**Imagen que contiene firmar, señal, dibujo, alimentos

Descripción generada automáticamente**

**Proyecto fase 2**

Programación 3

Ingeniería en sistemas

Universidad Mariano Gálvez de Guatemala, Sede Portales

9989 21 13256 – Erwin Adolfo Roldán Hernández

9989 21 392 – Javier Alejandro Avalos Galindo

Contenido

[Proyecto fase 1 3](#_Toc136290601)

[Definición de librerías 3](#_Toc136290602)

[Iostream 3](#_Toc136290603)

[Sstream 3](#_Toc136290604)

[Fstream 3](#_Toc136290605)

[Conio.h 3](#_Toc136290606)

[Windows.h 4](#_Toc136290607)

[Cstdio 4](#_Toc136290608)

[MMsystem.h 4](#_Toc136290609)

[Base de datos 4](#_Toc136290610)

[Carga de datos 5](#_Toc136290611)

[Lectura de las palabras almacenadas 6](#_Toc136290612)

[Estructura para recorrer archivo 6](#_Toc136290613)

[Recorrido de la base de datos 8](#_Toc136290614)

[Menú principal 8](#_Toc136290615)

[Traducir 9](#_Toc136290616)

[Reporte 13](#_Toc136290617)

# Proyecto fase 1

Creación de una aplicación en c++ que permita realizar la lectura de un archivo, que contenga registros estructurados de palabras en español y sus traducciones a italiano, francés, alemán e inglés.

El programa debe de poder cargar la información desde un archivo,  
El programa debe de permitir agregar nuevos nodos con las palabras y sus traducciones,  
El programa debe de permitir eliminar nodos del árbol.  
Debe de tener la capacidad de reproducir el audio de las palabras traducidas.

# Definición de librerías

Texto

Descripción generada automáticamente

## Iostream

Se definió para poder utilizar datos de entrada y salida ya que el usuario debe ingresar palabras en español y debe obtener la palabra en la traducción seleccionada.

## Sstream

Se utilizó está librería para poder trabajar flujos de datos en forma de cadenas de caracteres.

## Fstream

La utilizamos para poder realizar operaciones de entrada y salida en archivos ya que las palabras fueron almacenadas en un documento csv.

## Conio.h

Se utilizó para poder leer caracteres del teclado y poder y generar o modificar texto en pantalla.

## Windows.h

Es necesario interactuar con el sistema operativo, esta librería nos da la capacidad de interactuar con la base de datos antes mencionada y también con los archivos de audio de cada una de las palabras.

## Cstdio

Esta librería nos permite interactuar con archivos, como abrirlos, cerrarlos o incluso modificarlos agregando o eliminando datos.

## **MMsystem**.h

Seleccionamos esta librería para poder tener la capacidad de reproducir archivos en nuestro programa ya que se nos solicita reproducir un audio de la palabra traducida.

# Base de datos

Para poder almacenar el listado de palabras realizamos un archivo csv donde se utilizó una columna, en cada celda se ingresó inicialmente la palabra en español seguido de sus traducciones en italiano, inglés, francés y alemán. Cada una de estas palabras está separada por un punto y coma “;” para que el programa pueda interpretar que son datos relacionados más no son el mismo. Se ingresaron 100 palabras en español con sus respectivas traducciones haciendo un total de 500 palabras almacenadas dentro de la base de datos.

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

## Carga de datos

Lo primero que debemos hacer al momento de trabajar con una base de datos es cargar los datos que ya se encuentran en la misma. En este caso nuestra base de datos es un archivo CSV, por lo tanto, debemos crear una función “CargarDatos” que se encargue de obtener todos los datos que tenemos en la misma y guardarlos dentro de la estructura de árboles que a continuación se detallará.

Es importante saber que para cada una de las inserciones dentro de la BD se debe crear un nuevo nodo, el cual será colocado de acuerdo a las propias características del dato que estemos ingresando. La función “nuevoNodo” e “insertarNodo” realizaran estas acciones en nuestra estructura de datos.

