Trabajo Práctico Integrador

“Hotel Sol de las Sierras”

Entrega Final “Algoritmos y estructura de datos”

**Institución:** UTN - FRVM

**Carrera:** Ingeniería en Sistemas de información

**Materia:** Algoritmo y estructura de datos

**Empresa:** Hotel Sol de las Sierras

**Título:** Presentación del prototipado

**Profesores de Cátedra:**

* + Ing. Rinaldi, Mario
  + Ing. Simieli, Paola
  + Ing. Parruccia, Pablo

**Grupo:** Anarqui-Team (Grupo 2) – ([anarquiteam@protonmail.com](mailto:anarquiteam@protonmail.com))

**Integrantes:**

* + COSTAMAGNA, Leonel Osvaldo ([leonel18costamagna@gmail.com](mailto:leonel18costamagna@gmail.com))
  + DEMARCHI, Andrés ([demarchi.andres128@gmail.com](mailto:demarchi.andres128@gmail.com))
  + FANTINI, Marcial Domingo ([marcialfantini@gmail.com](mailto:marcialfantini@gmail.com))
  + GONZÁLEZ, Daniel Agustín ([danielito.2405@hotmail.com](mailto:danielito.2405@hotmail.com))
  + MAZA BIANCHI, Lucas ([mazalucas@hotmail.com](mailto:mazalucas@hotmail.com))
  + MOSCA, Sebastián Jesús ([sebastian.mosca@outlook.com](mailto:sebastian.mosca@outlook.com))
  + VARGAS, Milagros Asunción ([milagros-a-vargas-@hotmail.com](mailto:milagros-a-vargas-@hotmail.com))

**Año de cursado:** 1ͤ ͬ Año - Ingeniería en Sistemas de Información

Índice

**Contenido**

[**ARCHIVOS** 3](#_Toc57762555)

[**Registro** 3](file:///D:\VARGAS%20MILAGROS\Desktop\Entrega%20ARCHIVOS%20FINAL%20-%20copia.docx#_Toc57762556)

[**Campo** 3](file:///D:\VARGAS%20MILAGROS\Desktop\Entrega%20ARCHIVOS%20FINAL%20-%20copia.docx#_Toc57762557)

[**Organización de los archivos** 4](file:///D:\VARGAS%20MILAGROS\Desktop\Entrega%20ARCHIVOS%20FINAL%20-%20copia.docx#_Toc57762558)

[**Organización secuencial** 5](#_Toc57762559)

[**Organización directa o aleatoria** 5](#_Toc57762560)

[**Organización secuencial indexada** 6](#_Toc57762561)

[**Clasificación de los archivos según su función** 7](file:///D:\VARGAS%20MILAGROS\Desktop\Entrega%20ARCHIVOS%20FINAL%20-%20copia.docx#_Toc57762562)

[**Operaciones sobre los archivos** 8](file:///D:\VARGAS%20MILAGROS\Desktop\Entrega%20ARCHIVOS%20FINAL%20-%20copia.docx#_Toc57762563)

[**Actualización de archivos** 9](#_Toc57762564)

[**FLUJO DE ARCHIVOS** 10](#_Toc57762565)

[**Presentación del Sistema de Información** 12](#_Toc57762566)

[**Demostración del código** 13](#_Toc57762567)

# ARCHIVOS

En los principios de la computación, hubo un problema fatal a la utilización práctica y eficaz de una computadora, y ese problema fue el de inconstantibilidad.

Debido a la naturaleza volátil, degradable y de escaso almacenamiento de nuestras memorias principales, la data almacenada en variables es impersistente. La impersistencia es incierta, enigmática, e inverosímil.

Para lidiar con este problema, creamos memorias de mayor capacidad, más económicas, no volátiles e inalterable en condiciones propicias, y en estas memorias, almacenamos **archivos** (aunque si se indaga a mayor profundidad sobre el verdadero uso de los mismos, se podría entablar una relación entre los archivos y cualquier tipo de memoria, sin necesidad de ser no volátiles)

Un archivo o fichero es un conjunto de datos estructurados en una colección de entidades elementales denominadas registros o artículos, que son de igual tipo y constan a su vez de diferentes entidades de nivel más bajo denominadas campos

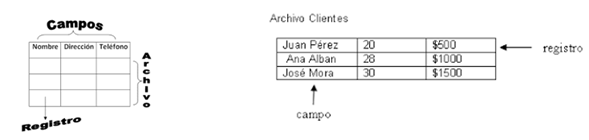
Es una colección de información, normalmente relativa a una entidad particular, y de campos lógicamente relacionados, que pueden ser tratados como una unidad por algún programa.

