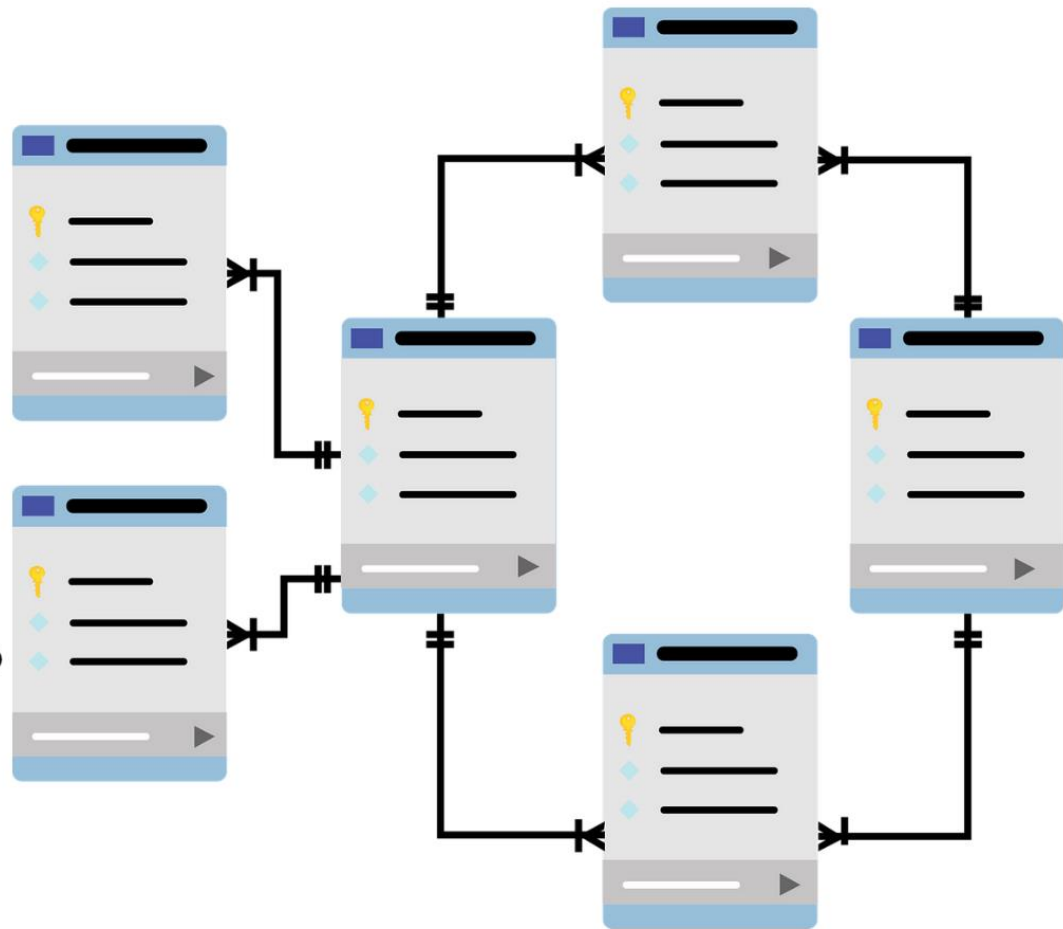


# CARRERA TÉCNICA EN OFIMÁTICA

## Módulo 4. Diseña y gestiona bases de datos ofimáticas

Quinto semestre



Submódulo 2

## **Dirección General de Educación Tecnológica Agropecuaria y Ciencias del Mar**

### **Créditos**

#### **Desarrollo de Contenido**

*Azucena del Carmen Loera Frías*

*Manuel Alejandro Moreno Raya*

*Salvador Gutiérrez Olague*

*Tawny Nava García*

#### **Revisión técnico – pedagógica y edición**

*Arit Furiati Orta*

*Itandehui García Flores*

*Judith Doris Bautista Velasco*

México, 2021

## Presentación

Actualmente los procesos de enseñanza y de aprendizaje se han diversificado en las formas, métodos y medios a través de los cuales se realizan para brindar una educación de calidad, por lo que cada día las instituciones educativas deben coadyuvar en dichos procesos a través de estrategias y acciones que favorezcan en los alumnos la adquisición de los aprendizajes tanto con la mediación de un docente de manera presencial como, en ocasiones singulares, a distancia.

Acorde con los principios de la Nueva Escuela Mexicana, los alumnos son sujetos activos y responsables de su propio aprendizaje, por lo que Dirección General de Educación Tecnológica Agropecuaria y Ciencias del Mar (**DGETAyCM**) pone a disposición de los estudiantes el presente material de apoyo que tiene el propósito de brindar elementos teóricos de los módulos profesionales de la **carreta técnica en ofimática**, así como el reforzamiento de estos a través de actividades de aprendizaje.

El material está organizado de modo progresivo para abordar los contenidos de la carrera Técnico en ofimática, en el presente material se trabajará el **Módulo IV “Diseña y gestiona bases de datos ofimáticas”**.

- Submódulo 1. Diseña bases de datos ofimáticas
- Submódulo 2. Gestiona información mediante el uso de sistemas manejadores de bases de datos ofimáticas

En este material se abordarán los contenidos del **submódulo 2**.

El primer apartado de cada lección denominado **“Contextualizando”** se muestra un primer acercamiento a los conceptos que se abordan, articulándolos con escenarios y situaciones de la vida cotidiana, con la intención de realizar asociaciones derivadas de los conocimientos previos de los estudiantes. En el apartado **“Vamos a aprender”** se integra información para analizar los conceptos y características de la temática. En la sección de **“Actividades de aprendizaje”** se proponen actividades para para asimilación de los principales conceptos y características del tema. En el apartado **“Autoevaluación”** se plantean una serie de indicadores de desempeño que buscan evaluar los aprendizajes e identificar los contenidos a reforzar. Finalmente, en la sección **“Para saber más”** se proporcionan recomendaciones para complementar los contenidos como videos y lecturas. En algunos casos, se ha incorporado la sección **“Ponlo en práctica”**, en se ofrecen algunos tutoriales o prácticas sugeridas para llevar a cabo con equipo de cómputo o dispositivo.

Deseamos que este material apoye la formación académica y sea una herramienta de utilidad en los procesos de aprendizaje para los estudiantes.

# Índice

	<b>Pág.</b>
<b>Submódulo 2. Gestiona información mediante el uso de sistemas manejadores de bases de datos ofimáticas</b>	
Consultas a bases de datos con lenguaje SQL----- (Salvador Gutiérrez Olague)	<b>7</b>
Genera y modifica consultas de bases de datos----- (Salvador Gutiérrez Olague)	<b>22</b>
Introducción a HTML----- (Azucena del Carmen Loera Frías)	<b>27</b>
Conexión con base de datos MySQL con PHP y diseño de formularios----- (Salvador Gutiérrez Olague)	<b>39</b>
Análisis y diseño de sistemas----- (Tawny Nava García y Manuel Alejandro Moreno Raya)	<b>76</b>

## Estructura didáctica

Este material está dividido en submódulos y a lo largo de cada uno de ellos encontrarás diferentes secciones las cuales te facilitarán el abordaje de cada contenido.

En esta sección se delimitarán conceptos y características del tema a revisar, así como articulación de los contenidos con tus conocimientos previos relacionados con el tema y la relevancia de éstos en tu formación profesional/académica.

Emplearás los contenidos revisados para asimilar los principales conceptos y promover el desarrollo de las competencias profesionales.

En esta sección encontrarás información para analizar los conceptos y características del tema con énfasis en las competencias profesionales.

Evaluarás tus aprendizajes sobre los temas abordados e identificarás los contenidos que debes reforzar.

En este apartado se te proporcionan recomendaciones para profundizar en los contenidos.

¡Vamos a aprender!

En la actualidad has escuchado mencionar el *computación en la nube*, o has oído decir "¿en la nube", "¿súbelo en la nube", pero ¿sabes la nube?

La *nube* es un modelo de soporte tecnológico que brinda acceso a un conjunto de recursos e informáticos compartidos, por ejemplo servidores, almacenamiento, aplicaciones y servicios.

La *nube* es un modelo de soporte tecnológico que brinda acceso a un conjunto de recursos e informáticos compartidos, por ejemplo servidores, almacenamiento, aplicaciones y servicios.

o precipita con esas grandes tormentas. ¿En otras cosas se pueden descargar o subir? ¿y el agua se almacena en las nubes?

Condensación  
Evaporación  
Precipitación  
Infiltración

**Actividades de aprendizaje**

Lee las siguientes oraciones y subraya la respuesta correcta.

- Este tipo de nube se caracteriza por ofrecer estos servicios pueden ser gratuitos o pueden ser de pago.  
a) Encriptar      b) Pública
- ¿Su uso es exclusivo de una persona o una empresa o son usuarios a los que la empresa les presta el servicio?  
a) Híbrida      b) Cifrar
- ¿Ofrece servicios donde se comparte información como música, videos, tutoriales, cocina, entre otros?  
a) Híbrida      b) Pública
- ¿Oculta el contenido de un mensaje a similitud de un correo electrónico?  
a) Cifrar      b) Pública
- Si al conectarte a la red no te solicita una contraseña o te pide un código de seguridad, ¿cómo se llama este tipo de conexión?  
a) Cifrar      b) Seguridad en la nube

**Autoevaluación**

Reflexiona y evalúa los conocimientos, habilidades y actitudes que adquiriste en esta lección.

Coloca una X en la columna que corresponda al desempeño que consideras que tienes para cada indicador.

Indicadores	Lo puedo hacer	Tengo dudas	Necesito trabajar más
Comprendo el concepto de computación en la nube.			
Conozco cuáles son las ventajas del uso de la computación en la nube.			
Entiendo cuáles son las desventajas de la computación en la nube.			

**Para saber más**

- Capacítate para el empleo (2021). *Curso Fundamentos de cómputo en la nube*. Fundación Carrión Slim. <https://capacitateparaempleo.org/pages.php?r=tema&tagID=8440>
- Surveillance. Self-defense (2018). *Qué debo saber sobre el cifrado*. <https://ssd.eff.org/es/module/3C2%BEq%4C2%A9-es-el-cifrado>

35

¡Ponlo en práctica!

Adicionalmente, en algunas lecciones podrás encontrar tutoriales o prácticas sugeridas para que practiques lo que aprendiste, en algún dispositivo electrónico o equipo de cómputo.

# Submódulo



## Gestiona información mediante el uso de sistemas manejadores de bases de datos ofimáticas

### Competencias profesionales

- Diseña bases de datos.
- Crea y modifica estructura de bases de datos.
- Gestiona información de bases de datos.
- Relaciona bases de datos.
- Crea y modifica consultas de bases de datos.
- Crea y modifica reportes de bases de datos.
- Crea y modifica formularios de bases de datos.

# Consultas a bases de datos con lenguaje SQL



## Contextualizando



Cuando vas a la biblioteca de tu escuela o cuando has consultado una biblioteca virtual, ¿qué datos te piden?, ¿cómo es la forma en la que preguntas al encargado o al buscador por un libro? ¿Debes saber el título exacto de la bibliografía que estás buscando? ¿Puedes hacer la búsqueda con otros datos como el autor?

Tal vez en alguna ocasión has buscado ropa o algún producto en tiendas virtuales ¿cómo buscas un tipo de prenda? ¿Has identificado los filtros que existen en estas páginas? ¿Te has dado cuenta de que puedes realizar tu búsqueda por talla, color, rango de precios, etcétera? ¿Cómo hacen estos sistemas para encontrar la información exacta que buscas?



## ¡Vamos a aprender!

Hablando de bases de datos, la **consulta** es una forma de acceder a los datos de una o varias tablas, de manera que puedes obtener la información precisa que requieras. En esta lección aprenderás acerca de algunos tipos de consultas como las simples y multitablas.

En lecciones anteriores trabajaste con el lenguaje **SQL** (Structured Query Language), en español, Lenguaje de Consulta Estructurada y pudiste darte cuenta de que las palabras están escritas en el idioma inglés.

En esta lección, aprenderás la forma de obtener información de una base de datos de manera más precisa y especializada, utilizando sentencias. Podrás generar búsquedas de datos específicos en las tablas que la conforman, establecer filtros y presentar al usuario final lo que requiere. Este proceso lo llevarás a cabo utilizando el lenguaje SQL en el programa **Laragon**.

Recordemos algunas sentencias que te serán de utilidad:

- La sentencia **Select**, (seleccione) en su forma más básica, te permite consultar u obtener información de una determinada tabla y te presenta todos los registros.
- El signo **\*** es un comodín que indica que queremos ver todos los registros.
- **Where** (Donde) es la palabra con la que determinas que se va a establecer una condición para la búsqueda de datos.



- **Like** En este caso se traduce “como” y es la palabra que utilizarás para comparar la cadena de caracteres que se busca y establecer la coincidencia con un registro en la tabla que se desea consultar.
- La sentencia **From** (desde) especifica de qué tabla queremos obtener los registros o datos.
- Cuando escribes **Nombre\_de\_tabla** indicas el nombre de la tabla de donde se obtienen los registros.
- **And**- Operador lógico utilizado para establecer dos condiciones en una consulta y que muestra únicamente los registros que cumplan con ambas.
- **Or**- Operador lógico utilizado para establecer dos condiciones en una consulta, pero que en este caso mostrará los registros que cumplan con una u otra.

El siguiente ejemplo puede ejecutarse en el programa mencionado, con las sentencias anteriormente revisadas y a las que a su vez podemos denominar **consultas simples**. La tabla con la que estarás trabajando en este apartado se llama **“libro”**, y los campos que la componen son Id\_libro, Título, Cantidad y Id\_Autor. Una vez que se ejecuta la consulta, puedes ver el contenido de la tabla completa con todos los datos de esta.

Vamos a observar el siguiente problema y su solución:

Si quieres realizar una consulta que te muestre la totalidad de libros que están en la biblioteca, visualizando todos sus campos, debes utilizar la siguiente sentencia:

**SELECT \* FROM libro;**

Con esta consulta estás especificando que quieres ver todos los campos y todos los registros de la tabla “libro”.

The screenshot shows a SQL query execution window with the following content:

Host: localhost Base de datos: biblioteca Consulta\*

```
1 SELECT * FROM libro;
```

libro (4x7)

Id_libro	Título	Cantidad	FKId_Autor
1	El Guardian en el centeno	5	1
2	Nueve historias	4	1
3	Franny y Zooey	6	1
4	El gran Gatsby	10	2
5	Tierra id la noche	6	2
6	Orgullo y prejuicio	5	3
7	Profesional ASP.NET 4.5 en C # y VB	7	4

Hasta el momento conoces la forma de consultar o de mostrar todos los registros de todos los campos, pero ahora verás cómo ir seleccionando información más específica.



La cláusula WHERE, especifica los criterios a cumplir para mostrar los datos que requieres. Observa la siguiente sintaxis:

**SELECT CAMPOS FROM TABLA\_DE\_DATOS WHERE CONDICIONES;**

Si observas bien después de la palabra WHERE, vienen las condiciones que vamos a especificar para obtener ciertos datos. Vamos a dar un ejemplo tomando en cuenta la tabla con la que trabajamos en el ejercicio del submódulo anterior. ¿La recuerdas?

Id_libro	Título	Id_Autor	Nombre	País
1	El Guardián en el centeno	1	JD Salinger	Estados Unidos
2	Nueve historias	1	JD Salinger	Estados Unidos
3	Franny y Zooey	1	JD Salinger	Estados Unidos
4	El gran Gatsby	2	F. Scott. Fitzgerald	Estados Unidos
5	Tierna id la noche	2	F. Scott. Fitzgerald	Estados Unidos
6	Orgullo y prejuicio	3	Jane Austen	Reino Unido
7	Profesional ASP.NET 4.5 en C # y VB	4	Scott Hanselman	Estados Unidos
7	Profesional ASP.NET 4.5 en C # y VB	5	Jason N. Gaylord	Estados Unidos
7	Profesional ASP.NET 4.5 en C # y VB	6	Pranav Rastogi	India
7	Profesional ASP.NET 4.5 en C # y VB	7	Todd miranda	Estados Unidos
7	Profesional ASP.NET 4.5 en C # y VB	8	Christian Wenz	Estados Unidos

Requerimos buscar en la base de datos el libro titulado "Orgullo y prejuicio" para ello colocaremos la siguiente indicación, en Laragon.

**SELECT \* FROM libro WHERE Título="Orgullo y Perjuicio"**

Ahora, observa lo que indica cada elemento de la instrucción.

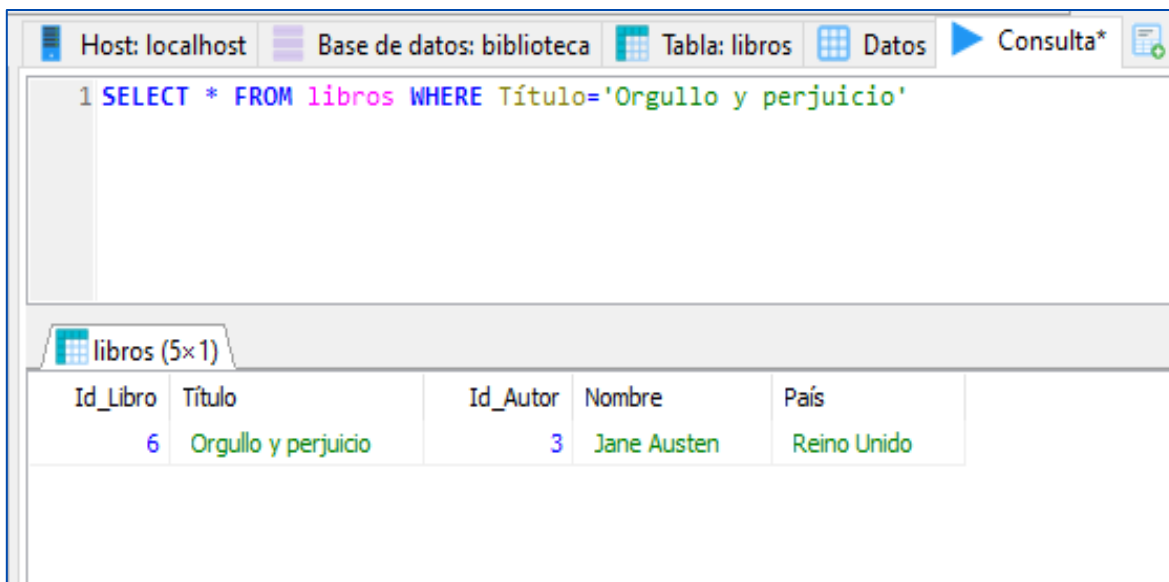
**SELECT** Es la sentencia que vas a aplicar, en este caso **seleccionar**

**\*** El asterisco indica que deseas revisar todos los campos de la tabla.

**FROM** (desde) especifica de qué tabla queremos obtener los registros o datos.

- Libro** Es el nombre la tabla que se está consultando
- WHERE** Dónde, es la palabra con la que determinas que se va a establecer una condición para la búsqueda de datos, en este caso se va a especificar el título exacto que se busca.
- Título** Es el nombre del campo que se está consultando
- Orgullo y prejuicio** Indica el nombre exacto del registro que deseas ver

Ahora observa de manera gráfica cómo obtendrás la información.



De esta manera te das cuenta de que puedes seleccionar los datos de varias maneras.

Revisa a continuación algunas opciones que tienes para poder ver los datos, para poder dar atención a tu cliente cuando tenga otras necesidades de consulta.

Es posible ver los datos utilizando dos condiciones y que ambas se cumplan, para eso utilizarás el operador **AND**, que podrás ver a través de un ejemplo ya que la sintaxis general, la conoces.

