

**UNIVERSIDAD RAFAEL LANDÍVAR DE  
GUATEMALA**

**FACULTAD DE INGENIERIA EN  
INFORMATICA Y SISTEMAS**

**CURSO: ESTRUCTURA DE DATOS I**  
**ING. PABLO GODOY**



**Universidad  
Rafael Landívar**  
Tradición Jesuita en Guatemala

# **ANÁLISIS Y DISEÑO DE SOFTWARE**

## **PROYECTO “MICROSQL”**

### **INTEGRANTES**

SEBASTIAN ORANTES CHANG	-	1085816
BRYAN MACARIO CORONADO	-	1283816

GUATEMALA, 05 DE ABRIL DEL 2017

# ÍNDICE

## Definición General

• 15	2
• Idea general	2
• Funcionalidad el software	2
• Funcionamiento interno	3
• Propósitos y objetivos	5

## Arquitectura del Sistema

• Diagramas de Casos	6
• Diagramas de Flujo	8
• Diagrama de Clase	10

Anexo	12
-------	----

# DEFINICIÓN GENERAL

## ¿En qué consiste el proyecto?

El proyecto consiste en desarrollar un manejador de base de datos (DBMS) y recibe el nombre de “micro SQL”, y tiene como características principales:

- Gestionar estructuras de datos de forma personalizada.
- Permitir que el manejador se pueda acoplar a varios idiomas distintos del inglés.

La idea sobre un manejador de base de datos es muy amplia, por lo que en el anexo del documento se encuentra una breve explicación sobre lo ¿qué es una base de datos?, ¿qué es SQL? y se adjuntó información del funcionamiento de un manejador ya desarrollado (MySQL).

## Idea General

Crear un programa que pueda atender las funcionalidades básicas y requeridas de un manejador de base de datos, además que posea una interfaz gráfica amigable para que la experiencia del usuario sea agradable, para ello se utilizará el lenguaje de programación C# desarrollando el proyecto en el entorno de Visual Studio, además de aplicar las estructuras de datos desarrolladas durante el curso (árboles), para obtener como producto final un instalador del manejador “Micro SQL”.

## Funcionalidad del software

### **1. Interfaz gráfica**

El programa deberá de mostrar una interfaz gráfica con tres secciones:

1. Una primera sección donde se pueda introducir código SQL para manipular los objetos de la base de datos.
2. La segunda sección deberá incluir un “grid” donde pueda visualizarse el resultado de las instrucciones de recuperación de datos ejecutadas
3. La tercera sección tendrá la función de visualizar de forma atractiva el listado de tablas creadas y sus respectivas columnas.

En la sección de código SQL, se pueden aplicar varias instrucciones en una sola ejecución, siempre y cuando estas instrucciones se encuentren separadas por una línea con la palabra GO.

Micro SQL deberá permitir la gestión de la base de datos tomando en cuenta los siguientes puntos:

- Debido a que el programa debe aceptar comandos en varios idiomas, se pueden sustituir las palabras reservadas del lenguaje SQL, por lo que al iniciar el programa se debe cargar un diccionario de palabras reservadas desde el archivo C:/microSQL/microSQL.ini. Dicho archivo consiste en un archivo CSV que contiene por cada línea una pareja de palabras de la siguiente forma:

**<Palabra Reservada>, <Comando en otro idioma>**

El usuario deberá tener la oportunidad de personalizar este diccionario, puede hacerlo mediante la sustitución del archivo por uno nuevo que contenga la nueva definición. Si no existiera dicho archivo, el sistema deberá generar uno nuevo colocando los valores por defecto.

Así mismo la configuración de la aplicación también debe permitir la carga de una nueva definición del diccionario desde la interfaz gráfica, así como una opción que permita restablecer a los valores predeterminados.

**IMPORTANTE:** Las palabras reservadas deberán ser mostradas en un color distinto en la interfaz gráfica en la sección para introducción de código SQL.

## 2. Funcionamiento interno

El sistema debe permitir las siguientes funcionalidades:

- **Creación de nueva tabla:**

Permite la creación de una nueva tabla en la base de datos (internamente será un árbol B).

- ❖ Los nombres de las columnas no pueden contener espacios.
- ❖ Los tipos de columnas admitidas son: INT, VARCHAR(100) y DATETIME.
- ❖ El máximo número de columnas por cada tipo de dato es 3.
- ❖ Se deberá agregar obligatoriamente una columna de nombre ID de tipo INT que actuará como llave primaria.
- ❖ Se debe validar que la sintaxis cumpla con todas las reglas mencionadas, de lo contrario deberá mostrarse un error en pantalla.

