



# Curso de Programación en PHP

## Nivel I

Universidad Autónoma de Entre Ríos  
Facultad de Ciencia y Tecnología - Oro Verde - v2.1



### Capítulo 4: Programación de base de datos

## **Extensiones de base de datos en PHP**

### **Introducción a MySQL**

### **SQL como lenguaje de consultas**

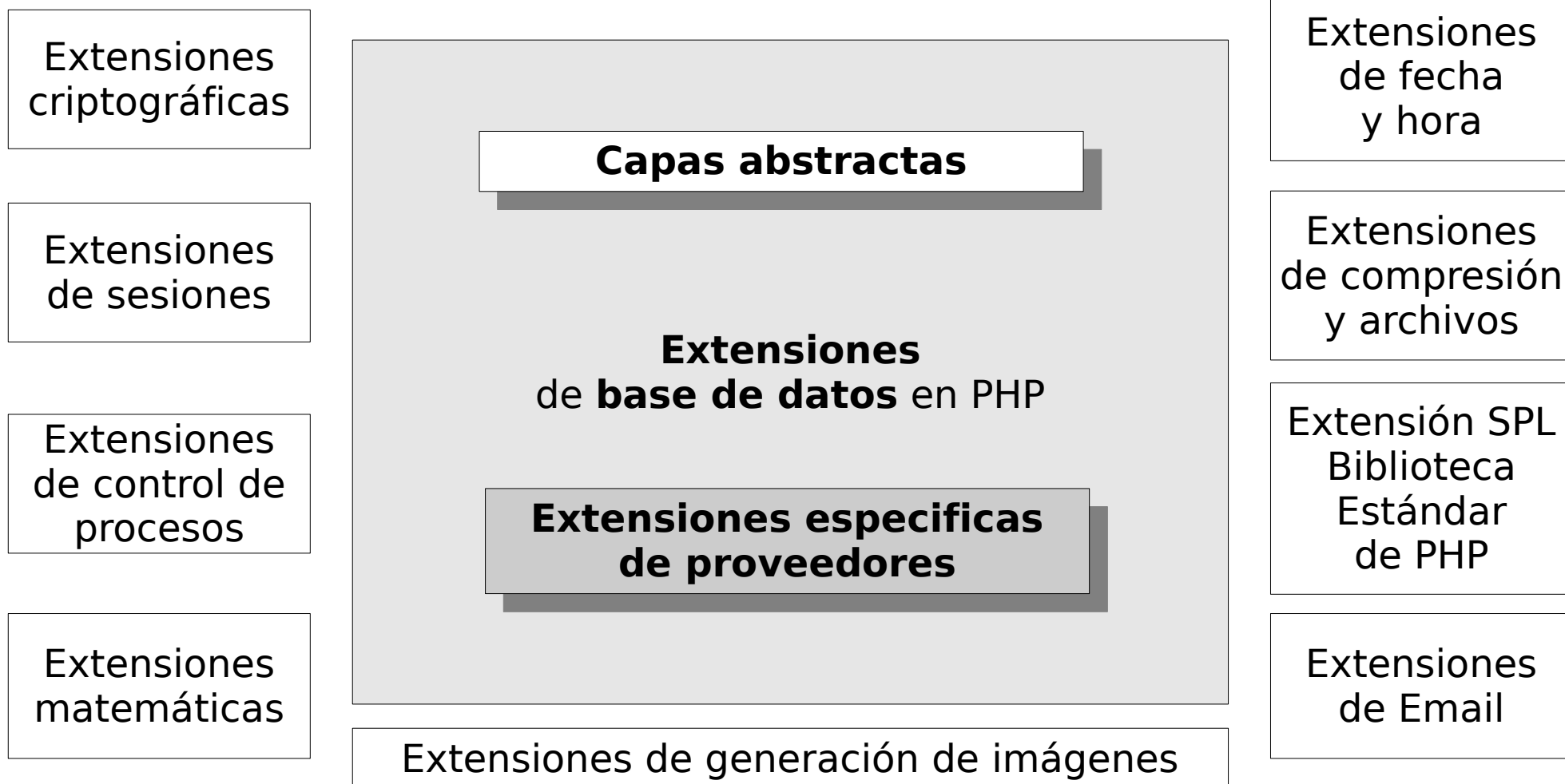
### **Introducción a PDO**

### **Usando métodos de la clase PDO**



### Clase 7: Introducción a SQL, MySQL y PDO

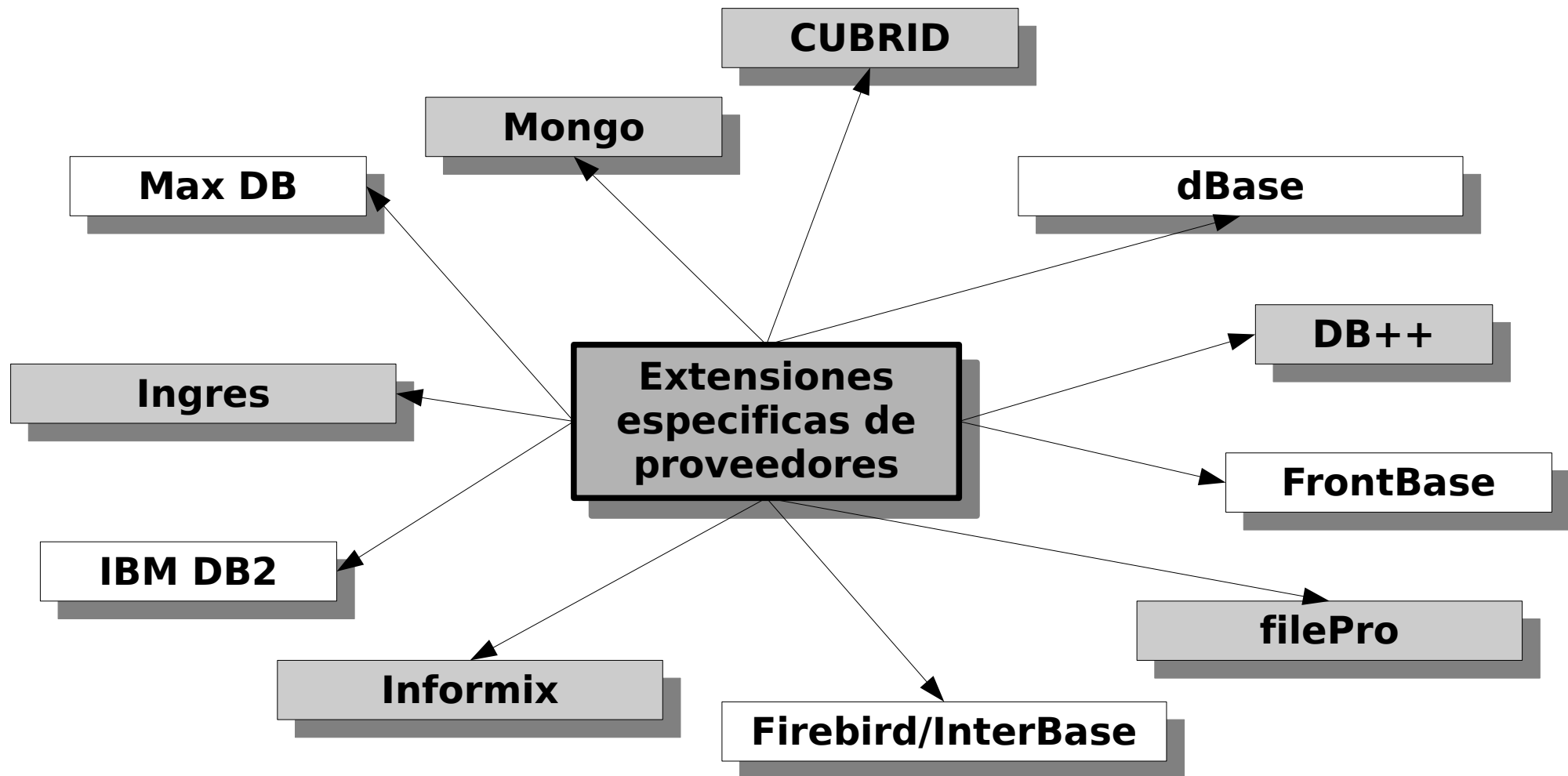