Id_libro	Título	Id_Autor	Nombre	País
1	El Guardián en el centeno	1	JD Salinger	Estados Unidos
2	Nueve historias	1	JD Salinger	Estados Unidos
3	Franny y Zooey	1	JD Salinger	Estados Unidos
4	El gran Gatsby	2	F. Scott. Fitzgerald	Estados Unidos
5	Tierna id la noche	2	F. Scott. Fitzgerald	Estados Unidos
6	Orgullo y prejuicio	3	Jane Austen	Reino Unido
7	Profesional ASP.NET 4.5 en C # y VB	4	Scott Hanselman	Estados Unidos
7	Profesional ASP.NET 4.5 en C # y VB	5	Jason N. Gaylord	Estados Unidos
7	Profesional ASP.NET 4.5 en C # y VB	6	Pranav Rastogi	India
7	Profesional ASP.NET 4.5 en C # y VB	7	Todd miranda	Estados Unidos
7	Profesional ASP.NET 4.5 en C # y VB	8	Christian Wenz	Estados Unidos

**SELECT \* FROM libro WHERE Id\_Autor=1 AND Cantidad=5;**

<b>Select</b>	Seleccionar es la sentencia que vas a aplicar.
<b>*</b>	El asterisco indica que deseas revisar todos los campos de la tabla.
<b>From</b>	Especifica de dónde vas a tomar los datos.
<b>Libro</b>	Nombre de la tabla que se va a consultar,
<b>Where</b>	Dónde, es la palabra con la que determinas que se va a establecer una condición para la búsqueda de datos.
<b>Id_Autor=1 AND Cantidad=5</b>	Se especifica que se desea ver los libros del autor con Id número 1 y que además tengan 5 en existencia.

Host: localhost Base de datos: biblioteca Tabla: libro Datos Consulta

```

1 |
2 | SELECT * FROM libro WHERE Id_Autor=1 AND Cantidad=5;

```

libro (2x4) libro (4x2)

Id_libro	Título	Cantidad	Id_Autor
1	El guardián en el centeno	5	1
3	Franny y Zooey	5	1

Si te das cuenta en el resultado, se aplican las dos condiciones y se filtran los registros que las cumplen.

Ahora observa que al aplicar el operador lógico OR, nos muestra los registros que cumplan una u otra condición.

Host: localhost Base de datos: biblioteca Tabla: libro Datos Consulta\*

```

1 |
2 | SELECT * FROM libro WHERE Id_Autor=1 OR Cantidad=5;

```

libro (4x7)

Id_libro	Título	Cantidad	Id_Autor
1	El guardián en el centeno	5	1
2	Nueve historias	4	1
3	Franny y Zooey	5	1
4	El gran Gatsby	5	2
5	Tierna id la noche	5	2
6	Orgullo y prejuicio	5	3
7	Profesional ASP.NET 4.5 en C # y VB	5	4

En el resultado de la ejecución de la consulta, aparecen aquellos registros que son del autor 1 y también de los que en existencia hay 5, deben cumplirse ambas condiciones.

Se concluye entonces que las anteriormente mencionadas son las sentencias más utilizadas en bases de datos y con las que se pueden llevar a cabo la mayoría de las consultas de cualquier tipo.

## Operadores

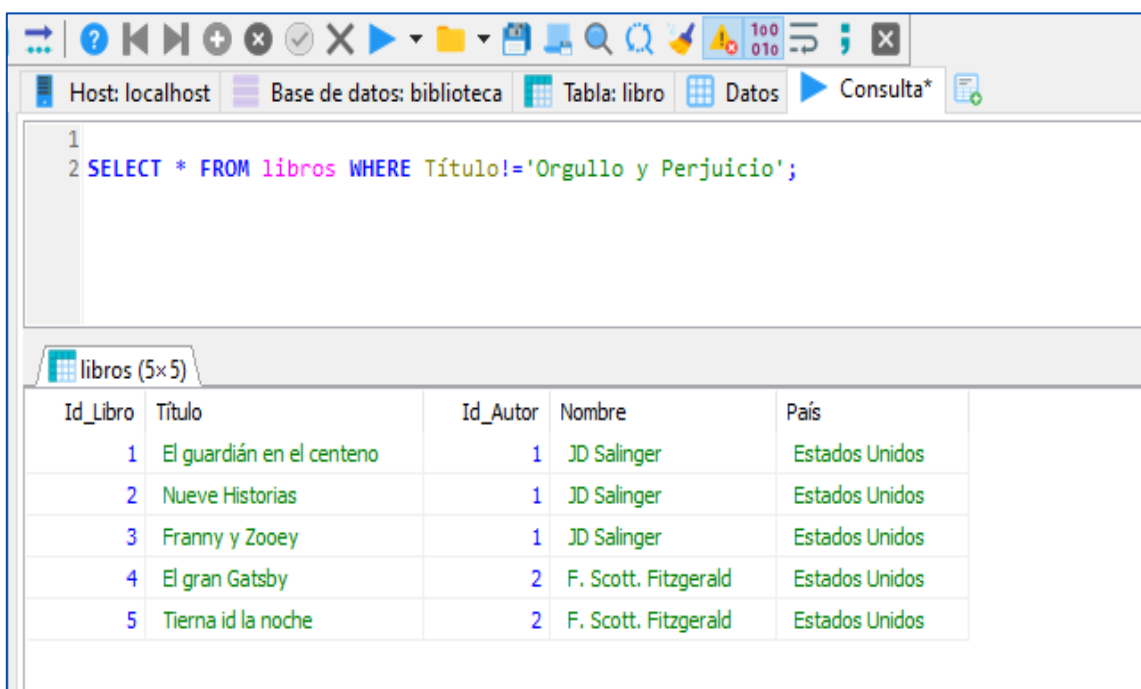
Los **operadores** son utilizados para realizar búsquedas en tablas de una manera más específica y permiten obtener resultados más pertinentes.

Ahora verás cómo funcionan los **operadores de comparación de cantidad** (igual, diferente, menor y mayor que). Para el operador de igualdad ya tienes un ejemplo anterior en la Ilustración 1.

Cuando quieres ocultar un registro o hacer excepción de él, utiliza el operador **“diferente que”** que se puede expresar de dos formas:

!=                                  <>

```
Select * from libros WHERE Título!='Orgullo y Perjuicio';
```



The screenshot shows a database management interface. At the top, there is a toolbar with various icons. Below the toolbar, the interface shows the following details:

- Host: localhost
- Base de datos: biblioteca
- Tabla: libro
- Datos
- Consulta\*

The SQL query entered in the editor is:

```
1  
2 SELECT * FROM libros WHERE Título!='Orgullo y Perjuicio';
```

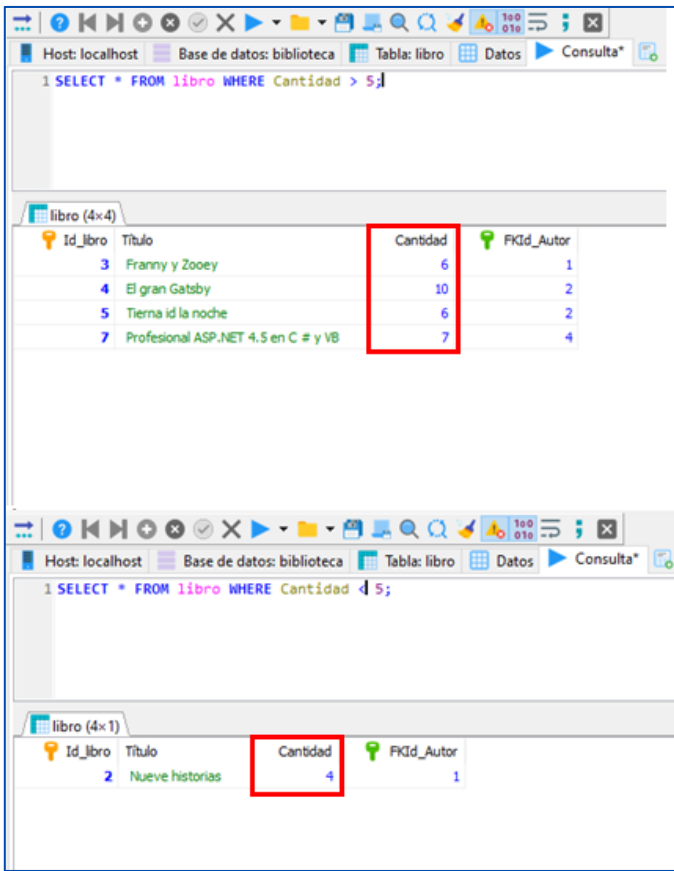
The results are displayed in a table with 5 rows and 5 columns:

Id_Libro	Título	Id_Autor	Nombre	País
1	El guardián en el centeno	1	JD Salinger	Estados Unidos
2	Nueve Historias	1	JD Salinger	Estados Unidos
3	Franny y Zooey	1	JD Salinger	Estados Unidos
4	El gran Gatsby	2	F. Scott. Fitzgerald	Estados Unidos
5	Tierna id la noche	2	F. Scott. Fitzgerald	Estados Unidos

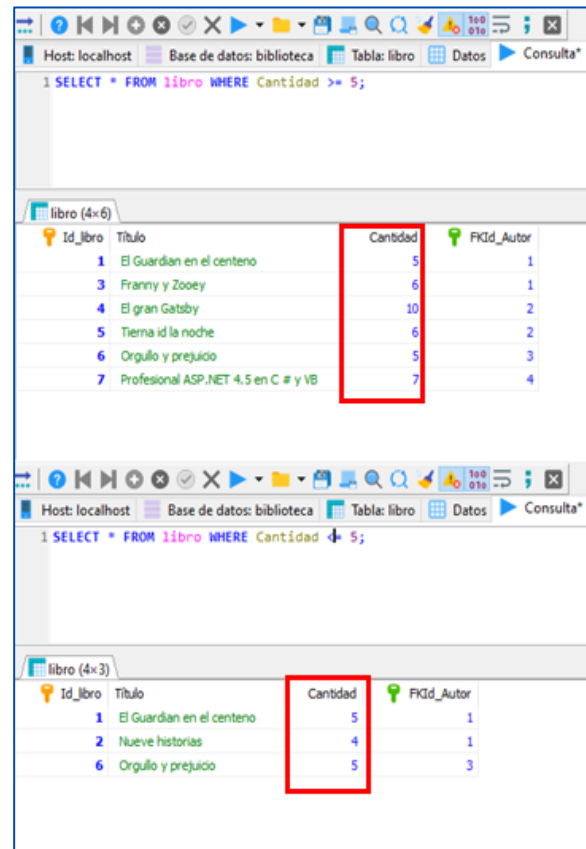
Como te puedes dar cuenta, aparecen todos los títulos, excepto el que especificamos en la consulta (Orgullo y perjuicio).

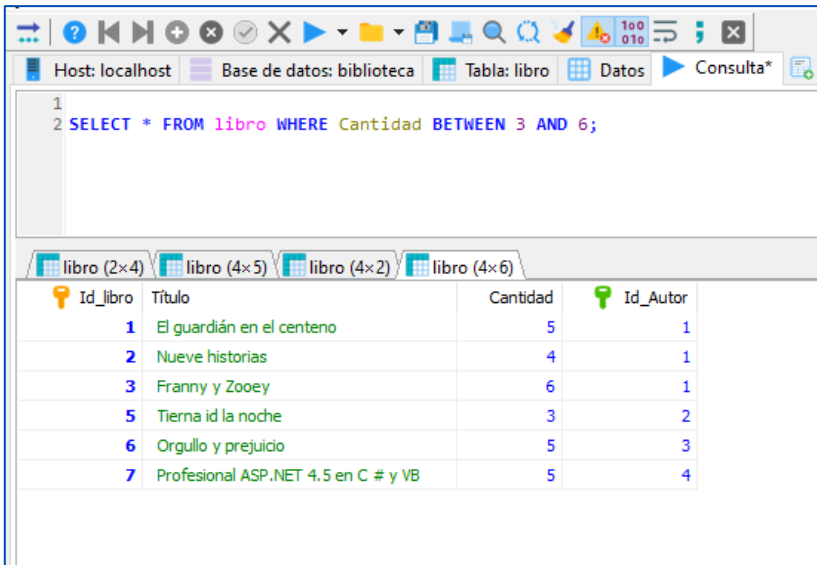
Ahora vamos a utilizar el operador **“menor que (<) y mayor que (>)**, que sirven principalmente para establecer parámetros de cantidad.

Por ejemplo: Se requiere saber la cantidad de libros que hay en existencia tomando en cuenta el parámetro 5. Cuando utilizamos el operador **>**, **se mostrarán los registros donde haya más de cinco libros en existencia, de igual manera utilizando <**, se verán aquellos donde hay menos de cinco.



En ambos casos no se muestra la cantidad de 5 ejemplares, si deseas que se incluya debes de usar `<=` o `>=`



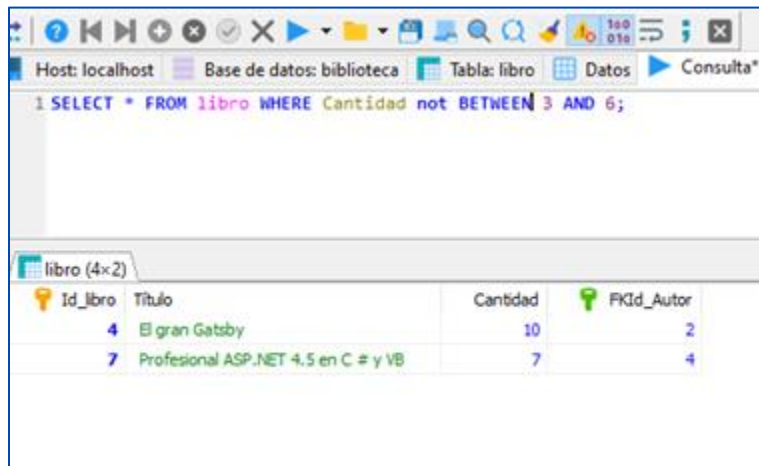


**BETWEEN** (entre) permite visualizar datos estableciendo un parámetro inicial y uno final. Verifica la sintaxis, encontrarás la palabra between y nuevamente la palabra AND, pero en este caso no como una condición, sino determinando el rango entre el cual se visualizarán los registros. Si quieres visualizar los libros que

tienen entre tres y seis unidades de existencia debes utilizar el operador de rango **BETWEEN**.

**Select \* from libro where Cantidad between 3 and 6**

También es posible utilizar este operador en su modo negativo NOT BETWEEN (no entre), que te mostrará todo, excepto lo que especificas en el rango. Es la misma sintaxis, únicamente se agrega NOT. El resultado es como se muestra en la siguiente captura:



En los ejemplos anteriores puedes observar que puedes consultar el registro de una tabla, especificando el campo y el registro exacto que estás buscando, sin embargo algunas veces no recuerdas exactamente el nombre que deseas ver, por tal motivo es necesario utilizar **LIKE** (operador de coincidencia), que se puede definir con los siguientes términos: “como” o “se parece a...”



No se quiere especificar las palabras exactas, lo que se quiere es que la primera palabra del Título sea “Orgullo”.

Consulta SQL:

**SELECT** Es la sentencia que vas a aplicar, en este caso seleccionar.

**\*** El asterisco indica que deseas revisar todos los campos de la tabla.

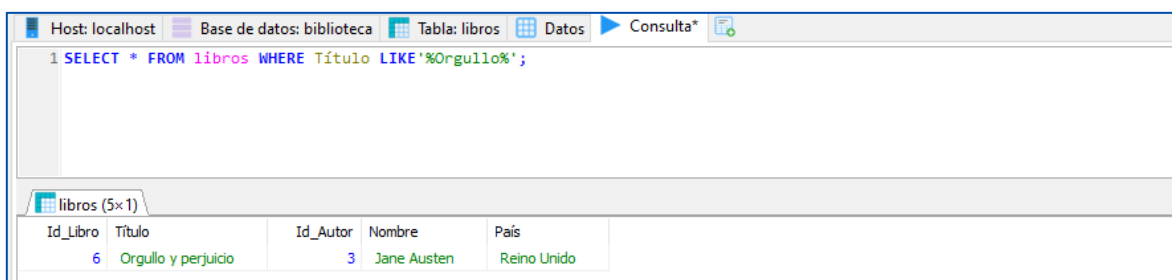
**FROM** (desde) especifica de qué tabla queremos obtener los registros o datos

**Libro** Es el nombre de la tabla que se está consultado

**WHERE** Dónde, es la palabra con la que determinas que se va a establecer una condición para la búsqueda de datos, en este caso se va a especificar el título exacto que se busca.

**Título LIKE '%Orgullo%'** Indica que en el campo título vamos a buscar cualquier registro que incluya la palabra “Orgullo”

**El caracter %** te servirá como comodín para buscar las coincidencias con la palabra que estás escribiendo en la consulta. Vamos a verlo de forma gráfica.



Si en la tabla existiera otro libro que en el título llevara la palabra “orgullo”, entonces observarías ambos registros al realizar la consulta.

Al igual que en varias sentencias, también para LIKE, existe la negación **NOT LIKE (diferente a)**, que quiere decir que podremos visualizar todos los registros que no están dentro del parámetro de comparación contemplado. La sintaxis es la misma de las sentencias anteriores, únicamente se agrega la palabra NOT.

### Consultas multitaslas

Así como podemos condicionar los campos que queremos ver, también podemos filtrar los campos que tenemos en diferentes tablas y uniéndolos en una sola consulta.

Tenemos nuestra tabla con el nombre de “libro” y otra llamada “autores”. Para poder visualizar los campos de ambas tablas en una sola consulta, debes incluir los nombres de ambas después del “from”.

```
SELECT * FROM libro, autores;
```

Id_libro	Título	Cantidad	Id_Autor	Nombre	País
1	El guardián en el centeno	5	1	JD Salinger	Estados Unidos
2	Nueve historias	5	1	JD Salinger	Estados Unidos
3	Franny y Zooey	5	1	JD Salinger	Estados Unidos
4	El gran Gatsby	5	1	JD Salinger	Estados Unidos
5	Tierra id la noche	5	1	JD Salinger	Estados Unidos
6	Orgullo y prejuicio	5	1	JD Salinger	Estados Unidos
7	Profesional ASP.NET 4.5 en C # y VB	5	1	JD Salinger	Estados Unidos
1	El guardián en el centeno	5	2	F. Scott. Fitzgerald	Estados Unidos
2	Nueve historias	5	2	F. Scott. Fitzgerald	Estados Unidos
3	Franny y Zooey	5	2	F. Scott. Fitzgerald	Estados Unidos
4	El gran Gatsby	5	2	F. Scott. Fitzgerald	Estados Unidos
5	Tierra id la noche	5	2	F. Scott. Fitzgerald	Estados Unidos
6	Orgullo y prejuicio	5	2	F. Scott. Fitzgerald	Estados Unidos
7	Profesional ASP.NET 4.5 en C # y VB	5	2	F. Scott. Fitzgerald	Estados Unidos
1	El guardián en el centeno	5	3	Jane Austen	Reino Unido
2	Nueve historias	5	3	Jane Austen	Reino Unido
3	Franny y Zooey	5	3	Jane Austen	Reino Unido

El problema que se presenta en este ejemplo es la duplicidad de registros. Por ejemplo, puedes observar que un autor se repite 7 veces ya que existen 7 títulos de libros.

Id_libro	Título	Cantidad	Id_Autor	Nombre	País
1	El guardián en el centeno	5	1	JD Salinger	Estados Unidos
2	Nueve historias	5	1	JD Salinger	Estados Unidos
3	Franny y Zooey	5	1	JD Salinger	Estados Unidos
4	El gran Gatsby	5	1	JD Salinger	Estados Unidos
5	Tierra id la noche	5	1	JD Salinger	Estados Unidos
6	Orgullo y prejuicio	5	1	JD Salinger	Estados Unidos
7	Profesional ASP.NET 4.5 en C # y VB	5	1	JD Salinger	Estados Unidos
1	El guardián en el centeno	5	2	F. Scott. Fitzgerald	Estados Unidos
2	Nueve historias	5	2	F. Scott. Fitzgerald	Estados Unidos
3	Franny y Zooey	5	2	F. Scott. Fitzgerald	Estados Unidos
4	El gran Gatsby	5	2	F. Scott. Fitzgerald	Estados Unidos
5	Tierra id la noche	5	2	F. Scott. Fitzgerald	Estados Unidos
6	Orgullo y prejuicio	5	2	F. Scott. Fitzgerald	Estados Unidos
7	Profesional ASP.NET 4.5 en C # y VB	5	2	F. Scott. Fitzgerald	Estados Unidos
1	El guardián en el centeno	5	3	Jane Austen	Reino Unido
2	Nueve historias	5	3	Jane Austen	Reino Unido
3	Franny y Zooey	5	3	Jane Austen	Reino Unido

Para evitar lo anterior necesitas indicar debe ser la unión entre las dos tablas, para ello requieres usar la sentencia **INNER JOIN** (unirse entre), sirve principalmente registros de dos o más tablas, cuando existe un campo en común.