## **Registro**

Un ejemplo de un registro puede ser la información de un determinado empleado que contiene los campos de nombre, dirección, fecha de nacimiento, estudios, salario, etc.

Se lo puede clasificar como un ítem o elemento de datos elementales, tales como un nombre, número de empleados, ciudad, número de identificación, etc, cuyo propósito es formar divisiones dentro de los propios registros a modo de clasificación de los bytes de datos almacenados. Suele estar caracterizado por su **nombre**, **tamaño  o  longitud**  y  su **tipo  de  datos**  (cadena  de  caracteres,  entero,  lógico,  etcétera.)

## **Campo**



*Como se puede ver en la imagen, los campos están presentes como la “clasificación” de los datos, mientras que los registros son una “colección” de los mismos*

## **Organización de los archivos**

La organización de los archivos esta fundamentalmente ligada a la tecnología empleada para la distribución de la data e información en el medio físico, este medio físico suele referirse como ***soporte***, y podemos hablar de soportes **secuenciales**, y de soportes **direccionables**.

Los soportes secuenciales fueron los primeros desarrollados y la tecnología consiste en organizar la data-información y meta data continuamente ([data1, data2, data3, data4...). Para poder acceder al registro 5, necesitarías acceder al registro 1, registro 2, registro 3, registro 4, y recién ahí uno podría acceder al registro 5.

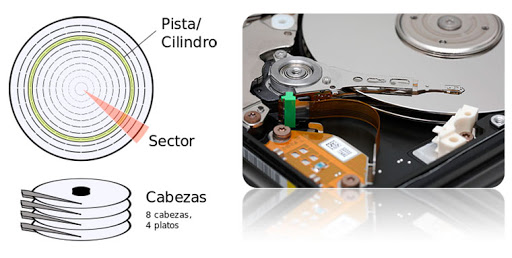
Mientras que los soportes direccionables proveen a los registros de *un campo clave*, una ***key*** que los identifica, una dirección. Esta dirección nos permite acceder a la información de un archivo sin tener que atravesar toda la secuencia de información.

Estos distintos soportes junto a la organización de los registros de los archivos, implican la existencia de dos tipos de acceso a los archivos, secuencial y directo, respectivamente. La significación de estos y cómo funcionan ya fue descrita en la definición de los distintos soportes.

Lo que no abordamos, es la organización de los registros en los archivos.

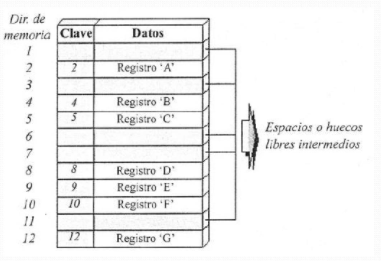
Generalmente, tratamos con tres tipos de formas en las que podemos organizar los archivos:

# **Organización secuencial**

* Los datos se almacenan secuencialmente dentro del archivo (uno después del otro), y es **imposible** acceder a los registros de forma directa sin tener que atravesar toda la serie de datos hasta encontrar lo buscado.
* En el último registro de esta organización, se encuentra un EOF (End of File) FDA.
* Como ventaja al organizar los datos uno tras otro, aprovecha mejor los espacios, siendo muy difícil que deje espacios en blanco o huecos

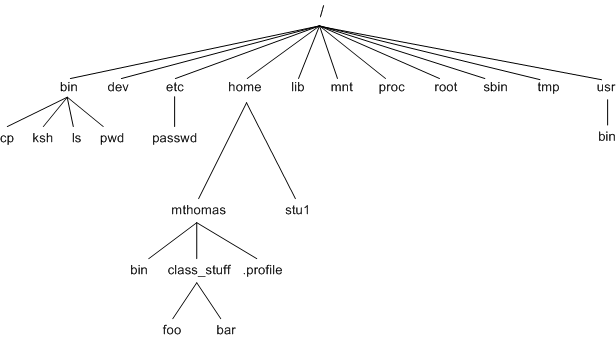
*Un ejemplo claro de este tipo de organización se puede visualizar en un disco magnético*

# **Organización directa o aleatoria**

* Los registros deben tener un campo ***key*** donde se almacena un dato único identificador de este.
* Orden físico no corresponde al orden lógico de los archivos.
* Una gran desventaja de este método es que el fichero contiene gran cantidad de huecos o espacios, por lo tanto el algoritmo para la conversión de las claves y el algoritmo necesario para el almacenamiento y tratamiento de sinónimos han de ser creados de modo que dejen el menor número de huecos libres y se genere el menor número de sinónimos.

