

**Equipo 3**

**Alumnos:**

[Gonzalo Ronzón Carniado](mailto:0207950@up.edu.mx)

[Raul Unda](mailto:0225484@up.edu.mx)

[Javier Vázquez Gurrola](mailto:0215391@up.edu.mx)

[Cristian Ramos Barragán](mailto:0228894@up.edu.mx)

[Francisco Anaya Viveros](mailto:0181879@up.edu.mx)

**Trabajo: Reporte analizador de canciones**

**Materia: Teorías de lenguajes y programación**

**Profesor: Félix Hernández**

**Fecha de entrega: 23/09/2021**

# Índice

[**Índice**](#_cke0in4diema) **2**

[**Resumen.**](#_mychf3fgvw9) **2**

[**Introducción**](#_n3igvg4aczlq) **3**

[**Contenido.**](#_miwrlvo0qaix) **4**

[Organización de los archivos de entrada](#_l250flwfe61j) 4

[**Solución**](#_wx4it8b4pb4p) **4**

[Declaraciones y definiciones](#_rykqsni9n7is) 4

[Reglas](#_9fl263xbx0dj) 6

[Reglas para la composición de expresiones regulares](#_daquxv74jmmb) 7

[Código adicional](#_lv1o4arc7j9y) 7

[Función Main](#_ejza25x5ri5a) 7

[Función analiza\_palabra\_encontrada](#_j815qwqk7ab2) 8

[Función analiza\_indice](#_wp7jb6ok00k7) 9

[Función ordena\_diccionario](#_4qc2uv5dyfeg) 10

[**Conclusiones.**](#_m6tj98y6whco) **11**

# 

# Resumen.

Se buscó desarrollar un analizador lexicográfico utilizando Flex y el lenguaje C, que fuera capaz de identificar las palabras utilizadas en las letras de 160 canciones con la finalidad de contarlas, agruparlas por artista y género, ordenarlas alfabéticamente, y al final imprimir toda esa información para que el usuario pueda conocer los resultados.

# Introducción

Flex es una herramienta pensada para generar escáneres, es decir, programas que reconozcan patrones léxicos en un texto. Flex es capaz de leer archivos de entrada proporcionados por el usuario, los cuales contienen la información a escanear.

Flex utiliza lo que llamamos Expresiones Regulares, las cuales son secuencias de caracteres que funcionan como modelos para la comparación, coincidencia y modificación de cadenas. Estas expresiones regulares ayudan al escáner a reconocer gramáticas, en otras palabras, gracias a las expresiones regulares el escáner es capaz de identificar lo que está escrito en el documento de entrada y así discriminar dependiendo lo que se esté buscando.

Las condiciones de arranque de Flex son un mecanismo para activar las reglas de manera condicional. En el escáner realizado nuestra condición es nombrada “%x ANALIZADOR”, esto quiere decir que <ANALIZADOR>{PALABRA} sólo será ejecutada cuando el analizador esté en la condición de arranque <ANALIZADOR>.

El objetivo del escáner realizado es identificar, contar, separar, organizar, agrupar y dar a conocer al usuario cómo es que el uso de palabras se utiliza y distribuye según el género musical y el artista. El escáner realizado funciona de la siguiente manera: lee todas las canciones (índice), luego se almacena cada elemento en su diccionario correspondiente, al final generar un archivo csv por cada diccionario.

Es decir, es capaz de contar las palabras y desplegarlas en listas por cada artista que nos dan a conocer la palabra, la cantidad de veces y que artista se está analizando. De igual forma nos desplegará un csv que muestra las palabras y la cantidad de veces dichas en las letras por cada género. El escáner realizado no logró ordenar alfabéticamente debido al tiempo de ejecución que se elevaba considerablemente, así que las palabras las va mostrando conforme las encuentra.

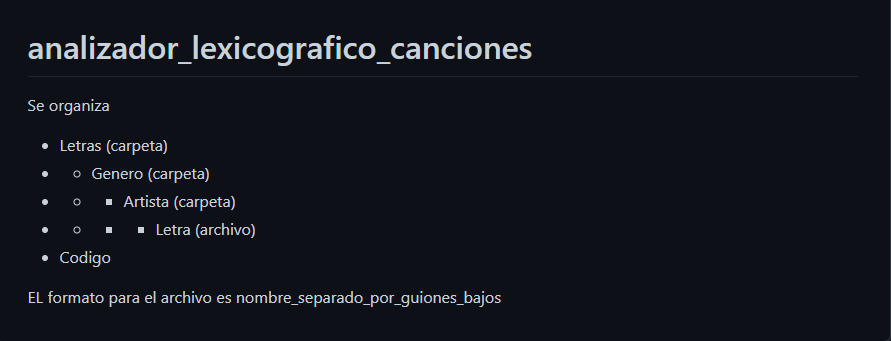
# Contenido.