**SELECT \* FROM libro INNER JOIN autores ON libro.FKId\_Autor = autores.Id\_Autor;**

En la consulta, estas indicando que requieres unir la tabla “libro” y la tabla “autores” y que se muestren a través de un campo en común que en este caso es el Id\_Autor, ya que aparece en ambas tablas.

```

1 SELECT * FROM libro
2 INNER JOIN autores ON libro.FKId_Autor = autores.Id_Autor;

```

Id_libro	Título	Cantidad	FKId_Autor	Id_Autor	Nombre	País
1	El Guardian en el centeno	5	1	1	JD Salinger	Estados Unidos
2	Nueve historias	4	1	1	JD Salinger	Estados Unidos
3	Franny y Zooney	6	1	1	JD Salinger	Estados Unidos
4	El gran Gatsby	10	2	2	F. Scott. Fitzgerald	Estados Unidos
5	Tierra id la noche	6	2	2	F. Scott. Fitzgerald	Estados Unidos
6	Orgullo y prejuicio	5	3	3	Jane Austen	Reino Unido
7	Profesional ASP.NET 4.5 en C # y VB	7	4	4	Scott Hanselman	Estados Unidos

En el resultado puedes observar que los libros se ordenan o agrupan de acuerdo con su autor.

Incluso podemos hacer más eficiente la presentación, mostrando únicamente los datos que queremos visualizar.

```

1 SELECT libro.Id_libro, libro.Título, autores.Id_Autor, autores.Nombre FROM libro
2 INNER JOIN autores ON libro.FKId_Autor = autores.Id_Autor;

```

Id_libro	Título	Id_Autor	Nombre
1	El Guardian en el centeno	1	JD Salinger
2	Nueve historias	1	JD Salinger
3	Franny y Zooney	1	JD Salinger
4	El gran Gatsby	2	F. Scott. Fitzgerald
5	Tierra id la noche	2	F. Scott. Fitzgerald
6	Orgullo y prejuicio	3	Jane Austen
7	Profesional ASP.NET 4.5 en C # y VB	4	Scott Hanselman

Puedes observar en esta consulta que están agrupados igual que la anterior, pero únicamente se muestran los campos que se especifican.

Además de las condiciones y variantes ya mencionadas, existe una amplia base de datos de operadores y herramientas para poder establecer criterios de visualización de registros de acuerdo con la necesidad que presente el usuario, de acuerdo con la situación que el programa de estudios marca, atendiendo a la competencia profesional planteada al inicio de este documento.

## Referencias

- Catherine M., R. (2009). Base de Datos. McGRAW-HILL INTERAMERICANA EDITORES, S.A. de C.V.



### Actividad de aprendizaje

Ordena y escribe los datos para que la sintaxis sea correcta.

FROM	campos	tabla_de_datos	WHERE	SELECT	condiciones
------	--------	----------------	-------	--------	-------------

FROM	Nombre_tabla	SELECT	'	%	*
Palabra_en_común	'	%	LIKE	WHERE	campo

FROM	WHERE	between	fin_rango	Nombre_tabla
inicio_rango	campo	and	SELECT	*

**Escribe las sintaxis de consulta que corresponda a cada situación.**

En la base de datos de una biblioteca, se tiene una tabla llamada "libro" (Id\_libro, Título, Cantidad), la cual contiene el campo o columna llamada "Cantidad", que expresa el número de existencias de cada ejemplar. Un usuario requiere una consulta que le muestre los títulos de libros que tengan entre 5 y 10 ejemplares.

---

---

---

En una base de datos de una tienda de abarrotes, existe una tabla llamada "productos" (Id\_Producto, Nombre, Precio, Cantidad, fk\_Id\_Categoría), que contiene un campo que es una llave llamada fk\_Id\_Categoría, relacionada con otra tabla con el nombre de "categoría" con su columna "Id\_Categoría", que es llave primaria. El cliente solicita que su base de datos permita los productos de la primera tabla y a qué categoría pertenecen.

---

---

---

En una base de datos de una escuela, hay una tabla llamada "alumnos" (Id\_Alumno, Nombre, Domicilio, Domicilio, Num\_Control), necesitas ver todos los registros y todos los campos que los nombres coincidan con "Roberto".

---

---

---



## Autoevaluación

Indicadores	Lo puedo hacer	Tengo dudas	Necesito trabajar más
Comprendo que son necesarias las consultas para poder brindar a un usuario de bases de datos, datos precisos, resultados ordenados e información específica.			
Identifico con claridad la aplicación de la sentencia WHERE.			
Comprendo la sintaxis para visualizar registros que tengan coincidencias a través de LIKE			
Sé cómo aplicar la sintaxis LIKE en diversas situaciones.			
Entiendo la manera de unir tablas para visualizar registros ordenados utilizando INNER JOIN:			
Soy capaz de generar consultas condicionadas con cada una de las variantes mencionadas en la lección. SELECT, FROM, WHERE, LIKE, AND, INNER JOIN.			

# Genera y modifica consultas de bases de datos



## Contextualizando



Las farmacias son comercios en los que se manejan una gran cantidad de productos, tienen diversos medicamentos de diferentes marcas y en distintas presentaciones.

Cuando has acudido a comprar un medicamento ¿has visto cómo los buscan? ¿crees que sepan de memoria en dónde están todos los tipos de medicamentos? ¿La persona que te atiende sabrá cuántas piezas de cada medicamento hay en la tienda? ¿cómo crees que organizan las medicinas? ¿Crees que estén clasificadas por el tipo de enfermedad que curan? ¿o por tipo de presentación de las medicinas: pastillas, jarabes, inyectables? ¿Crees que sea fácil ubicar si tienen el medicamento y en qué lugar de la tienda está? ¿Por qué?



## ¡Vamos a aprender!

Las bases de datos en la actualidad se han convertido, en un elemento muy importante de nuestra vida diaria, están presente en muchos de los lugares a donde vamos, por ejemplo, en una farmacia, una ferretería, o una zapatería, en ellas se almacena grandes cantidades de información para facilitar el acceso a los productos y la información de estos.

Imagina que una biblioteca tiene más de 1000 libros ¿cómo se ordenan los libros? ¿Una base de datos digital reduce el tiempo de búsqueda del encargado? ¿Si no tuvieran base de datos, cuánto tiempo crees que se toma el encargado en encontrar un libro?

En la lección anterior tuviste la oportunidad de conocer de manera muy clara las operaciones básicas para trabajar una base de datos, ahora que has visto su utilidad, aprenderás a hacer uso de las consultas, desde su forma básica hasta algo más especializado, que te permita acceder a la información de manera óptima.

Una vez que la base de datos cuenta con registros (información), puedes consultarlos o acceder a ellos realizando cualquiera de las siguientes acciones:

- Consultar uno o varios registros a la vez con ciertas características.
- Cálculos matemáticos para obtener resultados o estadísticas.



Para **consultar un registro**, debes hacer uso de los llamados operadores relacionales, que son los que nos permiten comparar dos valores y que te ayudan a completar una búsqueda.

Para **realizar algún cálculo matemático** (campo calculado), harás uso de los operadores aritméticos, que te permiten hacer operaciones matemáticas. Algunos de ellos ya los conoces, como la suma (+), resta (-) y multiplicación (\*), con la ayuda de estos, el lenguaje SQL y los campos, obtendrás algunos resultados o valores estadísticos deseados.

### Operadores aritméticos

Se puede tomar el ejemplo de la biblioteca que has revisado en las lecciones anteriores. Al momento de querer mostrar los resultados de la consulta que hayas realizado, es necesario incluir los operadores aritméticos, si es que con ellos se requiere realizar una operación matemática, de lo contrario no es imprescindible hacer uso de ellos.

Operador	Operación que realiza	Ejemplo de su uso
+	Suma	Select *, (campo1+campo2) from tabla
-	Resta	Select *, (campo1-campo2) from tabla
*	Multiplicación	Select *, (campo1*campo2) from tabla
/	División	Select *, (campo1/campo2) from tabla
Mod	Modulo (Residuo)	Select *, (campo1 mod campo2) from tabla
Sqrt	Raíz Cuadrada	Select *, sqrt(campo) from tabla
^	Potencia	Select *, (campo1 ^ 2) from tabla
()	Prioridad de operación	Select *, (campo1+campo2) /2 from tabla

Donde:

<b>SELECT</b>	Es la sentencia que vas a aplicar, en este caso <b>seleccionar</b>
<b>*</b>	El asterisco indica que deseas revisar todos los campos de la tabla.
<b>FROM</b>	(desde) especifica de qué tabla queremos obtener los registros o datos.

Los operadores aritméticos son utilizados únicamente cuando se requiere hacer una operación matemática para obtener el resultado que desees.

Ejemplo:

Un docente requiere los promedios de las tres evaluaciones parciales de sus estudiantes. Para ejecutar la consulta, únicamente es necesario sumar las tres calificaciones utilizando el operador aritmético + y después el de división para que se obtenga el resultado esperado, como se muestra en la siguiente línea de código SQL.

**Select \*, (parcial1-parcial2+parcial3)/3 from alumnos;**

Utilizando el ejemplo de la biblioteca, pudiera ser hacer la suma de los libros de ciertas categorías u otras operaciones que el usuario requiera.

Ahora continuemos con los **operadores relacionales**:

Operador	Operación que realiza	Ejemplo de su uso
>	Muestra el valor mayor	Select * from tabla where campo>valor
<	Muestra el valor menor	Select * from tabla where campo<valor
=	Muestra el valor igual	Select * from tabla where campo=valor
>=	Muestra los valores mayor o igual	Select * from tabla where campo>=valor
<=	Muestra los valores menor o igual	Select * from tabla where campo<=valor
<>	Muestra los valores diferentes	Select * from tabla where campo<>valor
Like	Muestra los valores que comiencen, terminen o contengan con un valor	Select * from tabla where like campo 'a*' Select * from tabla where like campo '*a' Select * from tabla where like campo '*a*'
not	Muestra todos menos ese valor	Select * from tabla where like campo <>'valor'

Al igual que en el ejemplo anterior utilizaste **select \*** y **from tabla** ahora agregarás la sentencia **where** (cuando), la cual te ayuda a poner la condición de filtrado de la consulta, con la ayuda de los campos, valores y operadores relacionales.

Una vez que has aprendido a realizar campos calculados en MySQL, ahora aprenderás a ordenar los registros de la base de datos.

El orden es importante ya que, en bases de datos grandes, encontrar un registro o dato en una base de datos desordenada es como buscar una aguja en un pajar.

Para ordenar registros en MySQL utilizarás la instrucción **Order By** que en español significa **Ordenar por** se compone de lo siguiente:

**Order by    Nombre    ASC**

**Ordenar por + Campo a Ordenar + Tipo de Ordenación**

Donde en tipo de Ordenación existen 2 tipos:

- Asc ascendente
- Desc descendente

Ejemplo:

El señor Martines quiere saber cuáles son todas las personas mayores de 18 años, inscritas a su negocio ordenados de la A a la Z por su nombre.

La consulta para resolver esto sería:

**Select \* from negocio where edad>18 order by nombre asc.**

## Referencias

- Olivier Heurtel. (2014). Php y MySQL. En Php y MySQL(20-50). ENI.

Imágenes tomadas de:

- <https://pixabay.com/es/>



## Actividad de aprendizaje

Lee el siguiente caso y realiza lo que se solicita.

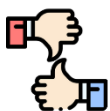
El encargado de la biblioteca de tu escuela quiere saber el costo con IVA de los libros que se muestran en la tabla para poder solicitar el recurso a la dirección y realizar la compra.

Ayúdalo creando una consulta con un campo para agregar el IVA a todos los libros.

Con la siguiente tabla, genera la instrucción SQL que le permita obtener el resultado deseado.

Libro	Año	Editorial	Precio
Física 3	2020	Pearson	\$420
Algebra	2019	Patria	\$410
Física 2	2019	Pearson	\$400
Harry Potter y la piedra filosofal	2020	Penguin	\$600
Física 1	2018	Pearson	\$400

También requiere ordenar los registros de los libros de forma ascendente de acuerdo con su precio. Ayúdalo creando una consulta que atienda la solicitud.



## Autoevaluación

Indicadores	Lo puedo hacer	Tengo dudas	Necesito trabajar más
Soy capaz de utilizar los operadores aritméticos válidos en MySQL			
Soy capaz de saber cuándo utilizar los operadores relacionales en MySQL			
Soy capaz de crear campos calculados mediante operadores aritméticos en MySQL			
Reconozco la estructura de un campo calculado en MySQL			
Soy capaz de realizar ordenación de registros mediante instrucciones SQL			

# Introducción a HTML



En la actualidad, es muy común escuchar hablar del Internet, aunque sus inicios se remontan a los años 60, no es hasta los años 2000 en que estuvo al alcance de la mayoría. ¿Alguna vez te has preguntado cómo era la vida de los jóvenes cuando no existía internet?

Antes, para mantenerte en contacto con tus amigos o con tu pareja tenías que visitarlos o hacer llamadas telefónicas desde tu casa o de un teléfono público; para encontrar música se compraban discos o video casete y escuchabas la radio; todos imprimían sus fotografías para poder mostrarlas a sus amigos y familiares; para conocer las tendencias de moda, maquillaje y las noticias de tus artistas favoritos se compraban revistas; si querías resolver una duda tenías que investigar debías acudir a una biblioteca o si querías enviar algún mensaje o dinero a algún familiar lo hacías a través del correo postal.



¿Cómo te comunicas con tus amigos y familiares? ¿Si tienes que hacer una investigación para alguna de tus materias, cómo lo haces? ¿Cómo comprar una película? ¿Cómo escuchas música? ¿Cómo envías una carta o mensaje?



Hoy en día a través de las redes sociales puedes compartir con tus amigos tus fotos, videos, pensamientos, así como conocer personas, encontrar contenido de todo tipo, todo ello, gracias al Internet que vino a facilitar y darnos acceso a mucha información. Con la llegada de los Smartphone ha sido aún más sencillo navegar en la red y encontrar sitios web de cualquier tema de interés, pero ¿Alguna vez te has preguntado cómo se construyen esos sitios web? En esta lección aprenderás cómo se construye un sitio web y sus principales características.

Para empezar se debe planificar la estructura del sitio a través de un mapa general en el que se definan algunas características como:

- El objetivo del sitio web
- Tipo de público al que va dirigido
- Los contenidos que tendrá (secciones)
- Estilo del sitio (colores, diseño)

Una vez que se termina el proceso de planeación, es momento de conocer el lenguaje que se emplea para el desarrollo, construcción o escritura de sitios web, se llama **HTML** (HyperText Markup Language), en español se traduce Lenguaje de Marcas de Hipertexto. Este lenguaje está compuesto por una serie de instrucciones a las que se les denomina **etiquetas** o **marcas**.

Todos los archivos HTML son sólo texto, así que se necesita un editor de texto, aplicación o programa que permita escribir texto plano sin agregar ningún formato específico, por ejemplo un bloc de notas.

**Una vez escrito el texto, se guarda con la extensión “.html” y al abrir el archivo (posicionar el mouse sobre dicho archivo, dar clic con el botón derecho y seleccionar la opción abrir), es el navegador el que interpreta y da forma en la pantalla para que el usuario pueda ver lo que se ha diseñado.**

El lenguaje HTML permite crear muchos elementos y hacer dinámico y atractivo un sitio web, todos los elementos que imagines se establecen a través de **etiquetas** o **líneas de escritura** y existen etiquetas para insertar imágenes, figuras, para agregar colores de fondo o color de la fuente, para insertar tablas y formularios, para crear listas, y para hipervínculos que nos permiten dirigirnos a otras páginas, entre otros. Es el lenguaje HTML el que permite crear páginas web, darles estructura y contenido. Existen diferentes tipos de etiquetas que se utilizan para crear un sitio web.

Etiqueta	Descripción
<p>	Define una parte que debe mostrarse como párrafo.
<li>	Define un elemento de una lista numerada.
<dl>	Define una lista de términos y sus significados asociados.
<figure>	Representa una figura ilustrada como parte del documento.
<figcaption>	Representa la leyenda de la figura.

## Etiquetas de texto

Etiqueta	Descripción
<a>	Representa un hiperenlace
<em>	Representa un texto enfatizado
<strong>	Representa un texto importante
<s>	Representa contenido que no es exacto, tiene estilo tachado
<cite>	Representa el título de una obra.
<time>	Representa un valor de fecha y hora.
<kdb>	Representa el texto que debe introducir o la tecla que debe pulsar el usuario.
<i>	Muestra el texto marcado con un estilo en cursiva o itálica.
<b>	Muestra el texto marcado con un estilo en negrita.
<u>	Muestra el texto subrayado.
<mark>	Representa un texto marcado o resaltado como referencia o anotación, debido a su relevancia o importancia.
 	Representa un salto de línea.

En una página web también puedes programar un formulario para obtener información de los usuarios y, existen etiquetas para poder programarlos.

## Etiquetas para la creación de formularios

Etiqueta	Descripción
<form>	Define un formulario. El principio y final de un formulario se define con las etiquetas <form> y </form>.
<fieldset>	Permite organizar en grupos los campos de un formulario.
<legend>	Representa el título de un <fieldset>.
<label>	Representa una etiqueta para un elemento en una interfaz de usuario. Este puede estar asociado con un control mediante la utilización del atributo.
<input>	Se usa para crear controles interactivos, que son elementos que pueden cambiar de acuerdo con las elecciones del usuario.



< button>	Representa un botón (elemento cliqueable que puede ser utilizado en formularios o en cualquier parte de la página)
<option>	Representa una opción en un elemento <select> o <datalist>.
<select>	Representa un elemento de control que permite la selección entre un conjunto de opciones <option>.
<optgroup>	Representa un conjunto de opciones, agrupadas lógicamente.
<datalist>	Representa un elemento de control que permite la selección entre un conjunto de opciones <option>.
<textarea>	Representa un elemento de control de edición de texto multilínea.
<output>	Representa el resultado de un cálculo.

Las tablas de contenido son un medio para organizar información en filas y columnas que te permiten almacenar los datos de forma ordenada.

### Etiquetas para la creación de tablas de contenido

Etiqueta	Descripción
<table>	Representa datos con más de una dimensión. El principio y final de una tabla se define con las etiquetas <table> y </table>.
<caption>	Representa el título de una tabla.
<tr>	Representa una fila. Las filas se engloban con las etiquetas <tr> </tr>
<td>	Representa una celda de datos en una tabla.
<th>	Representa el encabezado en una tabla.
<colgrup>	Representa un conjunto de una o más columnas de una tabla.
<col>	Representa una columna de una tabla.

Un documento HTML está delimitado por las etiquetas <html> y </html>. Dentro de este documento, podemos asimismo distinguir dos partes principales:

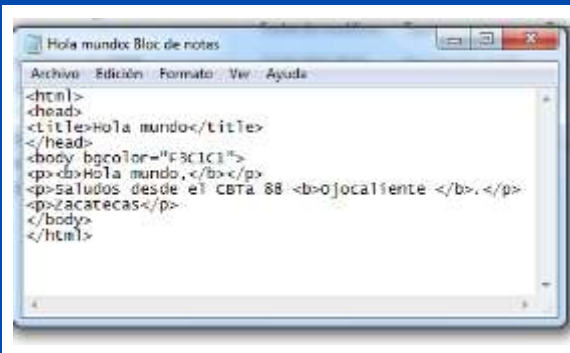
1. El encabezado o cabecera, delimitado por <head> y </head> donde colocaremos etiquetas de índole informativo como por ejemplo el título de la página.
2. El cuerpo, delimitado por las etiquetas <body> y </body>, que será donde colocaremos el texto e imágenes delimitados a su vez por otras etiquetas.

Entonces un documento HTML tendría la siguiente estructura:

```
<html>
  <head>
    Etiquetas y contenidos del encabezado
  </head>
  <body>
    Etiquetas y contenidos del cuerpo
    Texto e imágenes
  </body>
</html>
```

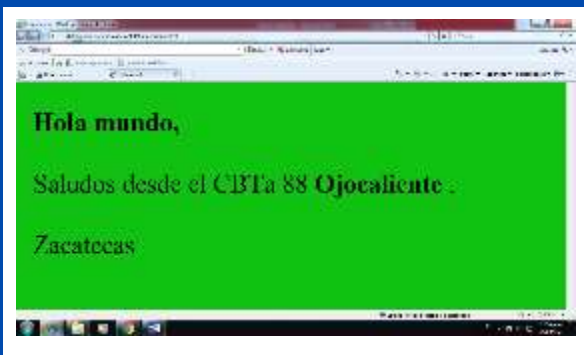
Las etiquetas tienen una manera de expresarse a la que se le llama sintaxis, se escriben entre los símbolos "<" y ">" y para cerrarse se escribe el nombre de la etiqueta precedido con el símbolo "/". También existen los atributos que nos permiten personalizar las etiquetas.