*Como se puede visualizar, entre medio de los registros suelen haber espacios en blanco*

# **Organización secuencial indexada**

* El tipo de sus registros contiene un campo clave identificador.
* Los registros están situados en un soporte direccionable por el orden de los valores indicados por la clave.
* Un índice para cada posición direccionable, la dirección de la posición y el valor de la clave; en esencia, el índice contiene la clave del último registro y la dirección de acceso al primer registro del bloque.

*Unix File System, un claro ejemplo de esta forma de organización*

## **Clasificación de los archivos según su función**

Los archivos pueden tener muchas funcionalidades, y todas dependen de cómo lo desee utilizar su usuario, qué datos decida guardar dentro de los registros del mismo, en que contexto lo utilice luego del almacenamiento de datos, etc. Pero para simplificar las subjetividades que puede llegar a generar esta cuestión, existen clasificaciones entabladas objetivamente como las siguientes:

## **Operaciones sobre los archivos**

Los archivos suelen ser permanentes y generalmente de simple acceso al usuario. Una persona desea guardar datos, para luego transformar un conjunto de los mismos en información, y consecuentemente la misma es sumamente valiosa para utilizarla en miles de formas y procesos, y para ello uno debe poder manipular fácilmente su contenido.

**Operaciones básicas que se pueden llevar a cabo sobre los archivos**

|  |  |
| --- | --- |
| *Nombre característico* | *Acción* |
| Apertura | Comunicación del archivo lógico con el físico |
| Creación | Definición del archivo |
| Lectura / Consulta | Acceder al archivo para ver su contenido |
| Fin de Archivo | Detecta el final del archivo |
| Destrucción | Borra completamente el Archivo |
| Reorganización | Ordena la información almacenada en los registros para una mayor optimización |
| Fusión | Reúne varios archivos en uno solo, logrando que de la misma manera los datos almacenados también queden en conjunto |
| Cierre | Cerrar la conexión |

## **Actualización de archivos**

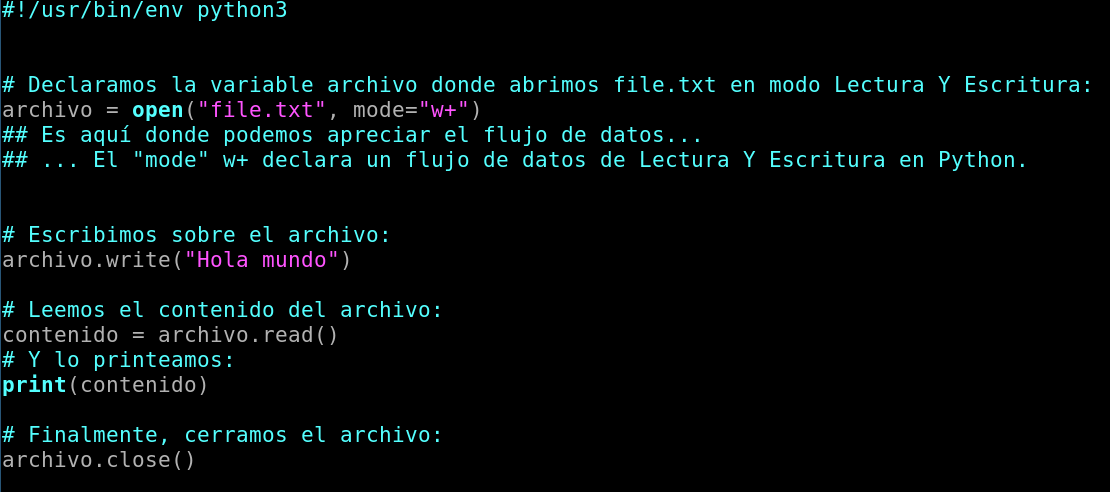
Corresponden al clásico **ABM** (Alta, Baja y Modificación de los archivos o información)

## **FLUJO DE ARCHIVOS**

Los flujos determinan el sentido de la comunicación (lectura, escritura, o lectura/escritura), la posibilidad de posicionamiento directo o no en un determinado registro y la forma de operar el archivo. Cerrar el archivo implica cerrar el flujo, por ejemplo.