- **Insertión de datos:**

- ❖ Permite el ingreso de datos a una tabla específica de la base de datos.
- ❖ Se debe validar que la sintaxis cumpla con todas las reglas establecidas, en caso no se cumpla debe generar error de sintaxis.
- ❖ Las columnas/datos para la inserción siempre se proporcionarán en el mismo orden en que fueron indicados en la creación de la tabla.

- ❖ Se proporcionarán valores para todos los campos ya que no se permiten campos “Nulos”.
- ❖ Los valores de campos VARCHAR deberán delimitarse por medio del símbolo ‘(apóstrofo o comilla simple)’.

- **Selección de datos:**

- ❖ Permite la extracción de datos de la Base de Datos, para una tabla específica.
- ❖ Los datos extraídos deben mostrarse en un “grid”.
- ❖ El contenido del “grid” se deben poder exportar a un archivo CSV escogido por el usuario.
- ❖ Se debe validar que la sintaxis cumpla con todas las reglas establecidas, de lo contrario deberá mostrarse un error en pantalla.
- ❖ Si los campos indicados en la consulta no existen deberá mostrarse un error en pantalla indicando cuales son los campos inexistentes en la tabla.

- **Eliminación de datos:**

- ❖ Permite eliminar datos almacenados en una tabla (eliminar filas de la tabla).
- ❖ Se debe validar que la sintaxis cumpla con todas las reglas establecidas, de lo contrario deberá mostrarse un error en pantalla.

- **Eliminación de tablas:**

- ❖ Elimina toda la tabla y sus datos, se deberán borrar todos los archivos asociados.
- ❖ Se debe validar que la sintaxis cumpla con todas las reglas establecidas, de lo contrario deberá mostrarse un error en pantalla.

## **Propósitos y Objetivos**

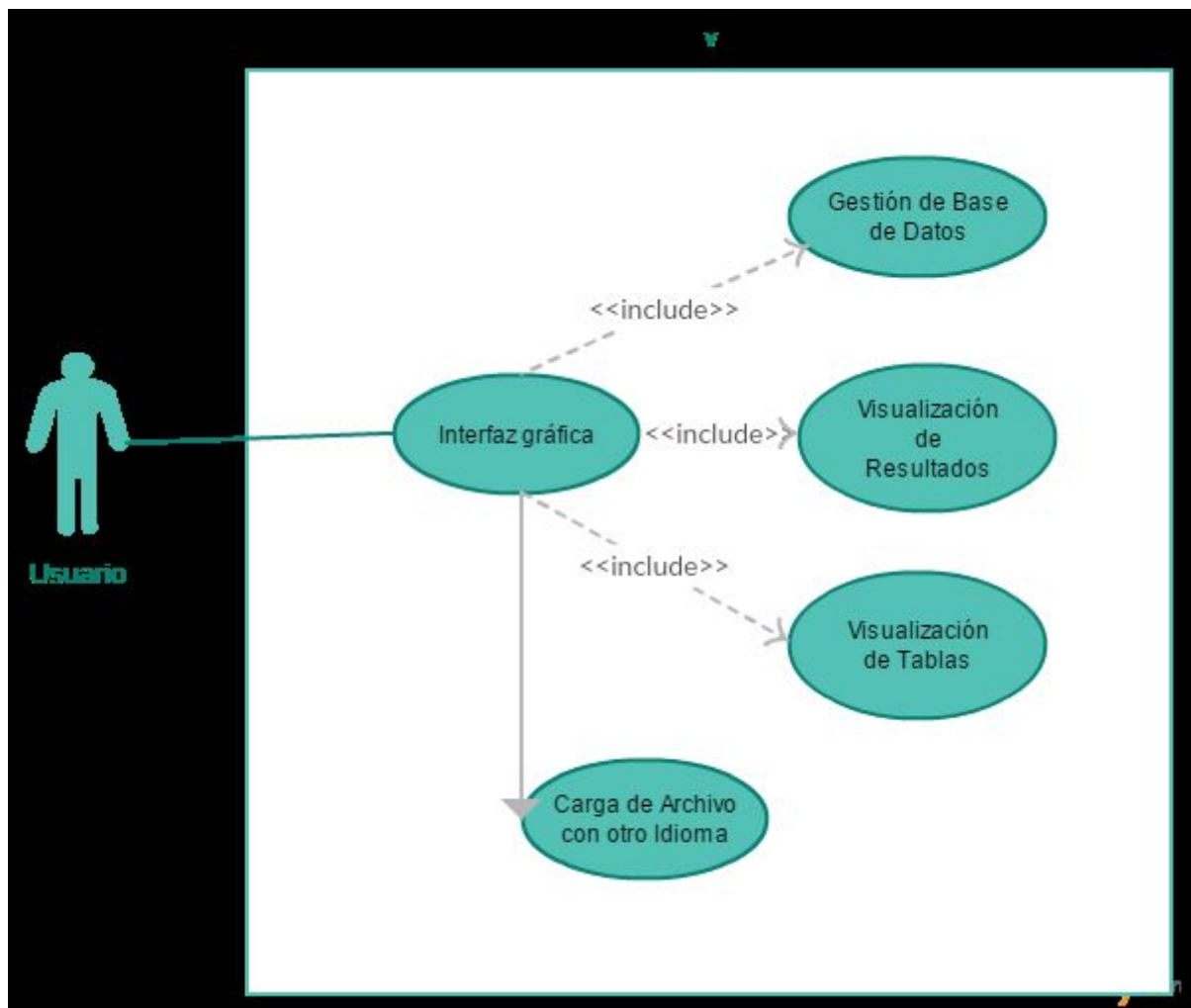
- Aplicar los conocimientos adquiridos durante el curso para el desarrollo de una aplicación de software.
- Codificar soluciones para un problema dado por medio de un enfoque Orientado a Objetos.
- Creación y manipulación de distintas estructuras de datos en C#.
- Aplicación de las mejores prácticas de desarrollo vistas en clase.
- Desarrollar el trabajo en equipo.
- Investigar sobre el tema de base de datos.