Así se ve escrito en el editor de texto

A screenshot of a text editor window titled "Hola mundo: Bloc de notes". The window contains the following HTML code:

```
<html>
<head>
<title>Hola mundo</title>
</head>
<body bgcolor="#F3C1C1">
<p><b>Hola mundo,</b></p>
<p>Saludos desde el CBTA 88 <b>ojocaliente </b>.</p>
<p>zacatecas</p>
</body>
</html>
```

Así se muestra el archivo abierto



A continuación se muestra un ejemplo sencillo de código HTML:

```
<html>
  <body>
    <p>Esto es un párrafo. Bienvenidos a esta página web.</p>
  </body>
</html>
```

**<b> HOLA MUNDO </b>**

Etiqueta de apertura      Contenido      Etiqueta de cierre

-----Elemento-----  
-----

## Referencias

Imágenes tomadas de:

- <https://pixabay.com/es/>
- <https://www.canva.com/>



***¡Ponlo en práctica!***

Es muy cómodo navegar por internet de una página a otra en busca de la información que te interese. Generalmente, las páginas suelen ser muy atractivas en cuanto a su diseño, colores, las imágenes que muestran pero hay un trabajo de programación detrás de ellas.

Vas a conocer cómo se ven desde su escritura siguiendo el procedimiento que se muestra a continuación:

1. Entra a internet desde un equipo de cómputo.
2. Busca una página de tu interés.
3. Presionar Ctrl+U y observa cómo se abre una pestaña con el código fuente HTML.

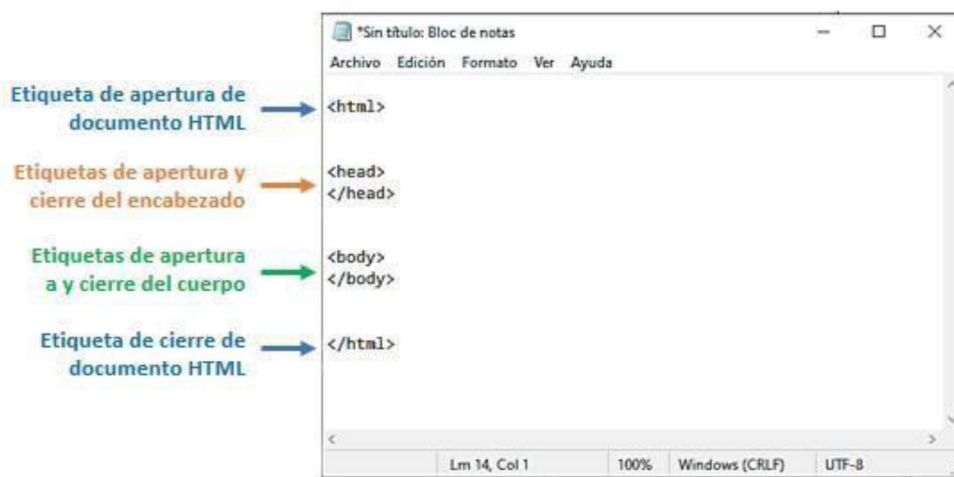
4. Observa el código y localiza las etiquetas que se utilizan. Trata de identificar las que revisaste en esta lección.

También hay otra forma más rápida consiste en hacer clic con el botón derecho del ratón en cualquier punto del área donde el navegador muestra la página web y elegir «Ver código fuente de la página».

Aparte de poder ver el código fuente HTML de una página web con las opciones antes descritas, Internet Explorer, Firefox y Google Chrome tienen también unas herramientas conocidas como inspectores de página que se pueden activar con F12. Con estas herramientas es posible visualizar una página web y seleccionar dentro de ella un elemento concreto del cual queremos conocer cuál es el código HTML con el que está hecho señalando el elemento en cuestión simplemente con el ratón. Al hacer esto, el código se mostrará en un área especial dentro del navegador en el que el usuario podrá ver el código HTML en cuestión, además, de las reglas CSS que aplican a ese código HTML en concreto. Este tipo de análisis resulta sumamente instructivo para aprender a desarrollar en HTML.

Ahora que has visto el código HTML de una página web realizarás una práctica sencilla.

1. Abre el bloc de notas de tu computadora. Se que encuentras en el Menú Inicio > Programas > Accesorios > Bloc de notas.
2. Una vez que tengas abierto el bloc de notas escribe cuidadosamente las siguientes etiquetas:
  - Etiqueta de apertura de documento HTML: <html>
  - Etiquetas de apertura y cierre del encabezado: <head> y </head>
  - Etiquetas de apertura y cierre del cuerpo: <body> y </body>
  - Etiqueta de cierre de documento HTML </html>



3. Encabezado. Escribe cuidadosamente entre las etiquetas de apertura y cierre del encabezado las siguientes etiquetas y texto para crear el tema principal:

```
<h4>
```

HTML es un lenguaje que se utiliza para el desarrollo de páginas de Internet

```
</h4>
```



4. Cuerpo. Escribe cuidadosamente entre las etiquetas de apertura y cierre del cuerpo las siguientes etiquetas y texto para crear el tema principal, como se muestra.

```
<body bgcolor="F3C1C1">
```

```
<p><b>BIENVENIDOS</b></p>
```

```
<p>- Un curso donde será una manera práctica de aprender -</b>.</p>
```

```
<p>La excelencia no es un acto, es un hábito (Aristóteles)</p>
```

```
<h4>
```

Datos del creador

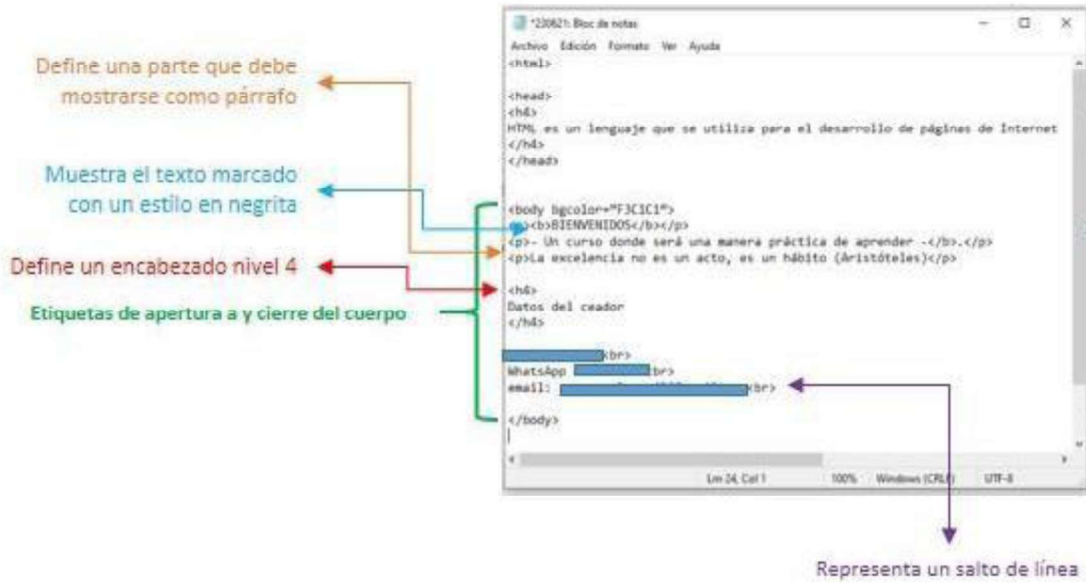
```
</h4>
```

```
Escribe aquí tu nombre<br>
```

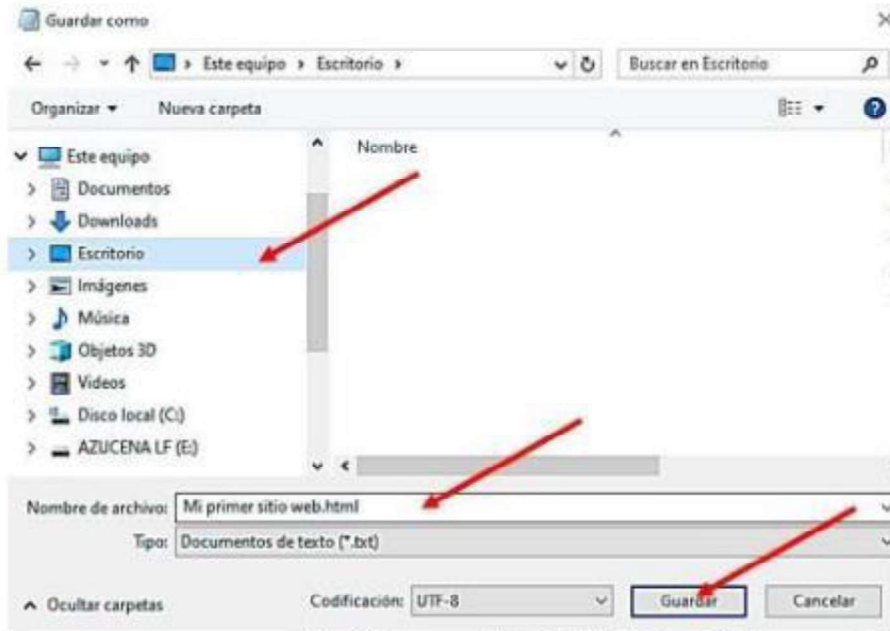
```
WhatsApp: Escribe aquí tu número telefónico<br>
```

```
email: Escribe aquí tu email<br>
```

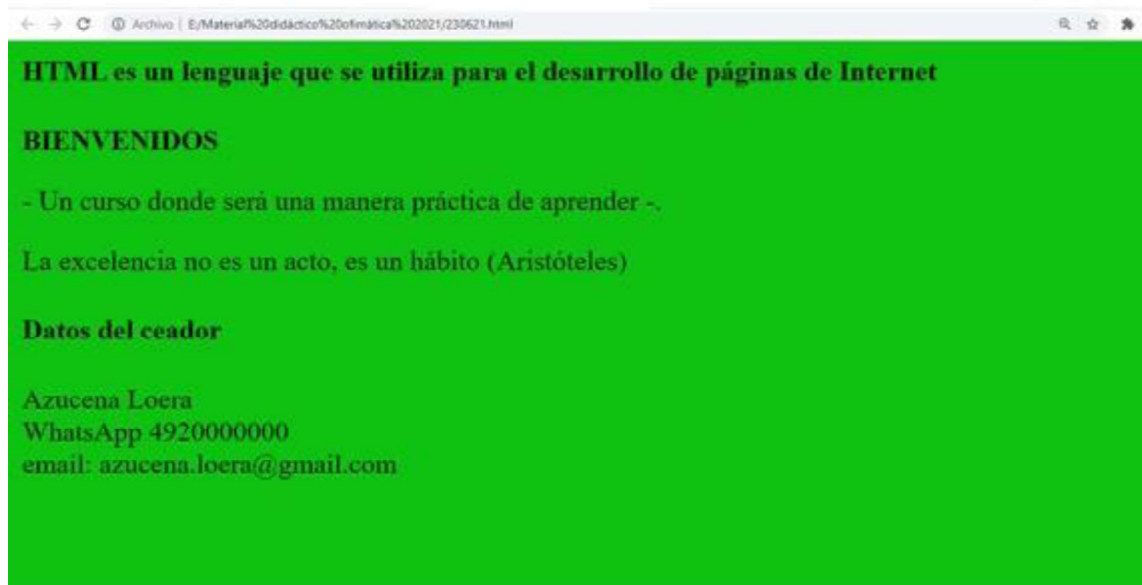
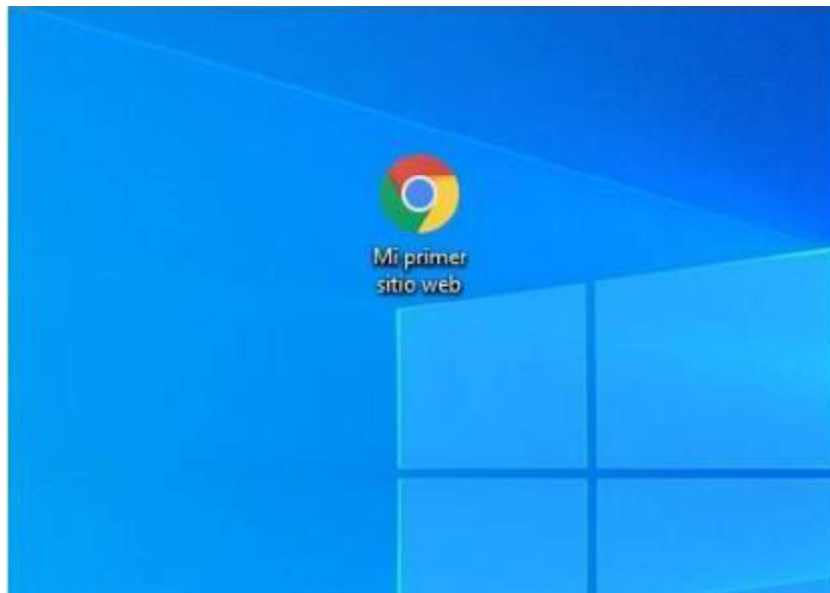
```
</body>
```



- Una vez que hayas escrito todas las etiquetas y contenido sugerido, guarda el documento dando clic en la pestaña "Archivo", clic en "Guardar como", Escritorio, Nombre del archivo "Mi primer sitio web.html" y clic en el botón guardar.



- Ahora abre el archivo que se generó y podrás ver el resultado. Solamente tienes que posicionar el cursor sobre el archivo, dar clic con el botón derecho del mouse y seleccionar la opción abrir.



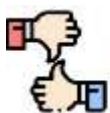




### Actividad de aprendizaje

**Lee las oraciones y subraya la respuesta correcta.**

1. Al planificar la estructura de un sitio web debes considerar:
  - a) Procesos de planeación y navegador
  - b) Objetivo del sitio web y los contenidos que tendrá
  - c) Desarrollo y construcción del sitio web
  - d) La escritura de los sitios web
  
2. Es el lenguaje que se emplea para el desarrollo, construcción o escritura de un sitio web es:
  - a) DOCX
  - b) PPTX
  - c) HTML
  - d) XLSX
  
3. Las etiquetas tienen una manera de expresarse, a la que se le llama:
  - a) Sintaxis
  - b) Nombre
  - c) Atributo
  - d) Personalizado
  
4. En un documento HTML se pueden distinguir dos partes principales que son:
  - a) El encabezado y el cuerpo
  - b) Principal y secundario
  - c) Esquemas y tablas
  - d) Descripción y artículo
  
5. Son el soporte en el que se integran los botones y las áreas de texto que se utilizan para que las personas introduzcan sus datos o que puedan elegir entre varias opciones:
  - a) Descripción
  - b) Formularios
  - c) Organizadores
  - d) Usuario



## Autoevaluación

Indicadores	Lo puedo hacer	Tengo dudas	Necesito trabajar más
Comprendo que el lenguaje que se emplea para el desarrollo, construcción o escritura de sitios web se llama HTML.			
Reconozco que lenguaje HTML está compuesto por una serie de instrucciones a las que se les llama etiquetas o marcas.			
Entiendo que las etiquetas tienen una manera de expresarse a la que se le llama sintaxis.			
Entiendo que un documento HTML está delimitado por las etiquetas <code>&lt;html&gt;</code> y <code>&lt;/html&gt;</code>			
Comprendo que las etiquetas se <b>escriben entre los símbolos “&lt;” y “&gt;”</b> y para cerrarse se escribe el nombre de la etiqueta precedido con el <b>símbolo “/”</b>			
Entiendo que en un documento HTML se pueden distinguir dos partes principales, que son el encabezado y el cuerpo.			
Soy capaz de poner en práctica el lenguaje HTML			

# Conexión con base de datos MySQL con PHP y diseño de formularios



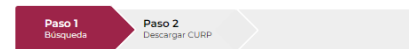
## Contextualizando

Cada día nuestra vida contempla más actividades que pueden realizarse desde un dispositivo electrónico, por ejemplo, trámites bancarios, pagos y consultas de información, productos o servicios.

Hace algunos años para obtener tu Clave Única de Registro de Población (CURP) era necesario trasladarte a la oficina correspondiente para tramitar una copia pero en la actualidad puedes hacerlo a través de una página del gobierno de México (<https://www.gob.mx/curp/>)

¿Cómo crees que funciona esta página? ¿Por qué crees que la página puede proporcionar tanta información? ¿Crees que existe una base de datos con la información de todos los ciudadanos? ¿Quiénes no podrían consultar su CURP?

## Consulta tu CURP



### Búsqueda

La consulta puede efectuarse indicando la clave CURP cuando ya la conoce o proporcionando su nombre y datos de nacimiento.

Clave Única de Registro de Población		Datos Personales	
Nombre(s)*:	<input type="text" value="Ingresa tu nombre(s)"/>	Primer apellido*:	<input type="text" value="Ingresa tu primer apellido"/>
Segundo apellido:	<input type="text" value="Ingresa tu segundo apellido"/>	Día de nacimiento*:	<input type="text" value="Seleccionar el día"/>
Mes de nacimiento*:	<input type="text" value="Seleccionar el mes"/>	Año de nacimiento*:	<input type="text" value="Ingresa el año Ej. 1943"/>
Sexo*:	<input type="text" value="Selecciona el sexo"/>	Estado*:	<input type="text" value="Selecciona el estado"/>



## ¡Vamos a aprender!

Uno de los lenguajes de programación más utilizados es el PHP, según el sitio oficial The PHP Group (2021), es un lenguaje de código abierto, adecuado para desarrollo web y que puede ser incrustado en HTML, es decir que puedes utilizar un mismo archivo y combinar código PHP con código HTML, siguiendo algunas reglas.

Cuando decimos que es un lenguaje de código abierto significa que es de uso libre y gratuito y que puede ser utilizado por cualquier persona que lo desee.

Las páginas web pueden ser de dos tipos: estáticas y dinámicas. La primeras son aquellas cuyos contenidos permanecen siempre igual, en cambio las páginas dinámicas se caracterizan porque su contenido es variable y se actualiza de acuerdo con la información que se quiera mostrar.

Existe un gran número de páginas y portales web que han sido creados con el lenguaje PHP, generalmente es utilizado para la creación de páginas dinámicas.

Tomando en cuenta el desarrollo de un sistema para una biblioteca, se está considerando que las bases de datos sean manejadas de manera dinámica por medio de un sistema Web; para ello deberás implementar la integración entre PHP (que es el lenguaje de programación que manipulará los datos en el flujo de las páginas WEB) y la base de datos que se había generado en MySQL (que es el servidor donde se almacena los datos estructurados).

MySQL es el sistema de gestión de bases de datos que trabaja con bases de datos relacionales, es decir, utiliza tablas múltiples que se interconectan entre sí para almacenar la información y organizarla correctamente.

Observa el proceso que se lleva a cabo en un **sistema web lineal**.

1. El usuario hace una solicitud de consulta de una página web.
2. Esta solicitud viaja por la nube de internet hasta que encuentra el servidor que tiene almacenado esta página.
3. El servidor busca en su almacenamiento interno la página solicitada.
4. Una vez encontrada la envía de regreso a la nube de internet hacia el usuario, si él necesita más información que se encuentra contenida en la página web recibida, volverá a hacer una solicitud de información y con esta se repetiría el mismo proceso, con esto se gastarían más datos de conexión.



Ahora, revisa el proceso de un **sistema web dinámico**.

1. El usuario realiza una solicitud en una página web.
2. La solicitud se envía vía internet para buscar el servidor que almacena los datos de la página.
3. El servidor web realiza las consultas necesarias, por medio del código PHP, en el servidor MySQL para que la información y las consultas sean almacenadas en una página dinámica HTML.
4. La página dinámica regresa al usuario pasando vía internet, pero esta página contiene más información, esta información permanece oculta hasta que el usuario

decida consultarla, estas consultas ya se realizan en el mismo cliente y ya no es necesario que vuelva a hacer una nueva solicitud de una nueva página web. Las flechas que se observan sin un número indica el procesamiento que se hace en el servidor, donde interviene el lenguaje de programación PHP, el gestor de bases de datos MySQL y la página HTML ya mencionada, entre los tres generan el resultado del que ya se habló.