Los archivos se clasifican en función del flujo de los datos o por el modo de acceso a los datos del archivo. En función de la dirección del flujo de los datos son de:

**La determinación del tipo de archivo se realiza en el momento de la creación del archivo.**

Por ejemplo, aquí podemos observar un algoritmo en Python 3.8 donde el flujo de datos opera en función de Entrada/Salida (write & read) de un archivo “file.txt” ya existente:

**Cuando lidiamos con archivos a un nivel más elevado, nos topamos que se presentan de dos formas:**

# **Presentación del Sistema de Información**

Nuestro prototipo del sistema de información cuenta con 4 solapas: “Personal”,” Habitaciones”, “Reservas” y “Pasajeros”, cada una de ellas indispensables ya sea para el recepcionista, como el encargado de reserva y el encargado de personal del hotel.

En su pantalla principal cuenta con las estadísticas que se actualizan de forma automática dependiendo de los ingresos de información que se producen en la Base de Datos referentes a las edades de los pasajeros registrados, como las diferentes nacionalidades o proveniencias de los mismo.

Hemos implementado el uso de muchas herramientas para completar el sistema de información que se va a describir a continuación. Se ha desarrollado a la completitud el sistema con las siguientes tecnologías:

* PHP
* HTML
* CSS
* JavaScript
* JQuery (librería)
* CanvasJS (librería)
* Bootstrap (FrameWork)
* MariaDB
* phpMyAdmin
* SQL

Facilitamos una web para la prueba de nuestro software donde se puede interactuar con el sistema sin necesidad de instalar nada. Si quieres utilizar este sistema en tu propio servidor, debes instalar una distribución de Apache mínimamente con PHP y una base de datos MySQL para su funcionamiento correcto.

# **Demostración del código**

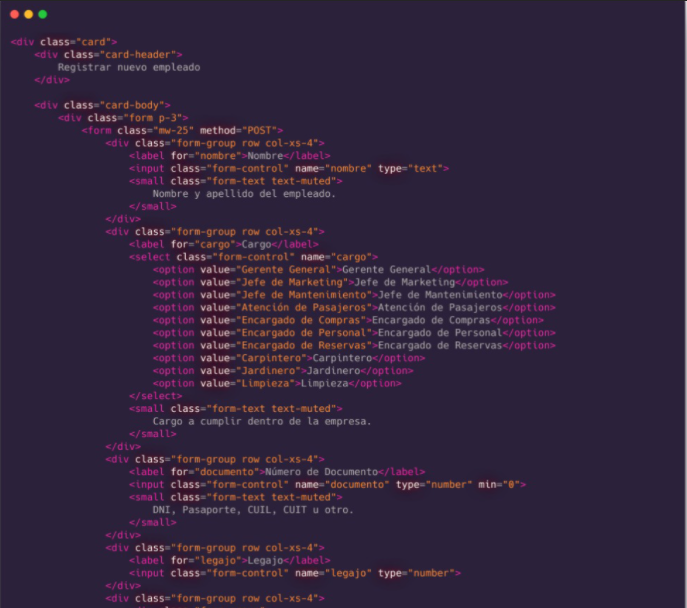
HTML - Descripción de la Sección Empleados y Head

Dentro de la etiqueta <head> generalmente se ingresan datos no visibles al usuario, y dentro de <body> se ingresan todos los elementos visuales de la página. Aquí solo se construyó un encabezado para la pestaña de empleados, y un cuadro de texto con una pequeña descripción