# ARQUITECTURA DEL SISTEMA

Para el desarrollo de la aplicación, se utilizará el lenguaje de programación C#, porque es un lenguaje con el cual el desarrollador está familiarizado y además se utilizará un dll que se ha estado desarrollando a lo largo del curso que posee diferentes estructuras de datos realizados en C#. Por otro lado se escogió C# por su paradigma que es POO y permite realizar una abstracción más fácil del problema.

La arquitectura básica de la aplicación es la que se presenta en el diagrama de casos de usos 1 y 2.

## Manejador de base de datos - Casos de uso 1

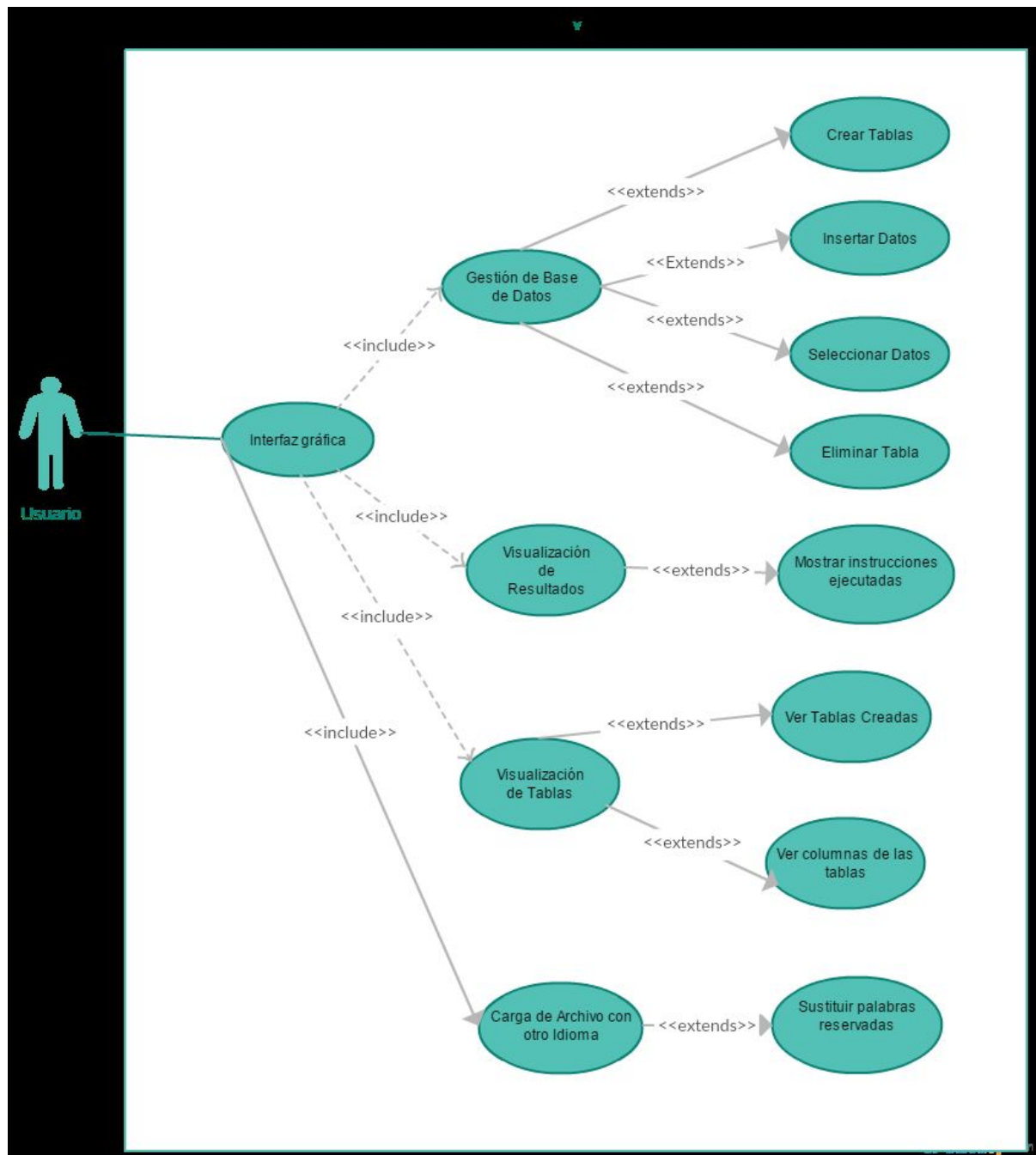