Para poder desarrollar una página dinámica con el uso del lenguaje PHP con MySQL, es necesario revisar algunas definiciones para que comprendas cómo es la comunicación en un sistema web dinámico.

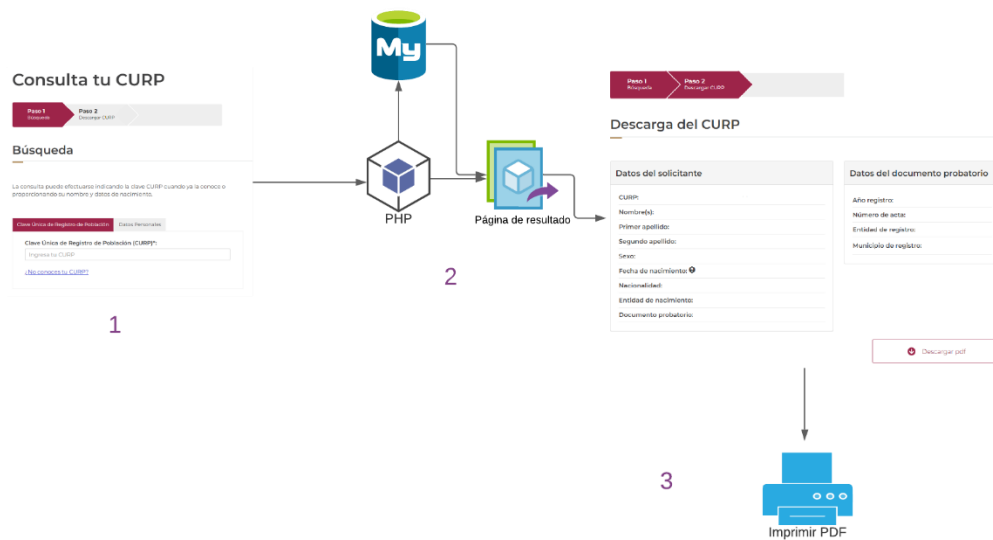


4

Tomaremos como ejemplo la página de consulta de la CURP para explicar la estructura y funcionamiento.

1. El sistema tiene dos formas de consulta: ingresar la CURP o los datos personales. El usuario puede elegir cómo realizará la búsqueda.
2. Una vez que hace la consulta al sistema cliente-servidor, el servidor realiza una consulta en la base de datos por medio de un lenguaje de programación que puede ser PHP o JavaScript. Esto también se conoce como aplicación web.
3. Una vez que se encuentra el resultado se crea una página que contiene la información necesaria para generar la CURP de manera electrónica y se puede realizar la impresión del documento.

Podrás observar que al igual que en la imagen anterior se encuentran flechas que indican el flujo de los datos, primeramente, la solicitud al servidor, luego el procesamiento de datos que es idéntico al de la imagen anterior, ya que es el trabajo que se realizaría en el ya mencionado servidor para que de él salga la respuesta que espera el usuario.



Es importante que sepas que los sistemas estáticos son lineales y no tienen interacción con el usuario, podemos ejemplificar un sitio web de tu artista favorito, ya que en él únicamente puedes leer la información, ver fotografías y videos, pero no puedes registrarte, ni interactuar con el sitio, únicamente leer. En cambio, un sistema dinámico interactúa entre cada una de las páginas que lo componen y con el usuario, por ejemplo Facebook, que puedes comentar, buscar gente, enviar y recibir información, por lo que debe contar con los siguientes elementos:

- Aplicaciones WEB:** Según Minera (2014) las aplicaciones se pueden clasificar en: portátiles, de escritorio y web. Para que una aplicación sea considerada aplicación web, esta debe ser ejecutada en el servidor web y no en el usuario, es decir, que los datos con los que se trabaja son procesados y almacenados en el servidor web, por ejemplo el sitio donde se obtiene la CURP, el proceso se ejecuta y los datos se encuentran en el servidor (computadora donde están guardados los archivos de esa página) y el usuario únicamente recibe la respuesta y el documento para imprimirse. Cabe mencionar que las aplicaciones portátiles son aquellas que pueden ser utilizadas sin instalarse en un equipo de cómputo, es decir sin ocupar espacio en disco y las de escritorio son aquellas que requieren instalarse en el equipo de cómputo y ejecutarse desde el mismo utilizando espacio en disco. Las aplicaciones (Apache e IIS son los exponentes más conocidos) se encargan de generar las respuestas, en los casos en los que es necesario, de otros actores como los mencionados antes: servidores de bases de datos (Oracle, SQL Server, MySQL, PostgreSQL, por ejemplo) y lenguajes de programación (PHP, ASP.net, JSP, o PERL, entre otros). (Minera, 2014)

Es de gran importancia que los sistemas dinámicos contengan la integración del lenguaje de PHP, este lenguaje ayudará a manipular la información que se encuentra contenida en un servidor de base de datos, el HTML como constructor visual, debe estar viene estructurado para que la información que sea manipulada sea mostrada de manera eficiente

- **Arquitectura cliente-servidor:** Es un modelo de software, en el que se determinan dos figuras: el servidor que es el que procesa, ejecuta y almacena la información y, el cliente que solicita o demanda la información.

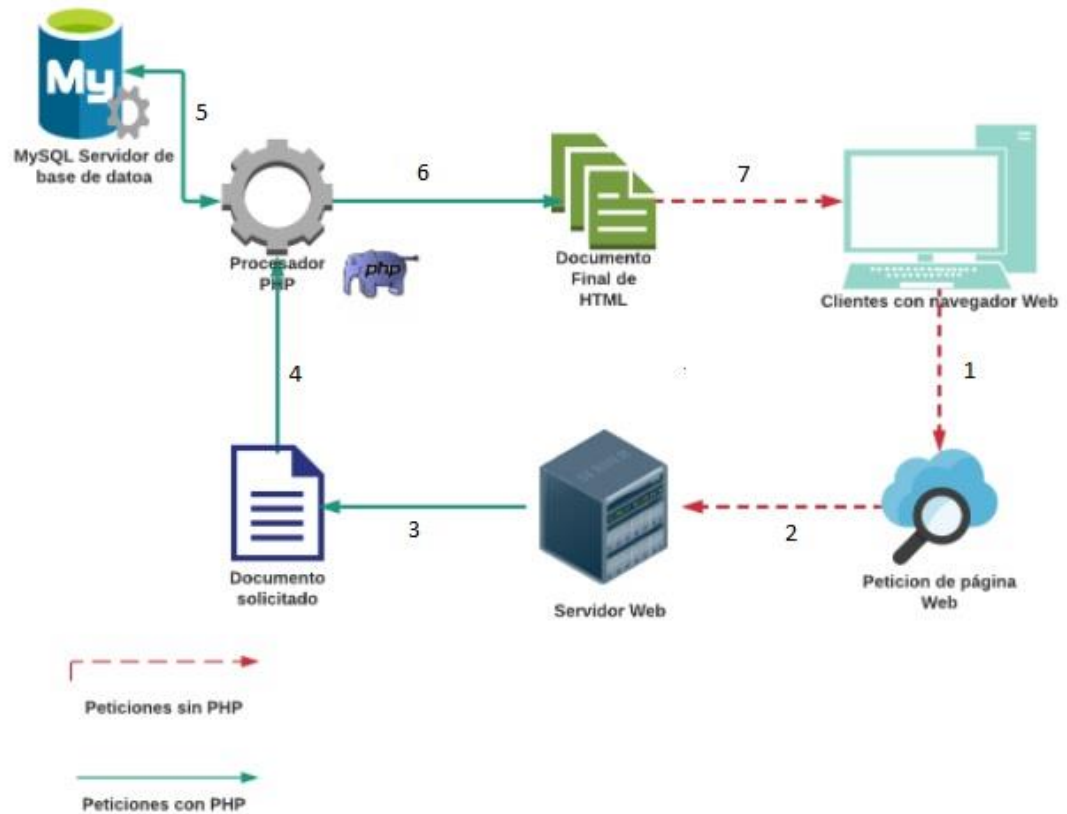
El cliente hace una petición, presionando un botón o haciendo clic en un enlace (por ejemplo, a través de un enlace), y el servidor deberá, por un procesamiento, resolver la demanda y devolver una respuesta.

La aplicación cliente es el navegador web, desde esta clase de programas accedemos a la interfaz del sistema para interactuar con él.

- **Lenguajes de script de servidor:** Las páginas web se crean mediante lenguaje HTML, éste es un lenguaje muy limitado para atender a los requerimientos que actualmente se exigen, como mostrar grandes cantidades de información almacenada en una base de datos o la generación y ejecución de procesamiento de datos. Por ello, aparecieron lenguajes y mejoras en el servidor, en donde el programador añade al código HTML códigos de otro lenguaje (PHP), pero con la diferencia, que este código no se le envía al cliente, sino que es el servidor el que lo interpreta, lo ejecuta y el cliente recibirá una página HTML normal con los resultados ya plasmados(Sánchez, 2012).

En el siguiente esquema se observa lo siguiente:

1. El cliente a través de su navegador web realiza una solicitud para acceder a alguna página web.
2. La solicitud se envía a un servidor para que en él se busque el archivo de página que se requiere.
3. Dentro del servidor se busca el archivo de página que el usuario requiere visualizar o al que solicita algún tipo de solicitud.
4. Después de encontrar el archivo, mediante el lenguaje PHP se ejecutan las peticiones planteadas por el usuario.
5. Se llevan a cabo las modificaciones y peticiones en el Gestor de Bases de Datos (MySQL), para que éste devuelva un resultado.
6. Los resultados encontrados son escritos en un documento HTML de manera que el usuario final pueda leerlos.
7. Se envía el archivo HTML a un navegador para que el usuario desde su dispositivo pueda visualizarlos.

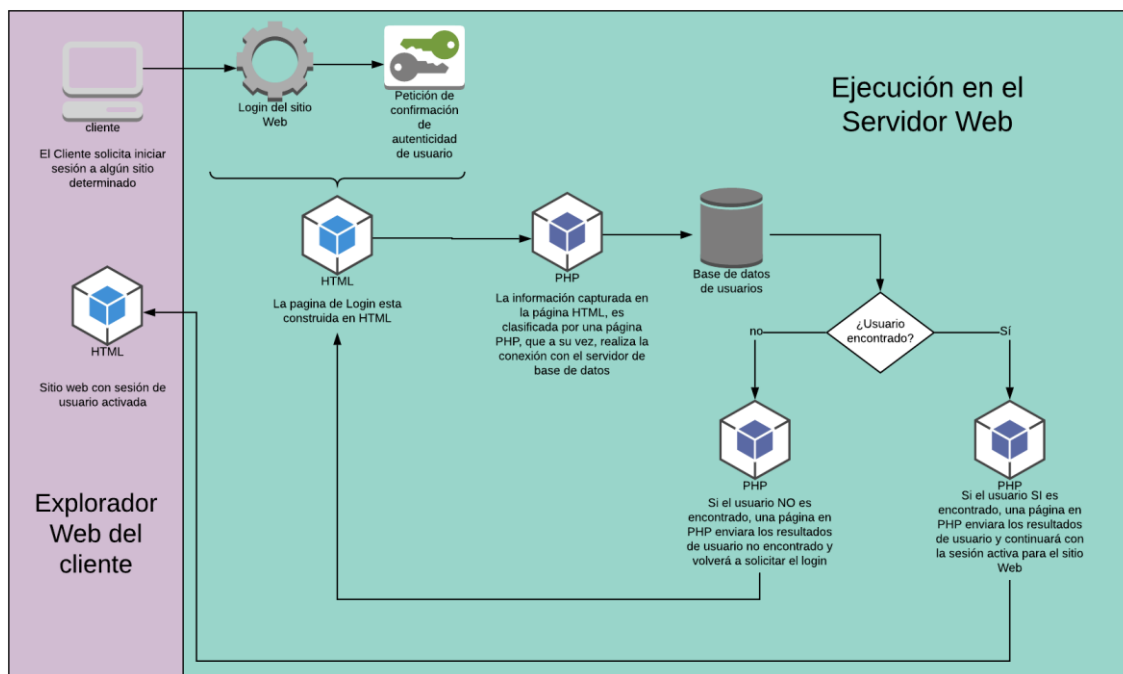


Un ejemplo básico para entender la interacción de un sistema web dinámico:

Cuando un usuario desea ingresar con su cuenta en un sistema web, como por ejemplo ingresar a una cuenta de correo electrónico o a Facebook (normalmente realizada en código HTML), se muestra un formulario de inicio de sesión; una vez que captura sus datos y presiona el botón de acceso (*login*), la información es enviada o redireccionada a una página construida en un lenguaje de programación (normalmente PHP). La página se ejecuta en el servidor web, verifica si el usuario y contraseña está almacenada en la base de datos MySQL. Cuando el usuario y contraseña están verificados habrá dos posibles escenarios:

- Si la página no encuentra el usuario y contraseña se genera una respuesta negativa por medio de una página con código PHP se crea una nueva página en HTML y vuelve a solicitar los datos ingreso del usuario y contraseña.
- Si la página encuentra los datos del usuario en la base de datos, una página con código PHP acepta su acceso y deja que el usuario ingrese al sitio web con su sesión activa.





La integración entre PHP y HTML se usa por medio de una etiqueta, en ella se integran todas las instrucciones de PHP.

- **Etiqueta PHP:** Cuando en un documento web queremos añadir código PHP se indica por esta etiqueta:

```
<?php
    //código php
?>
```

El código PHP se coloca en la zona de la página web donde más nos interese hacerlo. (Sanchez, 2012)

El cliente que está solicitando el sistema web dinámico, desea que se muestre textos propios, para ello es necesario entender cómo es que se muestra la información.

- **HTML usa PHP y PHP usa HTML:** Como se ha visto anteriormente el código PHP se incrusta dentro del código HTML. Lo interesante es que se puede incrustar código HTML en el PHP y tendría sentido ya que las funciones de escritura (**echo** y **print**) en realidad escriben hacia el resultado final, que en realidad es una página HTML por lo que se puede hacer algo como:

```
<?php
    echo "Mi nombre es <strong>Jorge</strong>";
```

```
?>
```

- **Bases de escritura:** Las normas básicas para escribir el lenguaje PHP, son las siguientes:
  - Todas las líneas de código deben de finalizar con un punto y coma
  - Se puede agrupar el código en bloques que se escriben entre llaves
  - Una línea de código se puede partir o sangrar (añadir espacios al inicio) a voluntad con el fin de que sea más legible, siempre y cuando no partamos una palabra o un valor.
  - PHP obliga a ser estricto con las mayúsculas y las minúsculas en algunos casos como el nombre de las variables; sin embargo, con las palabras reservadas del lenguaje no es estricto. Es decir, PHP entiende que **WHILE**, **while** e incluso **wHiLe** es lo mismo al ser una palabra reservada. Sin embargo **\$var** y **\$VAR** no son iguales al ser el nombre de una variable. (Sanchez, 2012)
- **Escribir en la salida:** Aunque hay muchas funciones de escritura (para escribir en lo que será la página final) las fundamentales son **echo** y **print**. **Echo** es la más utilizada y en realidad es un comando del lenguaje. Tras **echo** se pasa uno o más textos (más adelante diremos *expresiones de cadena*) que cuando son literales se escriben entre comillas. Si se usa más de un texto, se separan con comas:

```
<?php  
    Echo "Primer texto ", "segundo texto";  
?>
```

Como cualquier lenguaje de programación, se maneja información interna por medio de variables, para eso se necesita entender cómo es el manejo de las variables en PHP.

- **Introducción a las variables:** En todos los lenguajes de programación (y PHP no es una excepción) Las variables son contenedores que sirven para almacenar los datos que utiliza un programa. Dicho más sencillamente, son nombres que asociamos a determinados datos.
  - Las variables tienen un nombre (un identificador) que tiene que cumplir estas reglas:
  - Tiene que empezar con el símbolo **\$**. Ese símbolo es el que permite distinguir a una variable de otro elemento del lenguaje PHP.
  - El segundo carácter puede ser el guión bajo (**\_**) o bien una letra.
  - A partir del tercer carácter se pueden incluir números, además de letras y el guión bajo
  - No hay límite de tamaño en el nombre
  - Por supuesto el nombre de la variable no puede tener espacios en blanco (de ahí la posibilidad de utilizar el guión bajo)

Es conveniente que los nombres de las variables indiquen de la mejor forma posible su función. Es decir: `$saldo` es un buen nombre, pero `$x123` no lo es, aunque sea válido. (Sanchez, 2012)

- **Declarar:** La primera sorpresa para los programadores de lenguajes estructurados es que en PHP no es necesario declarar una variable. Simplemente se utiliza y ya está. Es decir, si queremos que la variable `$edad` valga `15`, haremos:

```
<?php
    $edad=15;
?>
```

Los textos se encierran entre comillas (simples o dobles, aunque se aconseja usar las dobles salvo cuando nos venga mejor las simples). Ejemplo:

```
<?php
    $nombre='Alejandro';
?>
```

- **Tipos de datos**
  - **Enteros:** A las variables se les puede asignar valores enteros. Los números enteros se usan tal cual. Pueden ser positivos o negativos:

```
<?php
    $n1=17;
    $n2=-175;
?>
```

- **Flotante:** Los números decimales en PHP son de tipo coma flotante. Este es un formato decimal para máquinas digitales que se manejan muy rápido por parte de un ordenador, ocupan poco en memoria, pero desgraciadamente no son exactos. Ejemplos:

```
<?php
    $n1=234.12;
    $n2=-12.3e-4; //eso es 12.3·10-4, es decir 0.00123
?>
```

- **Cadenas:** Se denomina así a los textos, que en programación se les denomina cadenas de caracteres o Strings. Se asignan a las variables entrecomillado (en simples o dobles) el texto a asignar. Ejemplo:

```
<?php
    $nombre= 'Alejandro Moreno';
?>
```

- Booleanos: Sólo pueden tomar como valores **TRUE** (verdadero) o **FALSE** (falso). Ejemplo:

```
<?php
    $verdadero=True;
    echo $verdadero; //escribe 1
?>
```

- **Operadores:** Lo habitual al programar en PHP es utilizar expresiones que permiten realizar comprobaciones o cálculos. Las expresiones dan un resultado que puede ser de cualquiera de los tipos de datos comentados anteriormente (enteros, decimales, booleanos, strings,...) (Sanchez, 2012)
  - Aritméticos  
Son:

Operador	Significado
+	Suma
-	Resta
*	Multiplicación
/	División
%	Modulo (resto)

Ejemplo de uso:

```
<?php
    $x=15.5;
    $y=2;
    echo $x+$y,"<br />"; //escribe 17.5
    echo $x-$y,"<br />"; // escribe 13.5
    echo $x*$y,"<br />"; // escribe 31
    echo $x/$y,"<br />"; //escribe 7.75
    echo $x%$y,"<br />"; //escribe 1, sólo coge la parte entera
?>
```

- Operadores condicionales: Sirven para comparar valores. Siempre devuelven valores booleanos. Son:

Condicionante	Significado
---------------	-------------

<	Menor
>	Mayor
<=	Menor o igual
>=	Mayor o igual
==	Igual, devuelve verdadero si las dos expresiones que compara son iguales
===	Equivalente, devuelve verdadero si las dos expresiones que compara son iguales y además del mismo tipo
!=	Distinto
!	No lógico (NOT)
&&	"Y" lógico
AND	"Y" lógico
	"O" lógico
OR	"O" lógico
XOR	"OR" exclusivo

- Concatenación: El punto (.) es un operador que permite unir textos. Su uso es muy sencillo. Ejemplo:

```
<?php
    $x="Hola ";
    $y="a todo el mundo";
    $z=$x.$y; //$z vale "Hola a todo el mundo"
?>
```

En el manejo de información y de variables, se presentan algunas condicionantes, en las cuales se utilizan las estructuras de control de PHP.

- **Estructuras de control:** Hasta ahora las instrucciones que hemos visto, son instrucciones que se ejecutan secuencialmente; es decir, podemos saber lo que hace el programa leyendo las líneas de izquierda a derecha y de arriba abajo.

Las instrucciones de control de flujo permiten alterar esta forma de ejecución. A partir de ahora habrá líneas en el código que se ejecutarán o no dependiendo de una condición.