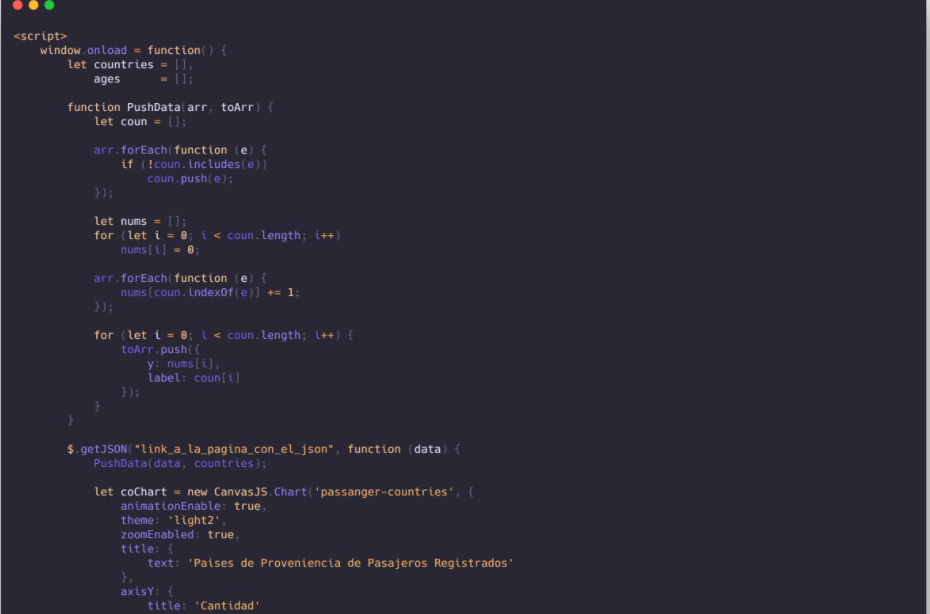
HTML - Tabla de la Sección Empleados

Aquí se ensambla la tabla donde se visualizarán en un futuro todos los datos de los empleados ingresados al sistema. La misma se conforma generando una etiqueta principal <table>, y luego dentro de la misma mediante <thead> se nos sitúa en la cabecera de la tabla, <tr> para crear una fila, y finamente con <th> se generan los campos uno por uno, indicando su contenido

****

HTML - Ejemplo Formulario para Registrar Nuevos Empleados

Aquí se puede visualizar un ejemplo de como realizamos los diferentes formularios que se encuentran en nuestro sistema. Mediante la etiqueta <form> se indica que lo que se encuentre dentro va a ser parte de un formulario de ingreso de datos. Para cada entrada se utiliza la etiqueta <input> y dentro de la misma se indica a que tipo de entrada pertenece (número, correo electrónico, texto simple, etc). En este conjunto de entradas se utilizó la etiqueta <select>, la cual crea una casilla en forma de lista desplegable con diferentes opciones a elegir, definidas cada una por <option>

****

JavaScript - Script para Renderizado de Gráficas Estadísticas en CanvasJS

Esta parte del código básicamente lo que hace es tomar un JSON de las dos API REST y convierte los datos contenidos en el mismo en un formato aceptado por CanvasJS, y asi poder generar debidamente la gráfica presente en la página principal del sistema



JSON - API REST 1

Representan el formato en el que llegan los datos provenientes de la API REST de los países de proveniencia de los pasajeros

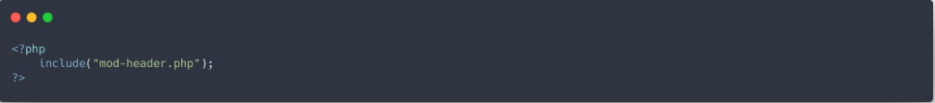


PHP - Conexión a la Base de Datos MySQL

Acá se puede visualizar como nos podemos conectar generalmente a una Base de Datos perteneciente a MySQL para realizar el almacenamiento de la información generada por el sistema

JSON - Convertido para CanvasJS

Representa el JSON convertido para ser interpretado finalmente por el CanvasJS

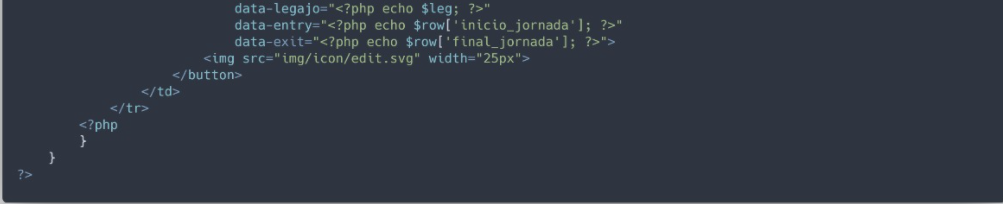


PHP - Ejemplo de la Importación de un Módulo

Asi se importan los módulos dentro de las páginas, mediante include()