## Organización de los archivos de entrada

Los archivos de entrada fueron organizados de la siguiente manera:

Existe una carpeta llamada Letras la cual contiene una carpeta por género, es decir 16 carpetas de los distintos géneros. Cada una de esas carpetas contiene tres carpetas, una por cada artista seleccionada a analizar, una vez dentro de las carpetas por artistas encontramos los archivos de entrada que contienen las letras.

Dichos archivos son nombrados con el nombre de la canción separado por guiones bajos. En la imágen se muestra un diagrama sencillo de la organización.



## Solución

### Declaraciones y definiciones

Flex cuenta con un formato específico para el código, el cual es el siguiente:

definiciones

%%

reglas

%%

código de usuario

Debido a esto, en nuestra primer sección declaramos todas las definiciones necesarias para la ejecución del escáner. Vamos a estar utilizando: variables char, como:

* directorio
* archivo\_de\_entrada
* archivo\_de\_entrada1
* archivo\_a\_abrir
* archivo\_a\_abrir1
* genero
* artista
* bandera

Dichas variables tipo char se utilizan más adelante en el código y se explicarán en la sección correspondiente. Igualmente, se declara una estructura que nos ayudará a almacenar los diccionarios de los géneros y artistas. Dicha estructura contiene él índice, la cantidad de veces que se ha encontrado una palabra y la palabra analizada.

Posteriormente, en la sección de reglas comenzamos declarando las tuplas con las cuales va a funcionar nuestro analizador. Es importante mencionar que el nombre asignado al archivo de salida se asigna aquí y se determinó como: ***"analizador\_lex\_canciones.c"***

Dentro de esta sección se declaran las expresiones regulares que se van a utilizar, al igual que nuestra condición de arranque.

En este caso las expresiones regulares son:

* PALABRA [A-ZÑÁÉÍÓÚa-zñáéíóú][A-ZÑÁÉÍÓÚa-zñáéíóú]+
* NO\_CERO [1-9]
* DIGITOS [0-9]
* NUMERO\_ENTERO {NO\_CERO}{DIGITOS}\*
* GUION [\_]
* NUMERO\_ENTERO\_GUION {GUION}{NUMERO\_ENTERO}
* PALABRA\_GUION {GUION}{PALABRA}
* NOMBRE {PALABRA}{PALABRA\_GUION}\*{NUMERO\_ENTERO\_GUION}\*
* NOMBRE\_DIAGONAL {NOMBRE}[/]
* EXTENSION [.]{PALABRA}
* URL {NOMBRE\_DIAGONAL}\*{NOMBRE}{EXTENSION}

### Reglas

Para saber cómo va a actuar el código al encontrar ciertos elementos que coincidan con las expresiones regulares previamente establecidas, es necesario declarar ciertas reglas. Las necesarias para el escáner son:

* {NOMBRE}: cada que los elementos coinciden con la expresión regular NOMBRE, la regla enviará como parámetro la palabra en la entrada de Flex a la función analiza\_indice, para posteriormente incrementar la bandera.
* {URL}: URL es una expresión regular que busca coincidencias con la ruta de los archivos de entrada. La regla busca abrir los archivos de entrada e ir recorriendo todas las letras de cada artista y de cada género.
* <<EOF>>: es una regla necesaria para Flex, para que este sepa qué hacer cuando se ha llegado al final de un archivo de entrada.

## Reglas para la composición de expresiones regulares

### Código adicional

Esta sección también se conoce como código de usuario. Aquí vamos a encontrar el comportamiento de las funciones generadas por el usuario, al igual que el comportamiento main del código.