- Sentencia condicional: Se trata de una sentencia que, tras evaluar una expresión lógica, ejecuta una serie de instrucciones en caso de que la expresión lógica sea verdadera. Si la expresión tiene un resultado falso, ejecutaría otras instrucciones. Su sintaxis es:

```

if(expresión lógica){
    instrucciones a ejecutar si la expresión es verdadera
    ....
}
else {
    instrucciones a ejecutar si la expresión es falsa
    ...
}

```

Ejemplo de sentencia if-else:

```

<?php
    if($nota>=6){
        echo "Aprobado";
    }
    else {
        echo "Reprobado";
    }
?>

```

- Sentencia switch: Esta instrucción se usa cuando tenemos instrucciones que se ejecutan de forma diferente según evaluemos el conjunto de valores posible de una expresión. Cada **case** contiene un valor de la expresión; si efectivamente la expresión equivale a ese valor, se ejecutan las instrucciones de ese case y de los siguientes. (Sanchez, 2012)

```

switch (expresión) {
    case valor1:
        instrucciones del valor 1
        [break]
    [case valor2:
        instrucciones del valor 1
        [break]
    [
        .
        .
        .]
    [default:
        instrucciones que se ejecutan si la expresión no toma
        ninguno de los valores anteriores
        ..]
}

```

Ejemplo:

```
<?php
srand(time());
$diaSemana=rand(1,7);
switch ($diaSemana) {
case 1:
    $dia= "Lunes";
    break;
case 2:
    $dia= "Martes";
    break;
case 3:
    $dia= "Miércoles";
    break;
case 4:
    $dia= "Jueves";
    break;
case 5:
    $dia= "Viernes";
    break;
case 6:
    $dia= "Sábado";
    break;
case 7:
    $dia= "Domingo";
    break;
default:
    $dia= "?";
}
?>
```

- Bucle while: Un bucle es un conjunto de sentencias que se repiten mientras se cumpla una determinada condición. Los bucles agrupan instrucciones las cuales se ejecutan continuamente hasta que una determinada condición que se evalúa sea falsa.

```
while (expresión lógica) {
    sentencias que se ejecutan si la condición es true
}
```

Ejemplo (escribir números del 1 al 100):

```
<?php
```

```

$i=1;
while ($i<=100){
    echo $i . "<br />";
    $i++;
}
?>

```

- For: Es un bucle más complejo especialmente pensado para rellenar **arrays** o para ejecutar instrucciones controladas por un contador. Una vez más se ejecutan una serie de instrucciones en el caso de que se cumpla una determinada condición. (Sanchez, 2012) Sintaxis:

```

for(inicialización;condición;incremento){
    sentencias
}

```

Ejemplo (contar números del 1 al 1000):

```

<?php
for($i=1;$i<=1000;$i++){
    echo $i . "<br />";
}
?>

```

Para que una página web sea dinámica y que almacene o muestre información, se utilizan algunos métodos de envío de información.

- **Métodos de envío de datos del formulario:** Como se observó en la lección anterior de formularios HTML, un formulario puede enviar la información usando el atributo **method** del formulario y eligiendo **GET** o **POST**. (Sanchez, 2012)



- Paso de datos mediante GET: El método GET lo que hace es añadir a la URL destinataria del formulario los datos recogidos del formulario. Ejemplo:



```

<form action="recogida1.php" method="get">
  <label for="nombre">Escriba su nombre</label>
  <input type="text" name="nombre" /> <br />
  <label for="apellidos">Escriba sus apellidos</label>
  <input type="text" name="apellidos" /> <br />
  <label for="nombre">Escriba su dirección</label>
  <input type="text" name="direccion" /> <br />
  <label for="nombre">Escriba sus teléfonos:</label><br />
  <input type="text" name="telefono1" maxlength="9" pattern="[0-0]{9}" /> <br />
  <input type="text" name="telefono2" maxlength="9" pattern="[0-0]{9}" /> <br />
  <input type="submit" value="enviar" /><br />
</form>

```

El resultado del código, rellenando los campos, sería:

Escriba su nombre

Escriba sus apellidos

Escriba su dirección

Escriba sus teléfonos:

Suponiendo que estaS probando el código en la página web [formulario.html](#), al pulsar enviar, se genera la URL:

```

../recogida1.php?nombre=Jorge&apellidos=S%C3%A1nchez+Asenjo&direccion=C%2F+Los+Vientos&telefono1=666666666&telefono2=999999999

```

El apellido Sánchez se convierte en `S%C3%A1nchez` por que se requiere codificar en Unicode y en las URL es necesario utilizar el operador `%`, al que se le pasa el código hexadecimal del carácter. En este caso hay dos, porque en `utf-8` el símbolo á usa dos bytes.

Usa pares entre el nombre de la variable (en el formulario asignada a través del atributo `name`) y el valor que se le dio en el formulario. (Sanchez, 2012)

En el ejemplo anterior, en la página receptora, `../recogida1.php` tiene este código:

```

<!DOCTYPE HTML>
<html lang="es-ES">
<head>
  <meta charset="utf-8" />
  <title></title>
</head>
<body>
  <pre>
  <?php
    print_r($_GET);
  ?>
  </pre>
</body>
</html>

```

El resultado sería:

```

      Array
(
    [nombre] => Jorge
    [apellidos] => Sánchez Asenjo
    [direccion] => C/ Los Vientos
    [telefono1] => 6666666666
    [telefono2] => 9999999999
)

```

- Paso de datos mediante POST: Los formularios se crean igual en ambos casos, pero ahora cambiamos el atributo method de la etiqueta form para elegir POST.

Ahora la URL no añade los parámetros del formulario, quedan invisibles. Realmente se pasan en el paquete http que comunica a ambas páginas, pero no se ve en la URL con lo que habría que utilizar métodos más sofisticados para poder recoger los datos que el usuario introdujo en el formulario. (Sanchez, 2012)

- **Recepción de datos de un formulario desde una página PHP:** PHP permite recibir los datos pasados por los parámetros usando dos variables, que son en realidad dos **arrays** que contendrán todos los valores del formulario. Se trata de **\$\_GET** y de **\$\_POST** cada una dedicada al método con su mismo nombre.

Puesto que son **arrays** (se estudian con profundidad más adelante) su manejo es distinto al de una variable normal. Podemos entender que son contenedores de todos los elementos del formulario.

Los controles del formulario que hayan recibido valores envían esos valores asociados al nombre de control especificado en el atributo **name**. De modo que, si el método de paso era **POST**, la página PHP accede al valor de ese control mediante:

```
<?php
```

```
$valor=$_POST['name'];  
?>
```

Donde **name** es el nombre del control. (Sanchez, 2012)

## Conexión a MySQL desde PHP

Antes de comenzar e interactuar con una base de datos, tenemos que establecer una conexión entre PHP y el servidor de base de datos MySQL. Para hacerlo, PHP nos ofrece la función **mysql\_connect**. (Minera, Desarrollo PHP + MySQL, 2011)

Su sintaxis es:

```
<?php  
mysql_connect('servidor','nombre de usuario','password');  
?>
```

- **Servidor:** Cadena de caracteres que debe contener el nombre del servidor o bien su dirección IP. Cuando hablamos de servidor nos referimos a la maquina en donde se encuentra instalado el servidor de bases de datos MySQL. Si estarnos trabajando en forma local podríamos utilizar como nombre de servidor localhost o 127.0.0.1 como dirección IP.
- **Nombre de usuario:** Se trata de nombre de usuario válido para poder tener acceso a la base de datos en cuestión.
- **Password:** Corresponde a la contraseña que corresponde al nombre de usuario que se ha ingresado anteriormente.

Ejemplo:

```
<?php  
mysql_connect('localhost','prueba','12345');  
?>
```

En el ejemplo conectamos con el servidor local de MySQL utilizando el usuario prueba con contraseña 12345.

Si la conexión es correcta, la función devuelve los valores de la conexión (el enlace a datos) que se suele asignar a una variable (en el ejemplo anterior se los queda **\$con**).

En el caso de que falle (por que el usuario no tiene esa contraseña, o el servidor no está funcionando, etc. etc.), entonces la función devuelve falso. (Sanchez, PHP (III). Sesiones y bases de datos, 2013)

- **Captura de errores:** Al utilizar bases de datos hay muchos posibles errores que pueden ocurrir. De hecho, si no funciona la conexión, todo lo demás que nuestra aplicación quisiera realizar, no sería posible.
  - `mysql_error`: Muchas veces nos conviene capturar los errores que proceden de la base de datos MySQL para desde el programa decidir lo que hacer dependiendo del error y con esta instrucción, se devolverá el último error detectado por MySQL.

Ejemplo:

```
<?php
$con=mysql_connect("localhost","prueba","12345");
if($con){
    ...
    mysql_close($con);
}
else{
    echo "Error: conexión no realizada, respuesta del servidor: ";

    mysql_error($con)."Nº de error".mysql_errno($con);
}
?>
```

- **Seleccionar base de datos:** En MySQL las tablas pertenecen a una base de datos, por lo que lo habitual es seleccionar la base de datos para de ese modo trabajar de forma cómoda con las tablas y vistas de dicha base de datos. PHP permite seleccionar la base de datos mediante la función `mysql_select_db` a la que se le pasa el nombre de la base de datos (entre comillas, es decir como **string**) y, opcionalmente, el nombre del enlace de base de datos (si no se indica este parámetro se usa el último enlace a base de datos realizado con `mysql_connect`). Existen varias formas de realizarlo:
  - Seleccionando la base de datos:

```
<?php
$db = 'curso_php';
mysql_select_db($db) or die("No pude seleccionar la base de datos");
?>
```

En esta instrucción nos da la oportunidad de detectar el error de conexión con la base de datos

- Definiendo en la instrucción SQL:

```
<?php
$res=mysql_db_query("base_ejemplo", "SELECT * FROM TABLA");
?>
```

En esta instrucción se determina que base de datos es la que se está ejecutando y la instrucción SQL, los resultados se almacenan en una variable para posteriormente se haga uso de dichos resultados (o false, si falla la instrucción SQL).

- **Método para mostrar la información de las tablas:** Los conjuntos de resultados pueden ser pasados como parámetro a otras funciones. La más interesante para poder analizar los resultados es `mysql_fetch_array`. Esta función permite recorrer fila a fila los resultados de un conjunto de resultados.
  - Ejecutar una consulta simple:

```
<?php
$res=mysql_db_query("base_ejemplo", "SELECT * FROM TABLA");
?>
```

- Captura en el arreglo los resultados:

```
<?php
while($registros[] = mysql_fetch_array($res));
?>
```

El array devuelto (`$registros`) será del tipo multidimensional, con un formato como el siguiente:

```
<?php
$registros = array(
    array('campo1'=>'valor', 'campo2'=>'valor', 'campo7'=>'valor'),
    array('campo1'=>'valor', 'campo2'=>'valor', 'campo7'=>'valor'),
);
?>
```

Es decir, que será un array, conteniendo otro array por cada registro encontrado.

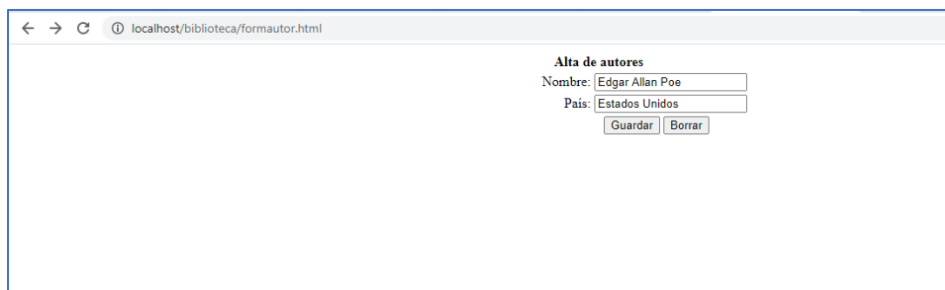
- **Ejecución de instrucciones SQL:** Por medio de la ejecución de instrucciones nativas de MySQL como `INSERT INTO`, `DELETE` o `UPDATE`, la respuesta tendrá como resultado un `result set`. Ejemplos:
  - Insertar nuevos valores:

```
<?php
$sql = "
    INSERT INTO  usuarios
              (username, email)
    VALUES    ('juanperez',
              'jperez@algundominio.ext')
";
$result = mysql_query($sql) or die("No pude ejecutar la consulta");
?>
```

Una vez que ya conoces la sintaxis ejemplificada para una inserción de datos, vamos a verlo gráficamente en una página con un formulario igual a los que ya has revisado en temas anteriores.

Primero la página del formulario y su forma gráfica.

```
C:\laragon\www\biblioteca\formautor.html - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
formautor.html altaautor.php
1 <html>
2 <head>
3 <meta charset="utf-8">
4 <title>Biblioteca Escolar</title>
5 </head>
6
7 <body>
8
9 <form action="altaautor.php" method="get">
10 <table width="600" border="0" align="center">
11 <tbody>
12 <tr>
13 <td colspan="6" align="center"><strong>Alta de autores</strong></td>
14 </tr>
15 <tr>
16 <td colspan="3" align="right">Nombre:</td>
17 <td colspan="3"><input type="text" name="txtname"></td>
18 </tr>
19 <tr>
20 <td colspan="3" align="right">Pais:</td>
21 <td colspan="3"><input type="text" name="txtpais"></td>
22 </tr>
23 <tr>
24 <td width="34">&nbsp;</td>
25 <td width="67">&nbsp;</td>
26 <td width="179">&nbsp;</td>
27 <td width="144" align="center"><input type="submit" name="submit" id="submit" value="Guardar">
28 <input type="reset" name="reset" id="reset" value="Borrar"></td>
29 <td width="47">&nbsp;</td>
30 <td width="103">&nbsp;</td>
31 </tr>
32 </tbody>
33 </table>
34 </form>
35 </body>
36 </html>
```



Como puedes observar en las ilustraciones, se está diseñando a través de HTML un formulario para registrar un autor en la base de datos y posteriormente se te muestra de forma gráfica como se ve en el navegador. Ahora vamos a ver la página que recibe la información y cómo se hace la conjugación de lenguaje SQL con PHP, de igual manera como ya se te mostró en el ejemplo anterior.

```

1 <?php
2 $host="localhost";
3 $user="root";
4 $dbnom="biblioteca";
5 $pass="";
6 $conexion=mysqli_connect($host,$user,$pass);
7 mysqli_select_db($conexion, $dbnom);
8 ?>
9 <?php
10 $consulta="Insert into autores(nombre,pais)values('$$_GET[txtname]','$$_GET[txtpais]')";
11 mysqli_query($conexion, $consulta);
12 echo "<h1><center>¡Registro de autor exitoso!</center></h1>";
13 ?>

```

localhost/biblioteca/altaaautor.php?txtname=Stephen+King&txtpais=Estados+Unidos&submit=Guardar

### ¡Registro de autor exitoso!

En la imagen anterior, como te darás cuenta en la primera parte se está haciendo una conexión a la base de datos “biblioteca” y a través de código PHP se está insertando un nuevo autor con los datos contenidos en las cajas de texto llamadas “txtname” y “txtpais”. Veamos el resultado en nuestra tabla.

Laragon\Biblioteca\autores\ - HeidiSQL Portable 10.2.0.5599

Archivo Editar Buscar Herramientas Ira Ayuda

Host: localhost Base de datos: biblioteca Tabla: autores Datos Consulta\*

Id_Autor	nombre	pais
1	JD Salinger	Estados Unidos
2	F. Scott. Fitzgerald	Estados Unidos
3	Jane Austen	Reino Unido
4	Scott Hanselman	Estados Unidos
5	Jason N. Gaylord	Estados Unidos
6	Pranav Rastogi	India
7	Todd miranda	Estados Unidos
8	Christian Wenz	Estados Unidos
9	Octavio Paz	México
10	Ernest Hemingway	Estados Unidos
11	William Faulkner	Estados Unidos
12	Juan Rulfo	México
13	Edgar Allan Poe	Estados Unidos

- Buscar información

Después de haber conocido la forma de insertar registros en cualquier tabla de una base de datos, es muy necesario saber la forma en la que le vas a proporcionar información a tu cliente o usuario.

```

    <?php
    $consulta="select * from autores where nombre like '%$_GET[txtsearch]%'";
    $resultado=mysqli_query($conexion, $consulta)or die("No se puede consultar");
    if (!$resultado)
    {
        echo " ";
        die;
    }
    $contador=1;
    $filas=mysqli_num_rows($resultado);

    while ($contador<=$filas)
    {
        $arreglo=mysqli_fetch_array($resultado);
        echo $arreglo ['Id_Autor'];
        echo "<br>";
        $contador++;
    }
    ?>

```

Debes analizar cada fragmento de código para saber cuáles son las instrucciones que se están ejecutando:

```

$consulta="select * from autores where nombre like '%$_GET[txtsearch]%'";
$resultado=mysqli_query($conexion, $consulta)or die("No se puede consultar");

```

En la primera línea se está pidiendo que se busquen todos los campos de la tabla autores, que coinciden con el nombre que el usuario escriba en una caja de texto que vamos a incluir en la misma página y en la segunda, se da la instrucción de ejecutar la consulta enviando un mensaje de error si esta no se completara.

```

if (!$resultado)
{
    echo " ";
    die;
}

```

Con esta instrucción podrás indicarle que si no encuentra ningún resultado coincidente, que envíe un espacio en blanco, evitando así que aparezca el mensaje de error anteriormente explicado.

```

$contador=1;
$filas=mysqli_num_rows($resultado);

while ($contador<=$filas)
{
    $arreglo=mysqli_fetch_array($resultado);

```



```

echo $arreglo ['nombre'];
echo "<br>";
$contador++;
}

```

Por último es necesario hacer un conteo de los registros que se encontraron, esto se hará con la instrucción "mysqli\_num\_rows", guardando el resultado en una variable llamada \$filas. Con un ciclo vamos a recorrer registro por registro para obtener la información, esto se logra con la instrucción "mysqli\_fetch\_array", que nos permitirá obtener los registros y guardarlos en un arreglo. Por último debemos imprimir ese arreglo indicando el campo que queremos visualizar, en este caso es el nombre.

Con la instrucción anterior deberíamos obtener la siguiente ventana de resultado.

localhost/biblioteca/buscarautores.php?txtsearch=+&submit=Buscar

JD Salinger  
 F. Scott. Fitzgerald  
 Jane Austen  
 Scott Hanselman  
 Jason N. Gaylord  
 Pranav Rastogi  
 Todd miranda  
 Christian Wenz  
 Octavio Paz  
 Ernest Hemingway  
 William Faulkner  
 Juan Rulfo  
 Edgar Allan Poe  
 Stephen King

Si observas se están imprimiendo todos los nombres, ya que no hemos especificado en una caja de texto, el que requerimos. De igual manera únicamente nos parecen los nombres de una manera no muy bien presentada, por lo que se hace necesario, primero hacer lo mismo pero especificando los demás campos que deseamos ver y por último insertar este código dentro de las celdas de una tabla para que haya mayor orden. Éste es el resultado:

localhost/biblioteca/buscarautores.php?txtsearch=+&submit=Buscar

### Consulta de información

Id del autor	Nombre	Pais
1	JD Salinger	Estados Unidos
2	F. Scott. Fitzgerald	Estados Unidos
3	Jane Austen	Reino Unido
4	Scott Hanselman	Estados Unidos
5	Jason N. Gaylord	Estados Unidos
6	Pranav Rastogi	India
7	Todd miranda	Estados Unidos
8	Christian Wenz	Estados Unidos
9	Octavio Paz	México
10	Ernest Hemingway	Estados Unidos
11	William Faulkner	Estados Unidos
12	Juan Rulfo	México
13	Edgar Allan Poe	Estados Unidos
14	Stephen King	Estados Unidos

Buscar por nombre del autor:  
 Nombre:

### Consulta de información

Id del autor	Nombre	País
9	Octavio Paz	México

Buscar por nombre del autor:  
Nombre:

- Eliminar un registro

```
<?php  
$sql = "DELETE FROM usuarios WHERE username='juanperez';  
$result = mysql_query($sql) or die('No pudo eliminar el registro');  
?>
```

- Actualizar un dato:

```
<?php  
$sql = "UPDATE usuarios SET email ='jperez@algundominio.COM";  
$result = mysql_query($sql) or die('No pudo actualizar el registro');  
?>
```

### Referencias

- Bahit, E. (2012). *Programador PHP*. Safe creative. <http://46.101.4.154/Libros/El%20lenguaje%20PHP.pdf>
- Minera, F. (2011). *Desarrollo PHP + MySQL*.
- Minera, F. (2014). *PHP Desde Cero: Manuales USERS (Spanish Edition)*. Argentina: Creative Andina Corp.
- Sánchez, J. (2012). *Introducción a PHP*.
- Sánchez, J. (2013). *PHP (III). Sesiones y bases de datos*.
- The PHP Group (2021). PHP. <https://www.php.net/>
- Robledano, A. (2019). Qué es MySQL: características y ventajas <https://openwebinars.net/blog/que-es-mysql/>

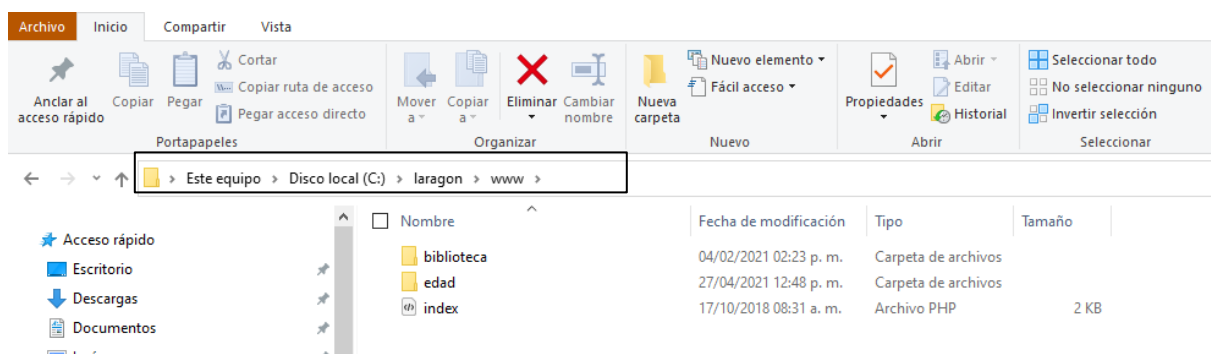


Comienza a crear el sitio web, para ello debes indicar cómo ejecutar el servicio web donde se almacenarán los archivos PHP y qué herramientas se utilizarás para diseñar estas páginas.