### Descripción:

El usuario interactúa con la interfaz de la aplicación, que le permite acceder a la función de gestionar base de datos, hace referencia a que el usuario puede introducir código SQL para generar acciones de parte de la aplicación. Visualizar los resultados que vaya generando el usuario de manera atractiva y poder visualizar las tablas que se vayan

creando. El primer paso antes de acceder a la interfaz gráfica es la carga del lenguaje que se trabajará por default se inicia con el idioma inglés.

## Manejador de base de datos - Casos de uso 2



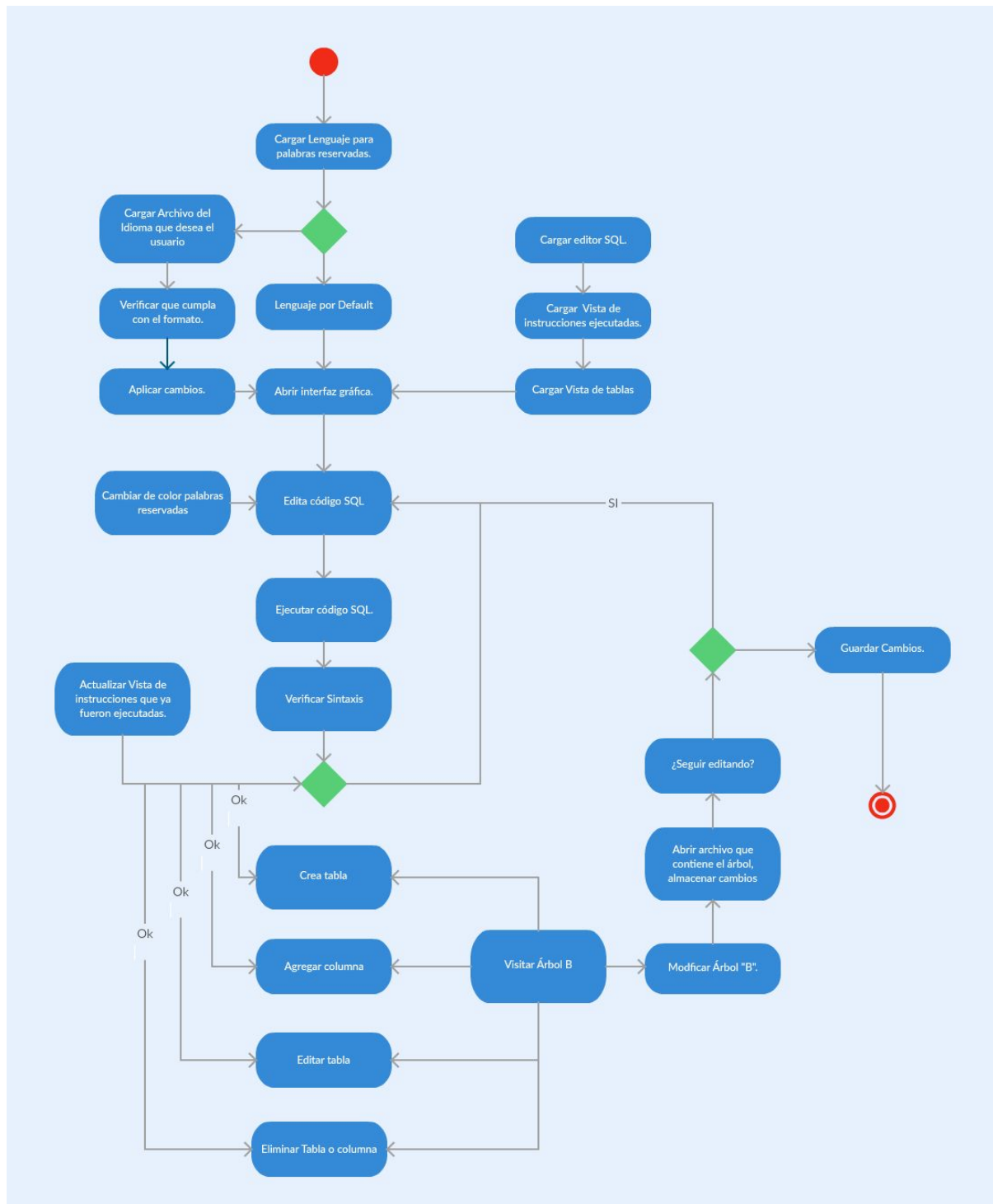
### Descripción:

El usuario interactúa con las funciones principales para acceder a funciones básicas que generan el funcionamiento de la aplicación, para ello existen palabras reservadas y vistas especiales que permiten visualizar y realizar dichas acciones.



Luego de la estructura básica que presenta el programa es necesario conocer cómo funciona internamente el programa, para ello se realizaron dos diagramas de flujo, en el primero se muestra el proceso lógico que en teoría debe seguir la aplicación, realizando acciones que el usuario puede notar visualmente como el cambio de color de las palabras reservadas o de procesos que no puede notar visualmente como lo es el de ingresar datos al árbol "b".

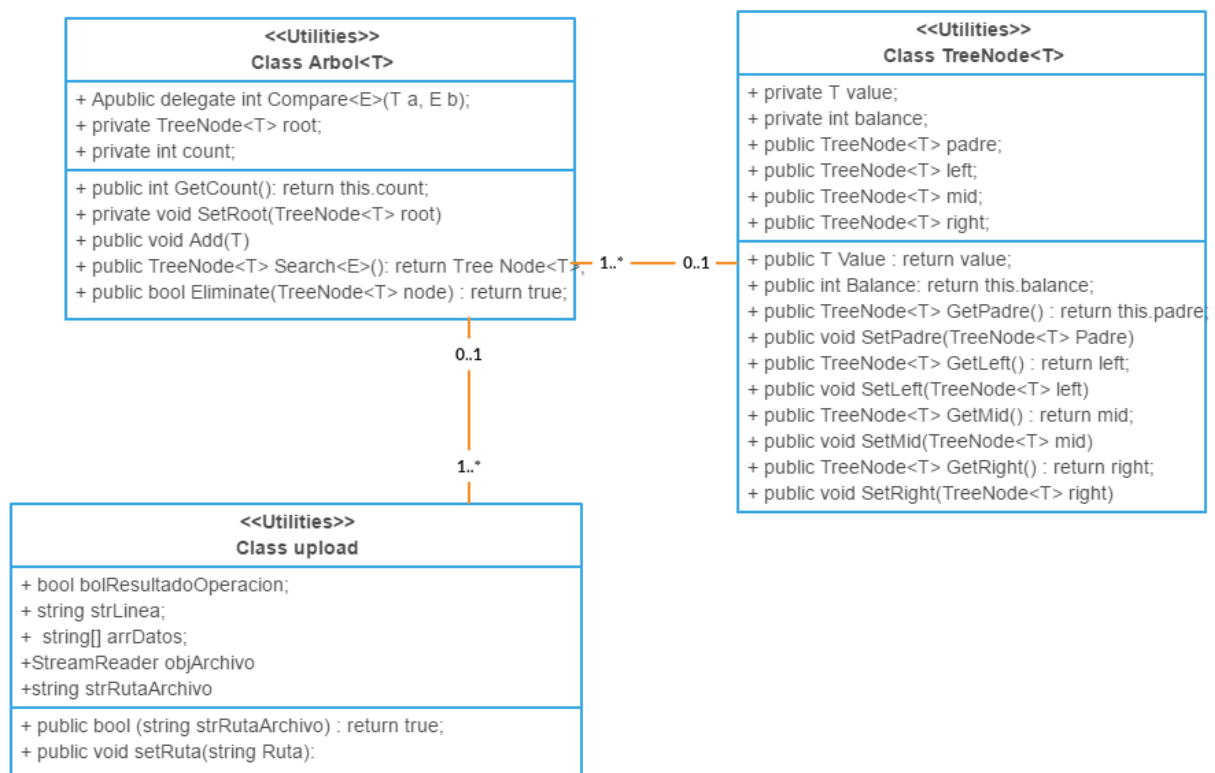
## Manejador de Base de Datos -Diagrama de Flujo 1



El diagrama de flujo 2 muestra el funcionamiento abstracto y lógico que debería tener la estructura de datos a utilizar, en este caso es un árbol del tipo “B” que almacenará los datos de las tablas, columnas que el usuario cree durante la ejecución, este árbol se encargará de almacenar lo creado por el usuario y los cambios que realice en sus datos, para ello se apoya en un archivo de texto que almacena datos en bytes para luego el árbol pueda ir a buscar dentro del archivo los valores que tenga almacenados.