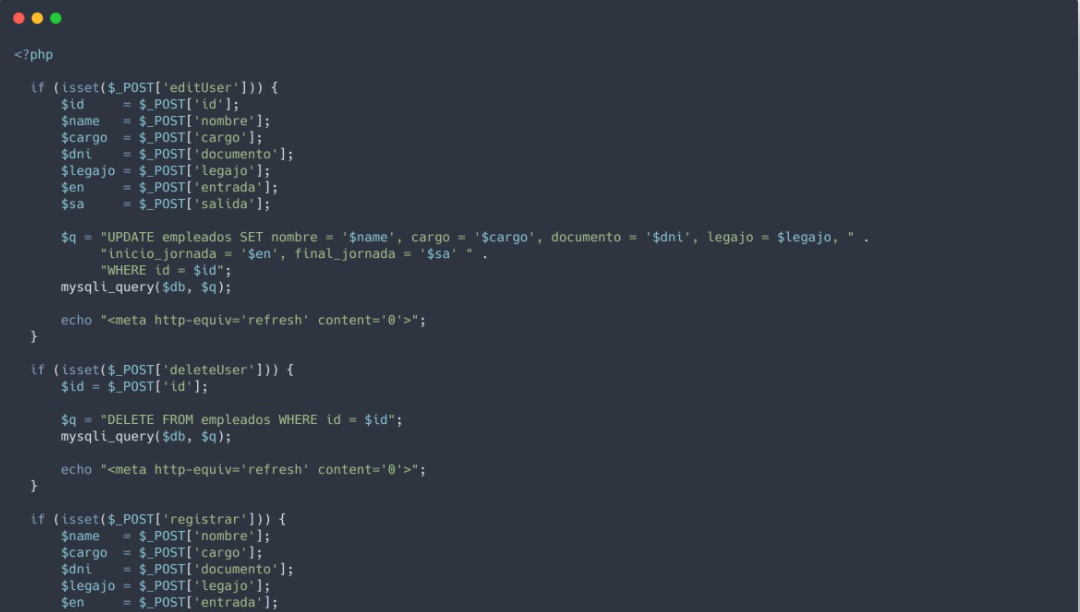
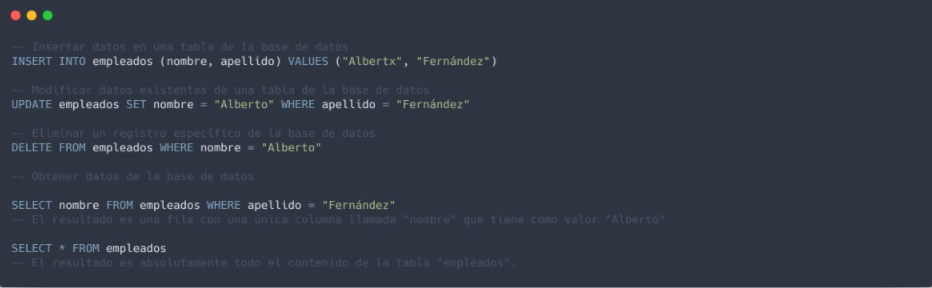
PHP - Ejemplo de un Módulo (Barra Navegable)

Parte del código donde se representa el armado de una barra navegable (la que se encuentra presente en la parte superior de la página en todo momento). La misma fue ensamblada principalmente con la biblioteca Bootstrap (muy utilizada en la creación de páginas webs por su simplicidad en el armado de interfaces y parte frontal de la misma). Además, cada botón está conectado a su respectivo archivo .php, en el cual dentro del mismo se encuentra toda la codificación encargada de llevar a cabo la parte “backend” o funcional del sistema



PHP - Ejemplo de Query MySQL para Imprimir Datos en la Aplicación

Pide datos de la base, los trae, itera sobre todos los resultados y los imprime en código HTML para luego mostrarlos en la tabla de empleados que se puede visualizar en la página



MySQL - Queries de Ejemplo y Comandos que utilizamos en el prototipo

Aquí decidimos mostrar a modo de ejemplificaciones los comandos que se suelen utilizar para realizar trabajos con MySQL de forma externa al mismo, ya sea para realizar ingresos de información nueva, eliminar registros determinados de la tabla, actualizar los mismos, o seleccionar un registro en particular para traerlo al programa para visualizar sus datos

PHP - Detección de Envío de Formularios e Inserción en la Base de Datos de los mismos

Aquí lo que se hace es detectar cuando el usuario del sistema realizó un envío de información a través de un formulario; posteriormente a ello, los datos son ingresados dentro de la Base de Datos perteneciente a la pestaña en la cual se encuentre. Además de registrar datos, también se realiza las actualizaciones y/o eliminaciones de los mismos dentro de la BBDD