33) San Roque
34) Bahía de Cádiz-La Janda
35) Barbate
36) Cádiz (zbs)
37) Chiclana
38) Conil
39) Medina-Sidonia
40) Puerto de Santa María
41) Puerto Real
42) San Fernando
43) Vejer de la Frontera
```

Finalmente, elegiremos, por ejemplo, el número 42, en este caso, San Fernando.



```
pato@pop-os: ~/Desktop/flexProject-UGR-main
=====
Seleccione una de las siguientes opciones:
=====
1) Mostrar lista de distritos sanitarios
2) Mostrar datos de un distrito sanitario
3) Eliminar fichero de datos
4) Salir

2

Introduce un distrito: 42
===== Datos Distrito San Fernando =====
Poblacion:      95001.0
Confirmados PDIA:      9457.0
Confirmados PDIA 14 dias:      974.0
Tasa PDIA 14 dias:      1025.2523657645709
Confirmados PDIA 7 dias:      565.0
Tasa PDIA 7 dias:      594.7305817833496
Total Confirmados:      9504.0
Curados:      8221.0
Fallecidos:      110.0
=====
Seleccione una de las siguientes opciones:
```