## Manejador de Base de Datos - Diagrama de Flujo 2

## Manejador de base de datos - Diagrama de clases



## Clase Arbol<T>

- Se encargará de almacenar toda la información en disco duro; por medio de un archivo como se especificó anteriormente.
- Cuenta con una dos variables privadas; la primera encargada del conteo de nodos interno que posea el árbol. La segunda es la raíz de dicho árbol. También utiliza un delegado para poder realizar la comparación de las llaves, ya que al ser genérico se debe especificar que se utiliza como llave de parte del objeto en cuestión. Además, cuenta con un único método privado para establecer el valor del nodo raíz y un método para realizar la inserción de nuevos nodos. Luego contiene funciones

públicas, la primera se encarga de obtener la cantidad de nodos en el árbol, la siguiente realiza una búsqueda específica de una llave en el árbol; de ser encontrada devuelve dicho objeto, de lo contrario retorna nulo. Por último, se encuentra la función eliminar, encargada de eliminar cualquier elemento específico, si es que la llave se encuentra en el árbol de lo contrario, retorna falso.

- Se requiere que los objetos cuenten con la implementación de la interfaz `Comparable<T>` y de definir la función `compareTo`. Además, requiere de la implementación de la clase `TreeNode<T>`.

## **Clase `TreeNode<T>`**

- Los nodos del árbol de tipo genérico, se encargan de almacenar la información de manera que se encuentre encapsulada para su fácil manipulamiento.
- Cuenta con una variable privada encargada de almacenar el dato tipo `T`, comúnmente es un objeto. Luego posee las variables públicas; los apuntadores del nodo utilizados para indicar cual es el padre del nodo y sus hermanos. todas las funciones y métodos que posee son públicos, para poder ser accedidos desde la clase `Tree<T>`. Contiene una función que retorna el valor que se almaceno en el nodo, además contiene métodos que retornan los apuntadores. Junto con estas punctiones se encuentran los métodos que establecen el valor de los apuntadores.
- Para esta clase no es necesario ningún requerimiento específico.

## **Clase `Upload`**

- Esta clase se encarga de realizar la subida de datos desde un archivo externo, hacia el árbol genérico.
- Posee una variable booleana que guardará el resultado de las operaciones que realizará el método. luego una variable de tipo `StreamReader`, objeto que maneja el archivo y una variable que guarda la ruta del archivo. La variable `strLinea` almacena temporalmente todo el archivo para luego ser separado y almacenado en `arrDatos`. Su función, realiza la carga del archivo, retornando si la operación fue llevada a cabo de manera exitosa. Por último el método `set Ruta` establece cual es la ruta donde se encuentra el archivo.
- Se requiere la implementación de la clase `Tree<T>` ya que es ahí donde se almacena toda la información.

## ANEXO

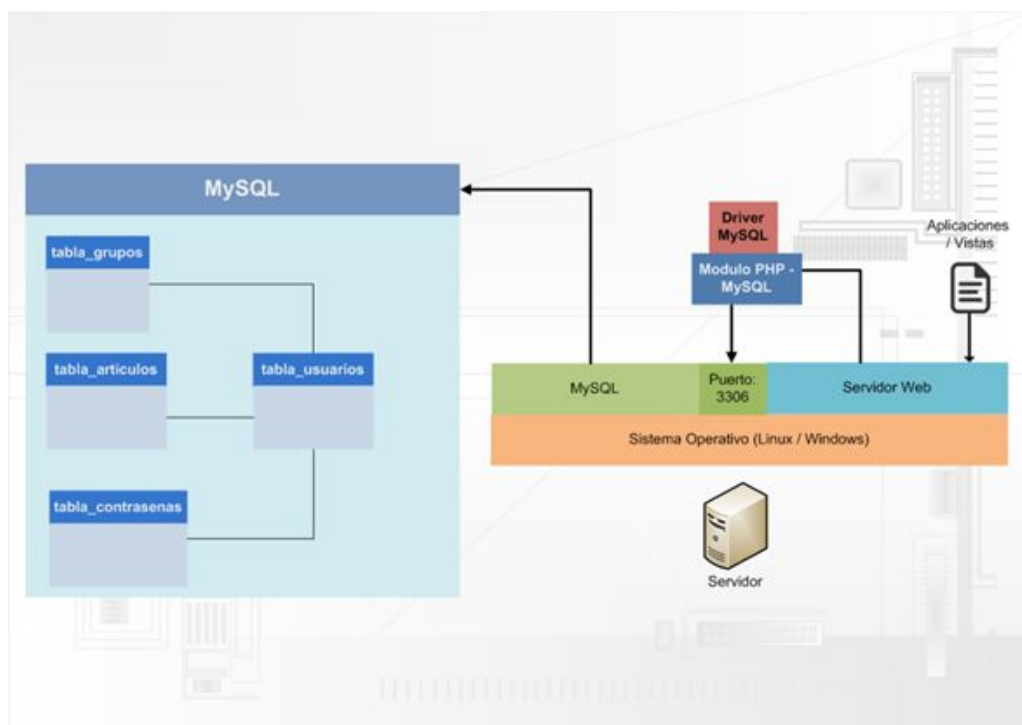
Las bases de datos son un universo aparte, cada tecnología lo es en cierto modo, ser un experto en cualquier tecnología requiere muchos años de experiencia y dedicación.