#### Función Main

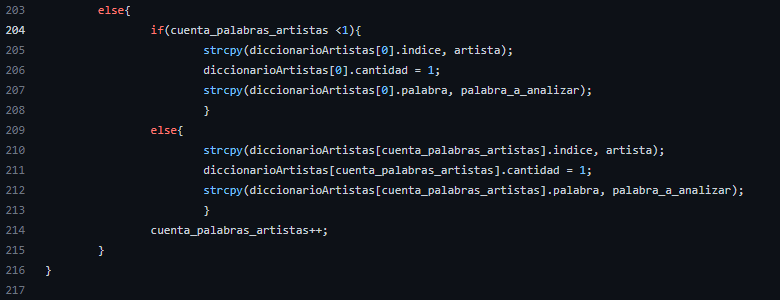
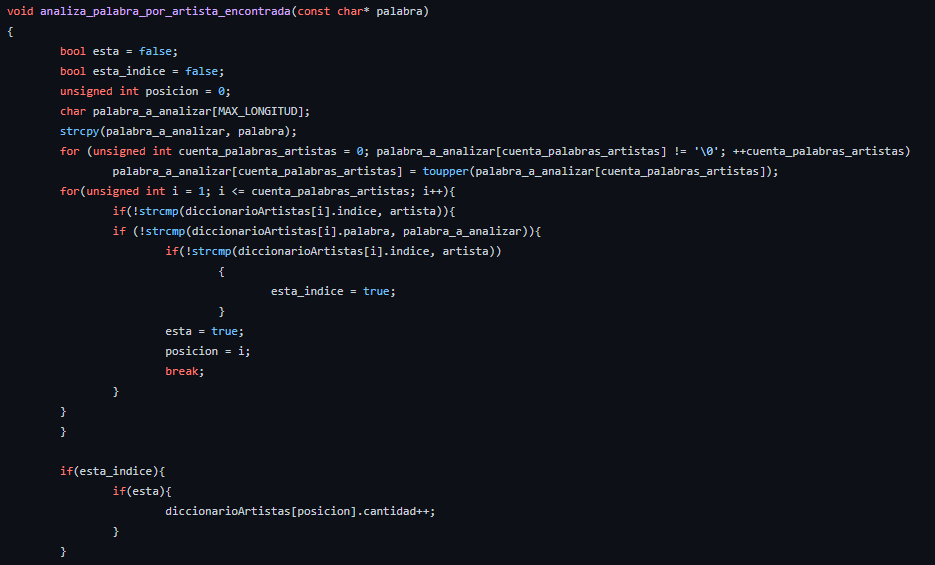
Como se solicitó en el proyecto, solo funcionará si se detectan los archivos en nuestro directorio, evitando que se detecte alguna entrada por teclado.

Definición de variables globales en donde almacenar la información. Se incluye una clase para los arreglos en donde se almacenan los datos obtenidos, mediante la cual se podrá guardar el número de repeticiones de una palabra determinada. Terminado el análisis se imprimirán las palabras dichas por artista y las palabras encontradas en cada género.

#### 

#### Función analiza\_palabra\_encontrada

Analiza la palabra encontrada con nuestro diccionario y si la encuentra le agrega +1 en el contador de palabras analizadas por artista

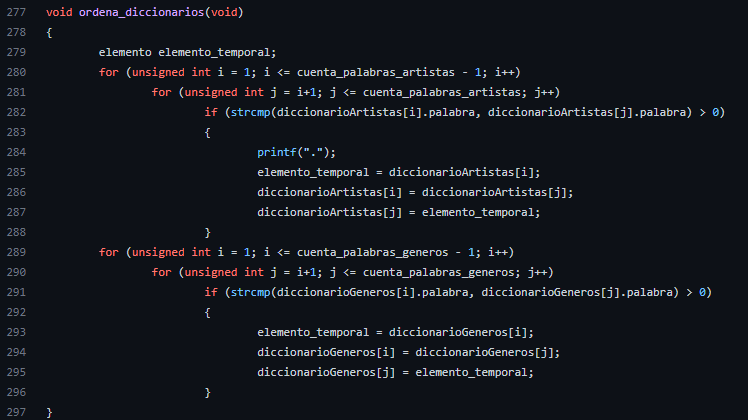


#### Función analiza\_indice

Siendo una función recursiva que cuando vuelve a ser llamada cambia el valor de la bandera que se encarga de seleccionar entre por género y por artista

#### 

Función ordena\_diccionarios

Se recorrerán los diccionarios de artistas y géneros mediante un for anidados que recorre el arreglo para comparar las palabras y utilizar un ordenamiento de burbuja para organizar el diccionario.

# Conclusiones.

Flex resulta una herramienta con amplios usos, en este caso cuyo propósito es analizar la letra de distintas canciones de distintos artistas y géneros nos ayuda a tener una idea de que tipo de letras tienen cada uno de los géneros. En los resultados del análisis se puede ver como las palabras se repiten mayormente por género. Al contrario de lo que pudimos pensar al principio, la mayoría de las palabras que se repiten son palabras como: me, las, no, mi, de. Esta misma idea de escáner se podría implementar en otros casos, quizá como en un detector de plagio o para comparar la escritura entre personas y así saber quién es el autor de que.