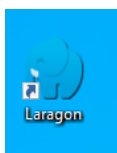
## Almacenamiento de páginas web

Cuando realizas la instalación de Laragon, se crean las carpetas que contienen los archivos del sitio web.

Para ingresar a la carpeta que contiene el sitio web, debes hacerlo del explorador de archivos, ir a las carpetas **C:\laragon\www**, en esta última observarás una página llamada **index.php**, esta página es la página inicial de tu servidor web local.

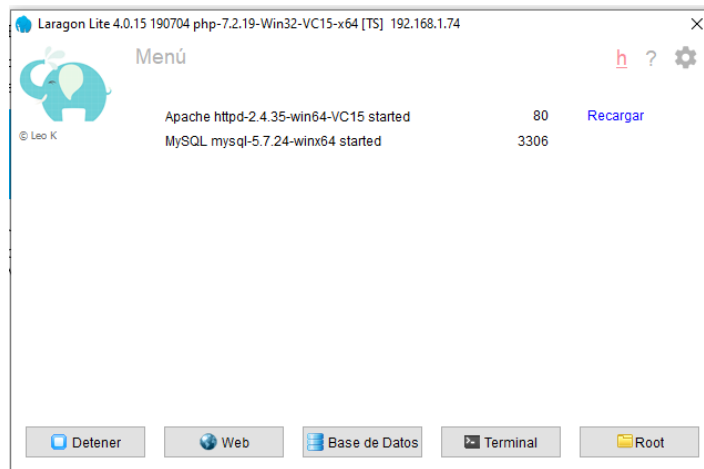



## Ejecución del servicio web



Para iniciar el servicio web, debes ejecutar el programa Laragon, su icono está en el escritorio, en la barra de inicio, en el menú de inicio o únicamente escribiendo "laragon", en el buscador de windows.

Una vez ejecutado el programa, aparece la pantalla inicial, para ello es necesario presionar el botón **Iniciar Todo**, posteriormente se ejecutarán los servicios web y MySQL.



Para verificar que el servicio web se está ejecutando, presiona el botón  y se abrirá el explorador web (Chrome, Firefox o el que tengas instalado) y aparecerá la pantalla inicial de Laragon.

# Laragon

Si no aparece este mensaje, ha ocurrido algún error al ejecutar los servicios web.

Apache/2.4.35 (Win64) OpenSSL/1.1.1b PHP/7.2.19  
 PHP version: 7.2.19 [info](#)  
 Document Root: C:/laragon/www

[Getting Started](#)

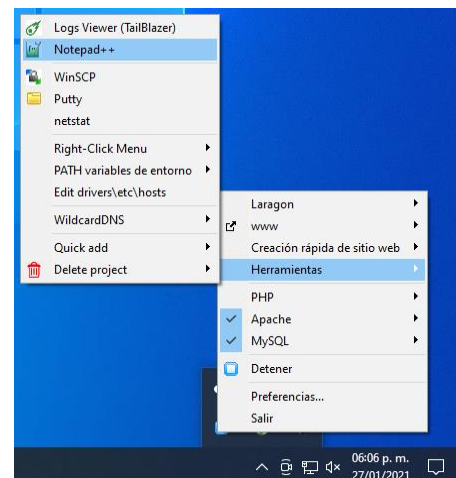


Cuando Laragon y los servicios web se están ejecutando, en los iconos del sistema de la barra de tarea de Windows aparece el de Laragon.

## Herramienta Notepad++

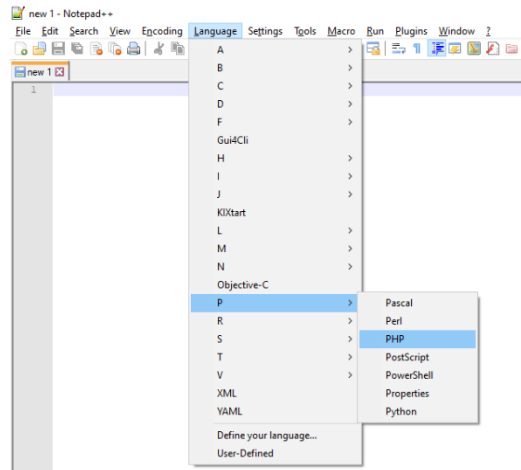
La herramienta Notepad++, es un editor de notas, pero con soporte para varios lenguajes de programación.

Para ejecutar esta herramienta da clic derecho en el icono de Laragon que está presente en la barra de tareas de Windows, en el menú contextual que aparece selecciona la opción **Herramientas** y posteriormente la herramienta **Notepad++**.



Una vez abierto el programa, es necesario establecer el lenguaje que utilizarás. En este caso realizarás HTML y páginas PHP, por lo que es necesario determinar este lenguaje. Para ello, ingresa al menú **Language**, ahí encontrarás un menú con letras en orden

alfabético que nos permite buscar a través de ellas el lenguaje con el que trabajarás; elige la opción **H** para seleccionar HTML o la opción **P** para seleccionar PHP.

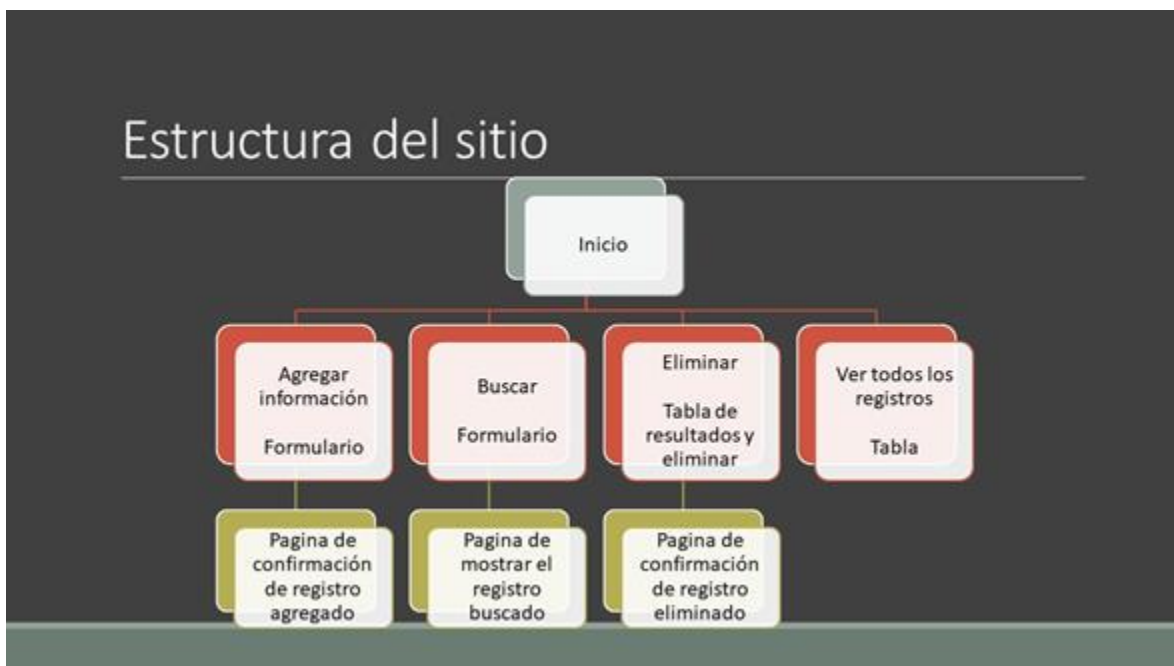


## Diseño web

Para llevar a cabo el diseño web, estaremos retomando un ejemplo previamente trabajado en ésta y otras lecciones, es la base de datos “biblioteca”, como recordarás, tiene varios campos que para desarrollar un sitio web, deberemos de tomar en cuenta.

Determinar la estructura del sitio web, toma en cuenta los requisitos que el cliente ha determinado para la biblioteca.

Vas a crear un sitio web que agregue libros, consulte libros, elimine libros y actualice libros.



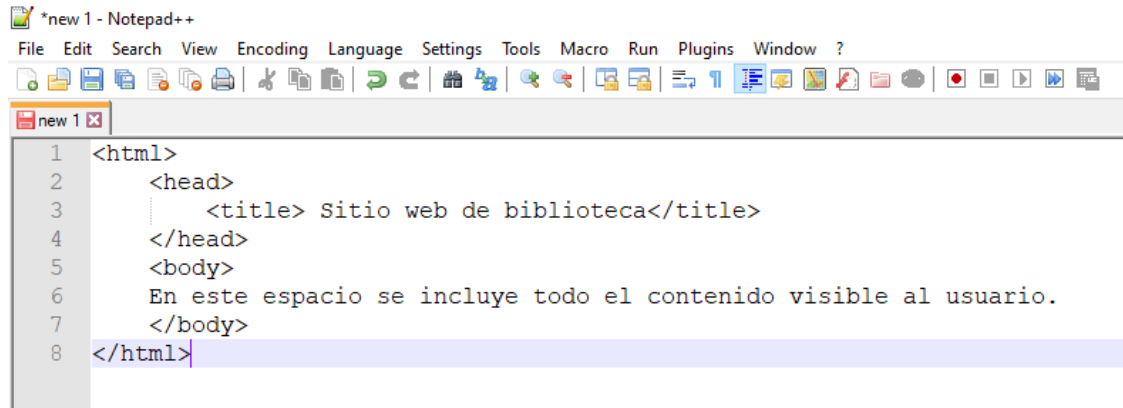
Cada uno de estos elementos, en realidad son páginas web, por lo que deben estar guardadas en el directorio **C:\laragon\www**. Para poder guardar cada página en el

directorio mencionado, deberás ingresar a Notepad++ y generar una estructura básica de página en HTML, como ya previamente se mencionó, posteriormente guardar este archivo como página web (.html o .php), en el directorio que ya se indicó.

## Página de inicio

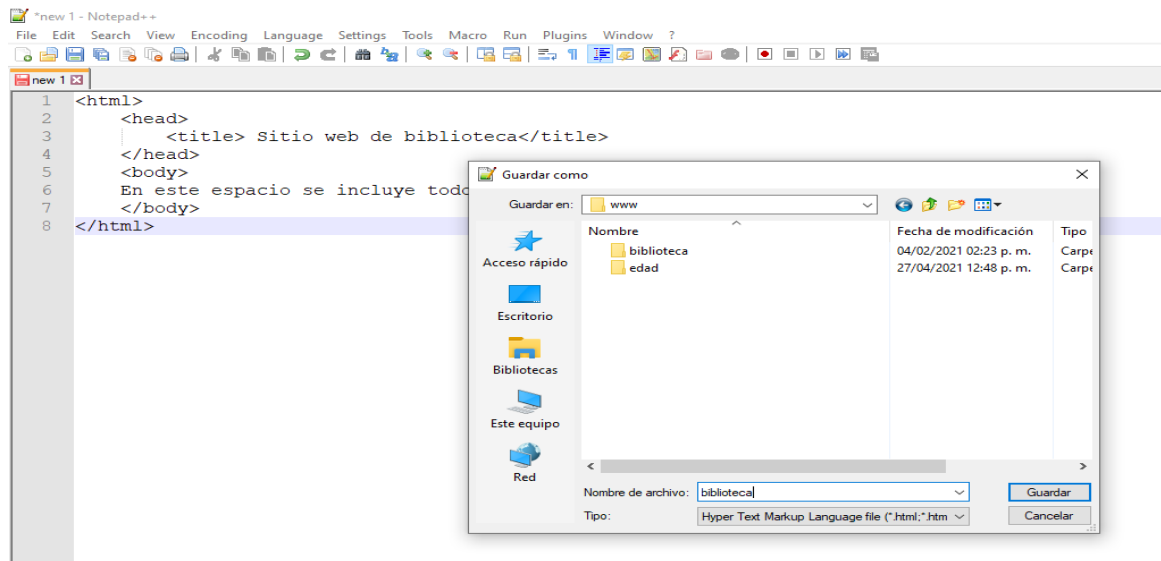
Esta página es la inicial, en ella se debe presentar el sitio y a su vez contener el menú de navegación entre las diferentes páginas, para construirla se sugiere lo siguiente:

1.- Genera un archivo de tipo .html, con la estructura básica de página web como se ejemplifica en la siguiente captura



```
*new 1 - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
new 1 x
1 <html>
2 <head>
3 <title> Sitio web de biblioteca</title>
4 </head>
5 <body>
6 En este espacio se incluye todo el contenido visible al usuario.
7 </body>
8 </html>
```

2.- Guardarlo en la carpeta c:/Iragon/www con el nombre biblioteca.html.



```
<?php
```

```
$consulta=_____ autores (_____) values ('$_GET[_____]');
```

```
mysqli____le____ ($conexion, $consulta);
```

```
echo "<h1><center>¡Registro de alumno exitoso!</center></h1>";
```

```
?
```

1.- \_query

2.- Insert into

3.- Título

4.- titulo

## Página de buscar

El cliente ha solicitado que realicemos búsqueda de algunos títulos de libros, por lo que solo se deberá escribir algunas partes del título del libro y con ello podrá hacer una búsqueda de este. Para ello debemos seguir los siguientes pasos:

- Paso 1: Construir el formulario de búsqueda  
Como primer paso es necesario crear un formulario en donde el usuario escriba el título de búsqueda del libro, para ello debe crear una página en HTML en la ubicación **C:\laragon\www** con el nombre de **buscar.html**. Para ello arma el siguiente código:

Orden	Código
1	<code>&lt;body&gt;</code>
2	<code>&lt;form action="buscar.php" method="post"&gt;</code> <code>&lt;p&gt;Titulo del libro: &lt;input type="text" name="titulo"&gt;&lt;/p&gt;</code> <code>&lt;input type="submit" value="Buscar"&gt;</code> <code>&lt;/form&gt;</code>
3	<code>&lt;/body&gt;</code>
4	<code>&lt;/html&gt;</code>
5	<code>&lt;h1&gt;Buscar libros&lt;/h1&gt;</code> <code>&lt;br /&gt;</code> <code>&lt;h3&gt;Por medio del siguiente formulario, escribe el título del libro y</code> <code>posteriormente presiona el botón Buscar.&lt;/h3&gt;</code> <code>&lt;p&gt;&amp;nbsp;&lt;/p&gt;</code> <code>&lt;p&gt;&amp;nbsp;&lt;/p&gt;</code>
6	<code>&lt;head&gt;</code> <code>&lt;meta charset=utf-8 /&gt;</code> <code>&lt;title&gt;Buscar libros&lt;/title&gt;</code> <code>&lt;/head&gt;</code>

Si observas un resultado como el siguiente cuando escribes la dirección de <http://localhost/buscar.html> tu resultado estará correcto:

### Buscar libros

Por medio del siguiente formulario, escribe el título del libro y posteriormente presiona el botón Buscar.

Titulo del libro:

- Paso 2: Diseñar la conexión de consulta de datos

Una vez que se tiene la página de formulario de buscar, si detectaste el formulario está usando el método POST y redirecciona el formulario a la página buscar.php, para ello se debe construir la página en donde realizara la conexión, la búsqueda y el resultado

de la búsqueda, para ello se debe crear una página con el nombre **buscar.php** en la ubicación de **C:\laragon\www**. Esta página está dividida en 3 secciones:

- a) Conexión MySQL
- b) Consulta del libro
- c) Mostrar resultado de la búsqueda

## Página de eliminar

Ahora el cliente nos ha solicitado que pudiéramos borrar algún libro determinado, con una advertencia previa a borrar el registro, para ello, se realiza un formulario como el diseñado en la página buscar.html, solamente que en esta ocasión se llamara **eliminar.html** y el formulario utilizara el método POST y direccionara a una página llamada **eliminar.php**, para esta tarea te ayudaremos a construir la página de **eliminar.html**.

```
<html>
  <head>
    <meta charset=utf-8 />
    <title>Eliminar libros</title>
  </head>
  <body>
    <h1>Eliminar libros</h1>
    <br />
    <h3>Por medio del siguiente formulario, escribe el título del libro y
    posteriormente presiona el botón Eliminar.</h3>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <form action="eliminar.php" method="post" onSubmit="return confirm('Pulce ACEPTAR
    para confirmar la eliminacion o CANCELAR la eliminacion');">
      <p>Titulo del libro: <input type="text" name="titulo"></p>
      <input type="submit" value="eliminar">
    </form>
  </body>
</html>
```

Nota: en esta ocasión el botón eliminar, pedirá una confirmación de que si desea eliminar el registro y el usuario tendrá que confirmar su eliminación.

## Página de ver registros

El cliente desea observar los registros de los libros que tiene guardados, para ello se mostrara un reporte de todos los libros registrados en la base de datos,





## Actividad de aprendizaje

Recorta y pega la estructura de la página.

```
<html>  
<body>  
  <h1>Menú principal</h1>  
  <br>  
  <ul>  
    <li><a href="biblioteca.html">Inicio</a></li>  
    <li><a href="agregar.html">Agregar libros</a></li>  
    <li><a href="buscar.html">Buscar libros</a></li>  
    <li><a href="eliminar.html">Eliminar libros</a></li>  
    <li><a href="ver.php">Ver todos los libros</a></li>  
  </ul>  
</body>  
</html>  
<head>  
  <title>Página de inicio del sitio Web "La Biblioteca".</title>  
</head>
```



Ordena el siguiente código de acuerdo y escribe el orden correcto en el recuadro vacío.

Si tienes las herramientas necesarias prueba el código en la herramienta Notepad++, guarda el archivo con el nombre **agregar.html**.

No.	Código desordenado
3	<code>&lt;body&gt;</code>
2	<code>&lt;head&gt;   &lt;meta charset=utf-8 /&gt;   &lt;title&gt;Agregar libros&lt;/title&gt; &lt;/head&gt;</code>
6	<code>&lt;/body&gt; &lt;/html&gt;</code>
4	<code>&lt;h1&gt;Agregar nuevos libros&lt;/h1&gt; &lt;br /&gt; &lt;h3&gt;Por medio del siguiente formulario, agrega nuevos libros, recuerda que solo debes capturar el título del libro y posteriormente presiona el botón Guardar.&lt;/h3&gt; &lt;p&gt;&amp;nbsp;&lt;/p&gt; &lt;p&gt;&amp;nbsp;&lt;/p&gt;</code>
1	<code>&lt;html&gt;</code>
5	<code>&lt;form action="confirmar_agregar.php" method="post"&gt;   &lt;p&gt;Titulo del libro: &lt;input type="text" name="titulo"&gt;&lt;/p&gt;   &lt;input type="submit" value="Guardar"&gt; &lt;/form&gt;</code>

1.- Nombre del host	biblioteca
2.- Nombre del usuario	localhost
3.- Contraseña	root
4.- Nombre de la base de datos	“vacío” o cualquiera que se establezca

**Genera las instrucciones para conectar a MySQL y para seleccionar la base de datos biblioteca.**

**Escribe recomendaciones que le darías a tus clientes para que lleve a cabo la captura de información de registros en los formularios que propones en tu página de captura.**

Construye tu resultado en el siguiente recuadro y recuerda escribirlo por medio de la herramienta Notepad++ para que se guarde en tu servidor web.