Solo para mencionarlo, tenemos por ejemplo bases de datos Relacionales (RDBMS – Relational Database Management System como Oracle o MySQL) y otros tipos (NoSQL – Not only SQL “como MongoDB”) que son bastante distintas, lo que tienen en común es el concepto de almacenar información y permitimos acceder a ella usando determinados métodos.

Las bases de datos tienen muchos conceptos tales como tablas, relaciones, vistas, tipos de almacenamiento, foreign keys, motor y demás, seguramente habremos escuchado de algunos de ellos, pero no entraremos mucho en detalle aún.

Como las bases de datos son un mundo tan grande y complejo, vamos a ceñirnos en las definiciones básicas que serán necesarias para comprender cómo funcionan (aquellas RDBMS), sus características y la forma en que suelen trabajar (generalidades), claramente teniendo como punto de referencia al producto MySQL que pertenecía a MySQL AB y posteriormente fue comprado por Sun Microsystems, que finalmente terminó en las manos de Oracle, actual dueño del producto.

MySQL es un motor que en comparación a Oracle DB es muy simple, su arquitectura es menos elaborada, pero es eficiente, fácil y rápido para algunos proyectos web, además de ser OpenSource.



**Fuente:** <http://migueleonardortiz.com.ar>

En el gráfico podemos apreciar el flujo de una aplicación web (PHP), que envía la solicitud al servidor web, el cual gracias al módulo (PHP-MYSQL) se comunica con la base de datos y ejecuta operaciones que han sido escritas en lenguaje PHP/SQL, al final de la solicitud, MySQL recibirá únicamente lenguaje SQL.

- **Tablas:** En la tabla azul del gráfico, se pretende mostrar cómo está constituida la base, es decir, cómo se va a organizar la información, verán varias tablas cada una con una función, almacenar usuarios o contraseñas, artículos (post/publicaciones) o los grupos a los que pertenece un usuario.
- **Relaciones:** Las relaciones (en el ejemplo del gráfico) nos llevaría lógicamente a pensar que cada usuario tenga una única contraseña, esto sería una relación **UNO A UNO**. “Cada usuario con su contraseña”

Sin embargo, el usuario puede escribir muchos artículos, lo cual será una relación **UNO A MUCHOS** “Un usuario puede ser autor de varios artículos”.

También puede existir una relación **MUCHOS A MUCHOS**, donde en nuestro ejemplo, un usuario puede tener varios grupos y un grupo puede tener varios usuarios. “Podemos tener un grupo Administradores que tiene muchos usuarios y también un administrador podría tener más de un grupo”.

## El lenguaje SQL

Algo importante para entender de una base de datos, además de almacenar información, es que todas las operaciones se realizan a través del lenguaje utilizado para ello que es **S.Q.L (Structured Query Language)** “lenguaje de consultas estructuradas”.

A su vez, SQL podría clasificarse según las sentencias:

- **Data Retrieval:** SELECT
- Consultar datos de la base
- **Data Manipulation Language (DML):** INSERT, UPDATE, DELETE, MERGE
- Manipular los datos de la base
- **Data Definition Language (DDL):** CREATE, ALTER, DROP (table), DROP (database), RENAME (table), TRUNCATE
- Manipular la base de datos o sus tablas
- **Data Control Language (DCL):** GRANT, REVOKE
- Permisos a la base de datos y su contenido (tablas)