Texto

Descripción generada automáticamente

Por ultimo, en nuestro main obtenemos la lista de palabras que tenemos almacenadas en nuestra BD

Texto

Descripción generada automáticamente

## Lectura de las palabras almacenadas

### Estructura para recorrer archivo

Se define una estructura de datos de árbol AVL, los mismo se realiza para que la búsqueda de las palabras sea mas eficiente y no tenga que recorrer todas las posibilidades dentro de las palabras del mismo idioma.

Texto

Descripción generada automáticamente

Se realizan distintas funciones para controlar el árbol y poder trabajarlo como AVL, la creación de nodos y la rotación de estos es importante dentro de esta estructura de datos, debido a que los arboles AVL se caracterizan en la velocidad de búsqueda que tienen. Es una de las estructuras de datos más versátiles para trabajar.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Además, se crean condiciones al momento de realizar la insecion de cualquier nodo dentro del árbol, estas condiciones nos ayudan a determinar si el árbol se encuentra o no balanceado en todos los posibles nodos con sus hijos.

Texto

Descripción generada automáticamente

También, creamos una función para recorrer el árbol como tal.

Texto

Descripción generada automáticamente

### Recorrido de la base de datos

Debido a que ya tenemos las librerías necesarias para poder interactuar con archivos del sistema, abrirlos, leerlos y modificarlos solo debemos crear primero una función para poder ejecutar el archivo el cual se devine como NOMBRE\_ARCHIVO después de declarar las librerías y entre comillas se coloca el nombre del archivo csv.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Luego de abrirlo el siguiente paso es poder recorrerlo por lo que utilizamos ifstream para poder acceder al archivo definido dentro de NOMBRE\_ARCHIVO, después de esto utilizamos un void **CargarDatos** el cual toma como delimitador el punto y coma “;” para separar los datos que se encuentran el archivo, luego se define a que idioma pertenece cada una de estas palabras ingresándola a un string nombrado con cada uno de los idiomas disponibles dentro del traductor.

# Menú principal

Se creo un menú principal en el cual el usuario podrá escoger entre 3 acciones a realizar dentro del programa: traducir, reporte, salir.

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

## Traducir

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico, Sitio web

Descripción generada automáticamenteSi el usuario selecciona la opción traducir utilizaremos la función idioma\_origen donde se debe indicar en que idioma se está ingresando la palabra, luego con la función idioma\_destino el usuario seleccionará a que idioma desea traducir la palabra ingresada.

Luego de hacer la inserción empieza en proceso de búsqueda, primero se hace la validación de cual fue el idioma que seleccionó, luego se utiliza una variable auxiliar llamada encontrado el cual nos ayuda a validar si la palabra existe o no dentro de la base de datos, si esta se encuentra dentro de la base de datos arrojará true, por ultimo se agrega un break para que el programa pueda seguir validando a través de un else if con los demás idiomas disponibles.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Ahora que ya pudimos recorrer los datos y validar si existe o no el siguiente paso es mostrarlos y reproducir el audio por lo cual de la variable “encontrado” el cual si arroja un true realizará un proceso en el cual primero se evalúa en que idioma fue encontrado, luego un cout mostrando la palabra en el idioma solicitado por el usuario y por último se ejecuta el string ruta el cual mandará a llamar el audio de la palabra a través de su localización en la carpeta del sistema operativo seguido del formato del mismo, en este caso wav, y de esta forma es como logramos mostrar y reproducir la palabra traducida.

Escala de tiempo

Descripción generada automáticamente con confianza baja

En caso de que encontrado sea igual a false simplemente se mostrará un mensaje en pantalla indicando que la palabra no existe.

Texto

Descripción generada automáticamente

## Reporte

En el caso de que el cliente solicite un reporte de las palabras existentes se generara un recorrido a través de un for ya que conocemos la cantidad de datos que están manejando y a través de un cout mostraremos las palabras que están en cada idioma.

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media