Construye la página de **buscar.php** con el siguiente código

```
<?>
if ($numero_resultados > 0) {
    echo '<h2>Se han encontrado ' . $numero_resultados . ' resultados.</h2>';
    echo '<ul>';
    while ($columnas_buscadas = mysqli_fetch_array($query_buscada)) {
        //En este caso solo mostramos el titulo y fecha de la entrada
        echo '<li>Número: ' . $columnas_buscadas['Id_Libro'] . ' - Título: ' . $columnas_buscadas['Titulo'] . '</li>';
    }
    echo '</ul>';
}
else {
    //Si no hay registros encontrados
    echo '<h2>No se encuentran resultados con los criterios de búsqueda.</h2>';
}

//cadena de conexion
$host="localhost";
$usuario="root";
$password="";
$base_de_datos="biblioteca";
$conexion=mysqli_connect($host,$usuario,$password);
mysqli_select_db($conexion,$base_de_datos) or die ("Ninguna DB seleccionada");

$title_buscar=$_POST["titulo"];
$query="SELECT * FROM libros WHERE titulo LIKE '%" . $title_buscar . "%'";
$query_buscada = mysqli_query( $conexion,$query);
$numeros_resultados = mysqli_num_rows($query_buscada);
//Si ha resultados

<?php
echo '<a href="biblioteca.html">Regresar al inicio';
```

Reorganiza el siguiente código para eliminar el registro.

Orden	Código
1	<code>\$titulo_buscar=\$_POST["titulo"]; \$query="DELETE FROM libros WHERE titulo='" . \$titulo_buscar . "'"; mysqli_query( \$conexion,\$query); \$resultado = mysqli_query(\$conexion,\$query);</code>
2	<code>?&gt;</code>
3	<code>&lt;?php</code>
4	<code>if (!\$resultado) {     echo "&lt;h2&gt;No se podido eliminar el registro&lt;/h2&gt;"; }else{     echo "&lt;h2&gt;El registro ha sido eliminado&lt;/h2&gt;"; }  echo '&lt;a href="biblioteca.html"&gt;Regresar al inicio';</code>
5	<code>//cadena de conexion \$host="localhost"; \$usuario="root"; \$password=""; \$base_de_datos="biblioteca"; \$conexion=mysqli_connect(\$host,\$usuario,\$password); mysqli_select_db(\$conexion,\$base_de_datos) or die ("Ninguna DB seleccionada");</code>

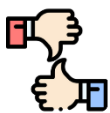
Escribe el código reorganizado en el siguiente espacio y pruébalo en la herramienta Notepad++:

Crema una página con el nombre **ver.php** por medio del siguiente código desordenado

```
<table border="1">
<tbody>
<tr>
<td>No.</td>
<td>Titulo</td>
</tr>
<tr>
<?php
while ($columnas_buscadas = mysqli_fetch_array($query_buscada)) {
//En este caso solo mostramos el titulo y fecha de la entrada
echo '<td>'. $columnas_buscadas['Id_Libro'].'</td><td>'. $columnas_buscadas['Titulo'].'</td>';
echo '</tr>';
}
?>
<?php
//cadena de conexion
$host="localhost";
$usuario="root";
$password="";
$base_de_datos="biblioteca";
$conexion=mysqli_connect($host,$usuario,$password);
mysqli_select_db($conexion,$base_de_datos) or die ("Ninguna DB seleccionada");
$query="SELECT * FROM libros";
$query_buscada = mysqli_query( $conexion,$query);
$numeros_resultados = mysqli_num_rows($query_buscada);
echo '<h2>Se han encontrado '.$numeros_resultados.' resultados.</h2>';
?>
</tbody>
</table>
<?php
echo '<a href="biblioteca.html">Regresar al inicio';
?>
```

Escribe en orden el código y pruébalo en la herramienta Notepad++

Realiza las pruebas necesarias a partir del menú principal del sistema biblioteca y comprueba que todas las paginas abren correctamente y no te marcan errores.



## Autoevaluación

Indicadores	Lo pude hacer	Tengo dudas	Necesito trabajar más
Utilizo MySQL y PHP para la organización de datos a partir de los requerimientos del cliente			
Soy capaz de crear bases de datos y sus tablas de datos			
Conozco las sintaxis de las sentencias SQL			
Puedo manipular la información de una base de datos utilizando las sentencias SELECT, INSERT INTO y DELETE.			
Soy capaz de desarrollar un sistema web por medio de PHP y MySQL			
Conozco la estructura básica de HTML, para generar un archivo de página web.			
Sé realizar la conexión entre una página web y un gestor de bases de datos a través de código PHP			
Conozco el funcionamiento básico del lenguaje PHP			

# Análisis y diseño de sistemas



Durante mucho tiempo, las organizaciones han reconocido la importancia de administrar recursos clave como las personas y la materia prima. Actualmente, la información ha encontrado su lugar apropiado como recurso clave. Los responsables de la toma de decisiones por fin comprenden que la información no es sólo un producto derivado de las operaciones comerciales, sino que además provee impulso a las empresas y puede constituir el factor decisivo para determinar el éxito o el fracaso de un negocio pequeño, mediano o grande.



Para maximizar la utilidad de la información, una empresa debe administrarla en forma apropiada, de la misma manera en que administra los demás recursos. Los analistas necesitan comprender que hay costos asociados con la producción, distribución, seguridad, el almacenamiento y la recuperación de toda información. Aunque la información está a nuestro alrededor no es gratuita y no debemos dar por hecho su uso estratégico para poner a una empresa en una posición competitiva.

El hecho de que las computadoras se puedan conectar de manera rápida y sencilla a una red, aunado a la capacidad de acceder a Internet, ha creado una explosión de información en la sociedad en general, y en los negocios en particular. El proceso de administrar la información generada por computadora se aleja de manera considerable del proceso de manejar los datos producidos en forma manual; por lo general debemos administrar una mayor cantidad de información haciendo uso de una computadora y de bases de datos, recordando para ello, los tipos básicos de sistemas de computación **que un analista debe conocer.**





***¡Vamos a aprender!***

## **Conceptos preliminares del análisis y diseño de sistemas**

El análisis y diseño de sistemas de información consiste en investigar sistemas y necesidades actuales para proponer sistemas mejores. Por brevedad, suele decirse *análisis de sistemas*.

**Sistemas de información.** Son sistemas que procesan datos para producir información. Los datos son las entradas y la información la salida. Pueden ser manuales, cuando no hay intervención de computadoras, o computarizados.

**Componentes de los sistemas de información.** Son personas, datos, soportes de datos, máquinas, procedimientos, programas, controles, formularios, reglamentaciones, etc. Toda organización tiene sistemas de información que se inserta y conecta en las estructuras administrativas del mismo modo que los nervios en el cuerpo.

**Análisis.** Consiste en recolectar e interpretar hechos sobre el sistema actual y las necesidades de información actuales y previsibles en el futuro, para detectar:

- Lo que está bien
- Lo que está mal
- Lo que sobra
- Lo que falta

**Diseño.** Consiste en planear y desarrollar un nuevo sistema que soluciones los problemas detectados en el sistema actual y los supere ventajosamente. El nuevo sistema puede limitarse a remendar el sistema actual, pero también puede ser un cambio de grandes dimensiones.

**Investigación.** Es el proceso utilizado por la ciencia y la tecnología para adquirir conocimiento, formular hipótesis, probarlas o rechazarlas. Es un proceso porque consta de una serie de etapas.

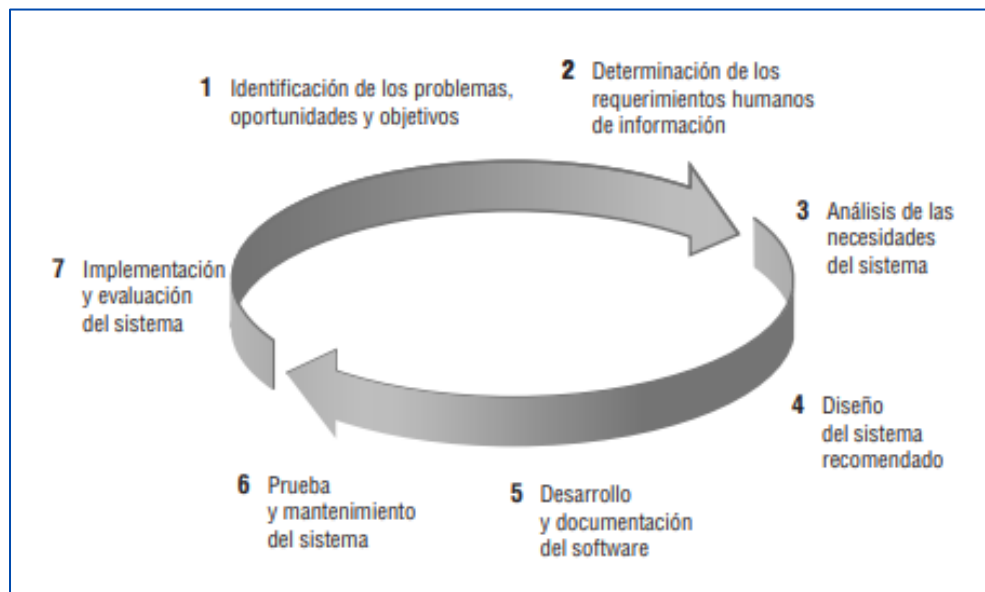
Una vez revisados los conceptos preliminares del análisis y diseño de sistemas, procedamos a revisar porque es importante y tan útil al momento de desarrollar un sistema de información.

## **Necesidad del análisis y diseño de sistemas**

El análisis y diseño de sistemas que los analistas de sistemas deben llevar a cabo busca comprender qué necesitan los humanos para analizar la entrada de datos de manera sistemática, procesar o transformar los datos, almacenarlos y producir información en el contexto de una organización específica (reportes o consultas). Mediante un análisis detallado, los analistas buscan identificar y resolver los problemas correctos. Además, el análisis y diseño de sistemas se utiliza para analizar, diseñar e implementar las mejoras

en el apoyo para los usuarios y las funciones de negocios que se puedan llevar a cabo mediante el uso de sistemas de información.

Si un sistema se pone en marcha sin una planificación apropiada, a menudo los usuarios quedan muy insatisfechos y dejan de usar el sistema. El análisis y diseño añade estructura a los sistemas, y constituye una actividad costosa que de otra manera se realizaría al azar. Se puede considerar como una serie de procesos que se llevan a cabo en forma sistemática para mejorar una empresa mediante el uso de sistemas de información computarizados. El análisis y diseño de sistemas implica trabajar con los usuarios actuales y eventuales de los sistemas de información para ofrecerles soporte en su empleo de las tecnologías en un entorno organizacional.



La participación del usuario en el proyecto de sistemas es imprescindible para el desarrollo exitoso de los sistemas de información computarizados. Los analistas de sistemas son el otro componente esencial para desarrollar sistemas de información útiles.

Los usuarios avanzan al primer plano a medida que los equipos de desarrollo de software se internacionalizan más en cuanto a su composición. Esto significa que hay más énfasis en trabajar con los usuarios de software; en realizar un análisis de su empresa, sus problemas y objetivos, y en comunicar el análisis y diseño del sistema planificado a todos los involucrados.

### **Determinación de los requerimientos de información del factor humano**

La siguiente fase a la que entra el analista es determinar las necesidades de los usuarios involucrados, mediante el uso de varias herramientas, para comprender la forma en que interactúan en el contexto laboral con sus sistemas de información actuales. El analista utilizará métodos interactivos como entrevistas, muestreos e investigación de datos duros,

además de los cuestionarios y los métodos discretos, como observar el comportamiento de los encargados al tomar las decisiones y sus entornos de oficina, y los métodos integrales como la creación de prototipos.

El analista utilizará estos métodos para plantear y responder muchas preguntas relacionadas con la interacción humano-computadora (HCI), incluyendo preguntas tales como: “¿Cuáles son las fortalezas y limitaciones físicas de los usuarios?”, o dicho en otras palabras, “¿qué hay que hacer para que el sistema sea perceptible, legible y seguro?”, “¿cómo puede diseñarse el nuevo sistema para que sea fácil de usar, aprender y recordar?”, “¿cómo puede el sistema ser agradable o incluso divertido de usar?”, “¿cómo puede el sistema apoyar las tareas laborales individuales de un usuario y buscar nuevas formas de hacerlas más productivas?”.

En la fase de requerimientos de un sistema de información, el analista se esfuerza por comprender qué información requieren los usuarios para realizar sus trabajos. En este punto el analista examina cómo hacer que el sistema sea útil para las personas involucradas. Ante ello, el analista de sistemas debe plantearse las siguientes preguntas:

- ¿Cómo puede el sistema ofrecer un mejor apoyo para las tareas individuales que se deben llevar a cabo?
- ¿Qué nuevas tareas habilita el nuevo sistema que los usuarios no podían realizar sin él?
- ¿Cómo se puede crear el sistema de manera que extienda las capacidades de un usuario más allá de lo provisto por el sistema anterior?
- ¿Cómo puede el analista crear un sistema amigable para los trabajadores?



Fig. 2 Análisis más diseño, es igual a calidad

Las personas involucradas en esta fase son los analistas y los usuarios, por lo general los gerentes y los trabajadores de operaciones. El analista de sistemas debe conocer los detalles sobre las funciones del sistema actual:

- el quién (las personas involucradas)
- el qué (la actividad de la empresa)
- el dónde (el entorno en el que se lleva a cabo el trabajo)

- el cuándo (la coordinación)
- y el cómo (de qué manera particular se realizan los procedimientos actuales) de la empresa a la que está estudiando.



Después, el analista debe preguntar por qué la empresa utiliza el sistema actual. Puede haber buenas razones por las cuales la empresa trabaje con los métodos actuales, razón por la que se deben tener en cuenta al diseñar un nuevo sistema. El desarrollo ágil es una metodología orientada a objetos (OOA) para el desarrollo de sistemas, en la cual se incluye un método de desarrollo (junto con la generación de los requerimientos de información) así como herramientas de software. No obstante, si la razón de seguir con las operaciones actuales es que “siempre se ha hecho de esa forma”, el analista querrá mejorar los procedimientos. Al terminar esta fase, el analista deberá comprender la forma en que los usuarios realizan su trabajo al interactuar con una computadora y deberá empezar a comprender cómo mejorar la utilidad y capacidad de uso del nuevo sistema. También deberá saber cómo funciona la empresa y tener información completa sobre personas, objetivos, datos y procedimientos involucrados.

### Cualidades del analista de sistemas

El analista de sistemas dentro del desarrollo de un software tiene una relevancia importante ya que se encarga de darle estructura a un sistema computacional en pequeña proporción o bien, un sistema empresarial. Algunas de las virtudes o características que describen al analista son las siguientes:

- Es un solucionador de problemas
- Es una persona que disfruta diseñar soluciones factibles
- Debe contar con capacidad para relacionarse con los demás
- Debe tener una ética personal y profesional firme
- Debe ser una persona auto disciplinada

Por todo lo antes revisado en esta lección, como podrás darte cuenta es sumamente importante entender el análisis y diseño de sistemas para poder diseñar e implementar informes de datos en un escenario real, ya que los reportes constituyen la esencia de un sistema de información, ya que si un sistema no arroja salidas, pues no tendría ninguna razón de ser, ¿no lo crees?



**Para saber más**

Para reforzar el conocimiento sobre el tema **Análisis y diseño de sistemas**, revisa el siguiente vídeo escaneando el código QR o dando clic en el enlace.



<https://youtu.be/wf3UazuQs6c>

### Referencias

- Kendall & Kendall (2011). Análisis y diseño de sistemas (8va. Ed.), pp. 28-49. Prentice Hall.
- Cáceres, E. (2014). Análisis y diseño de sistemas de información.

Imágenes tomadas de:

- <https://www.canva.com/>



### Actividad de aprendizaje

Lee las siguientes oraciones y subraya la respuesta correcta

1. Consiste en investigar sistemas y necesidades actuales para proponer sistemas mejores .  
a) Análisis y diseño de sistemas    b) Sistema    c) Diseño    d) Información
2. Nos sirve para detectar lo que está bien, lo que está mal, lo que sobra y lo que falta en un sistema.  
a) Base de datos    b) Software    c) Análisis    d) Diseño
3. Es una cualidad del analista de sistemas  
a) Soluciona problemas    b) Amable    c) Atento    d) Eficiente
4. ¿Cuántas fases componen el ciclo del desarrollo de sistemas?  
a) Seis    b) Cinco    c) Tres    d) Siete
5. ¿En un sistema que representa la actividad de la empresa?  
a) El quién    b) El qué    c) El dónde    d) El cuándo
6. ¿En un sistema que representa la coordinación?  
a) El quién    b) El qué    c) El dónde    d) El cuándo
7. ¿En un sistema que representan las personas involucradas?  
a) El quién    b) El qué    c) El dónde    d) El cuándo
8. ¿En un sistema que representa el entorno en el que se lleva a cabo el trabajo?  
a) El quién    b) El qué    c) El dónde    d) El cuándo

### Realiza un análisis y diseño del sistema

En la lección 10, trabajaste con la base de datos **Biblioteca** y con la tabla **Libro**, ¿lo recuerdas?. Ahora vas a analizar el contenido de la tabla y sus posibles mejoras.

La tabla **Libro** cuenta actualmente con dos campos: ID\_Libro y Título ¿son suficientes estos dos campos para describir las características de un libro? no ¿verdad?

Ahora es tu turno de sugerir campos para la tabla **Libro** que permitan obtener mejores resultados a la hora de realizar una consulta.

En la siguiente tabla se muestran unos ejemplos, debes llenar los espacios faltantes.

Nombre del campo	Tipo de dato	Justificación
ID_Libro	Autonumérico	
Título	Varchar(30)	
Edición	Varchar(10)	Este campo permitirá registrar el número de edición del libro.
NPaginas	Int	Este campo permitirá registrar el número de páginas del libro

La tabla **Autores** cuenta actualmente con tres campos: ID\_Autor, Nombre y País, ¿son suficientes estos dos campos para describir las características de un autor?, no ¿verdad?

Ahora es tu turno de sugerir campos para la tabla **Autores** que permitan obtener mejores resultados a la hora de realizar una consulta. En la siguiente tabla se muestran unos ejemplos, tu tarea es llenar los espacios faltantes.

Nombre del campo	Tipo de dato	Justificación
ID_Autor	Autonumérico	
Nombre	Varchar(30)	
País	Varchar(30)	
Sexo	Varchar(10)	Este campo permitirá registrar el sexo del autor (Femenino o Masculino).
Edad	Int	Este campo permitirá registrar la edad del autor



**Contesta las siguientes preguntas.**

1. ¿Por qué es importante solicitar los requerimientos al inicio del desarrollo de un sistema?

---

---

---

---

2. ¿En qué porcentaje consideras que influye hacer un buen análisis del sistema? arguméntalo.

---

---

---

---

3. ¿Qué características de diseño debe poseer la interfaz (vista) de un sistema para que sea agradable para el usuario?, enlístalas.

---

---

---

---

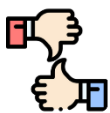
4. Con base en lo analizado hasta el momento, describe las siete fases del ciclo de desarrollo de sistemas (puedes continuar en la parte de atrás de la hoja).

---

---

---

---



## Autoevaluación

Indicadores	Lo pude hacer	Tengo dudas	Necesito trabajar más
Comprendo los conceptos básicos del análisis y diseño de sistemas.			
Soy capaz de proponer mejoras significativas a la tabla Libro para obtener mejores resultados de consulta.			
Conozco las siete fases del ciclo del desarrollo de sistemas con una argumentación informática.			