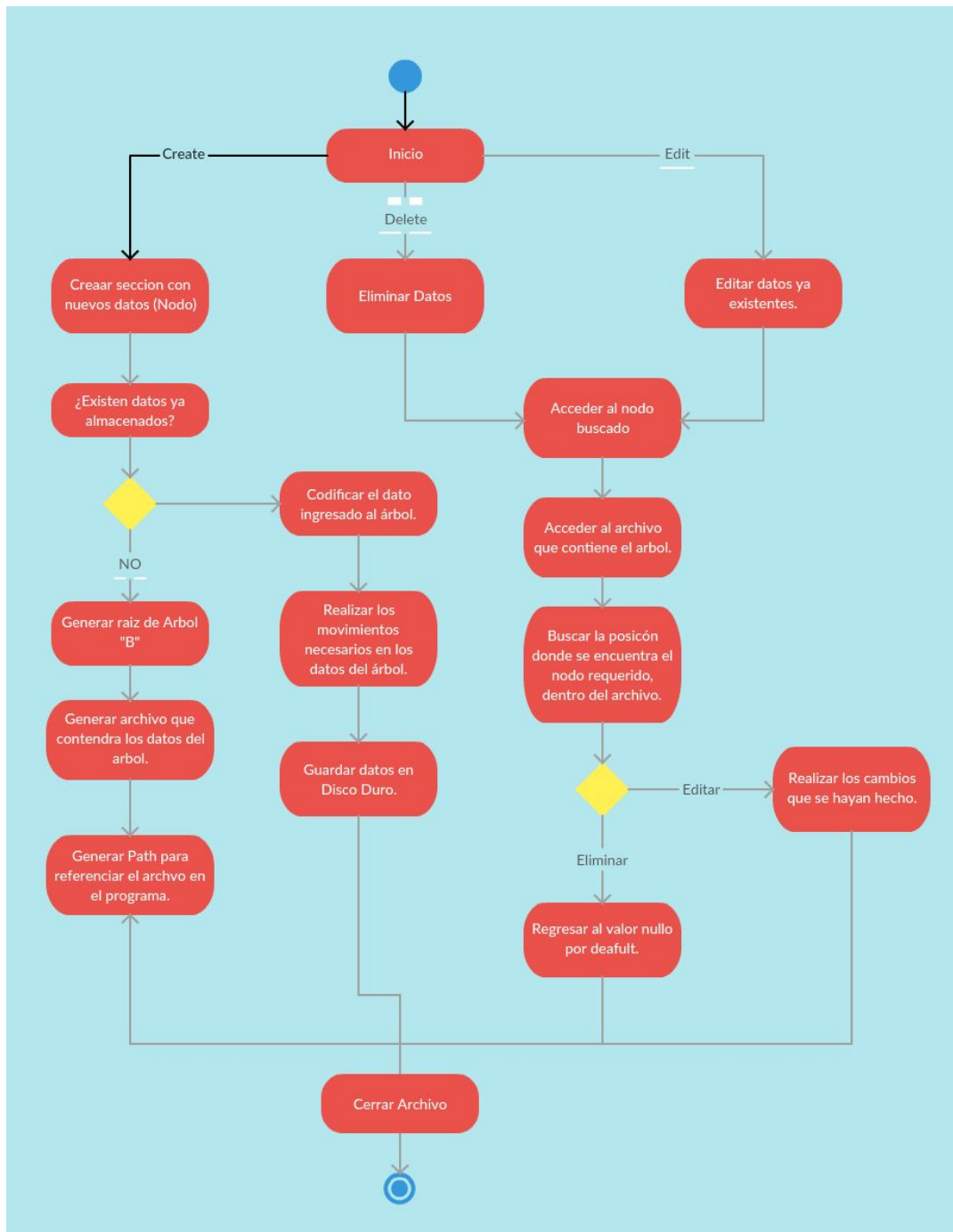
## Las tareas básicas en MySQL

Son las cuatro operaciones básicas que se realizan de forma más habitual, el muy conocido C.R.U.D (Create, Read, Update, Delete) “crear, leer, actualizar o borrar”.

Generalmente, desde nuestra aplicación (PHP, Python, PERL...) nos valdremos del módulo php-mysql del servidor web (Apache HTTP Server por ejemplo) para comunicar a la aplicación con la base. Este módulo tiene el driver (controlador) para comunicarse con la base de datos.

Usualmente, MySQL escucha en el puerto 3306, hay tres formas muy conocidas para comunicarnos con la base:

- Cliente de MySQL
- \*Squirrel SQL (aplicación java / Opensource)
- TOAD (aplicación windows / paga)
- \*phpmyadmin (web, ya viene incluido en XAMPP / Opensource)
- CLI (Command Line Interface)
- \*Es una interfaz de línea de comandos (una pantalla negra, similar a una terminal en Linux o a un CMD en Windows)
- Desde la aplicación web \*Podemos interactuar con la base a través de un lenguaje de programación usando SQL, el servidor web y el módulo lenguaje-de-programación/mysql (Ej: php-mysql).



**E-gráfia:** Ortiz. L. (2015). ¿Qué es MySQL y como funciona? Consultado en Red: <http://migueleonardortiz.com.ar/mysql/que-es-mysql-y-como-functiona/994>