## Extensiones en el lenguaje PHP





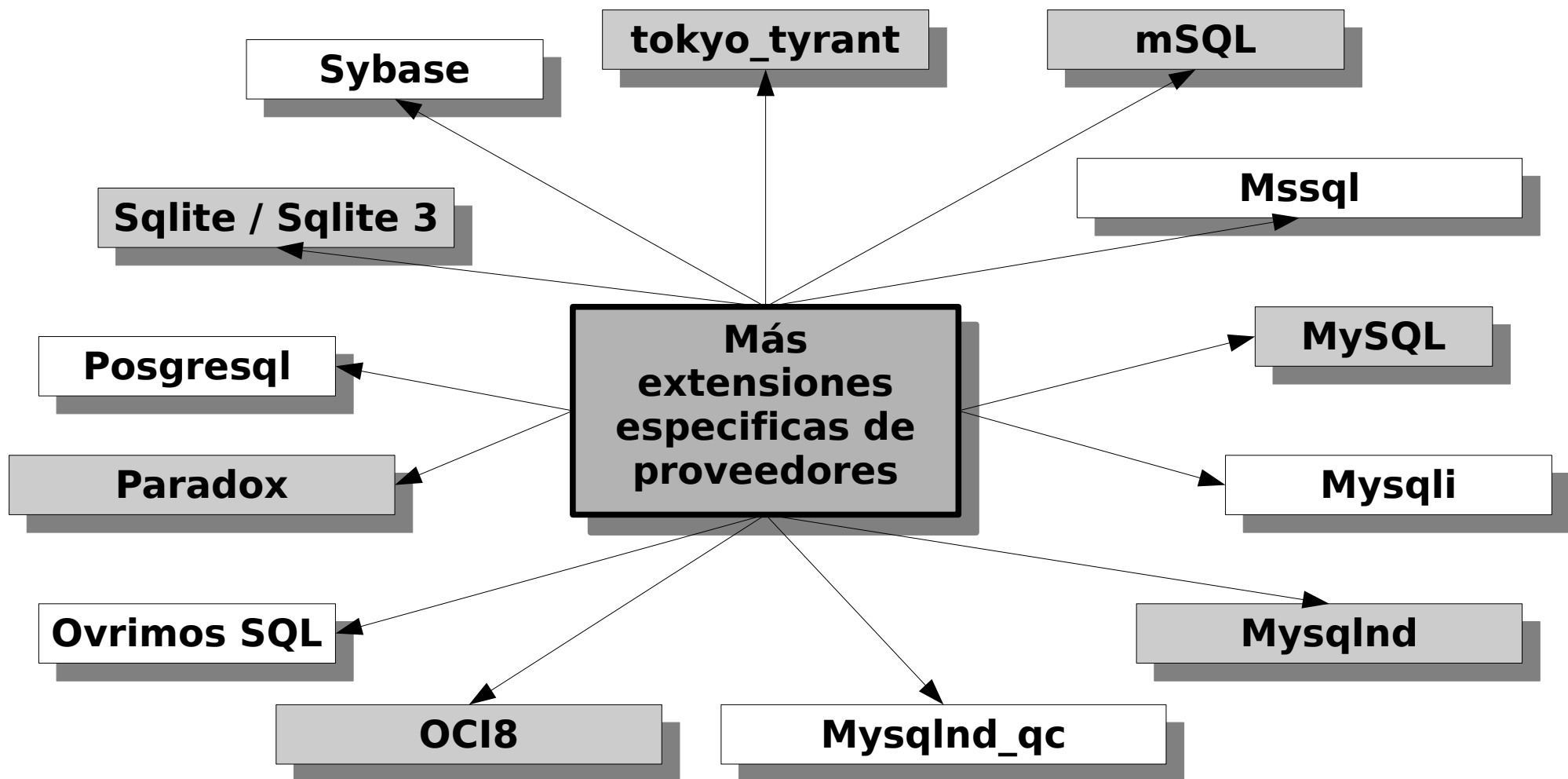
### Clase 7: Introducción a SQL, MySQL y PDO

## Extensiones específicas de proveedores de DB



### Clase 7: Introducción a SQL, MySQL y PDO

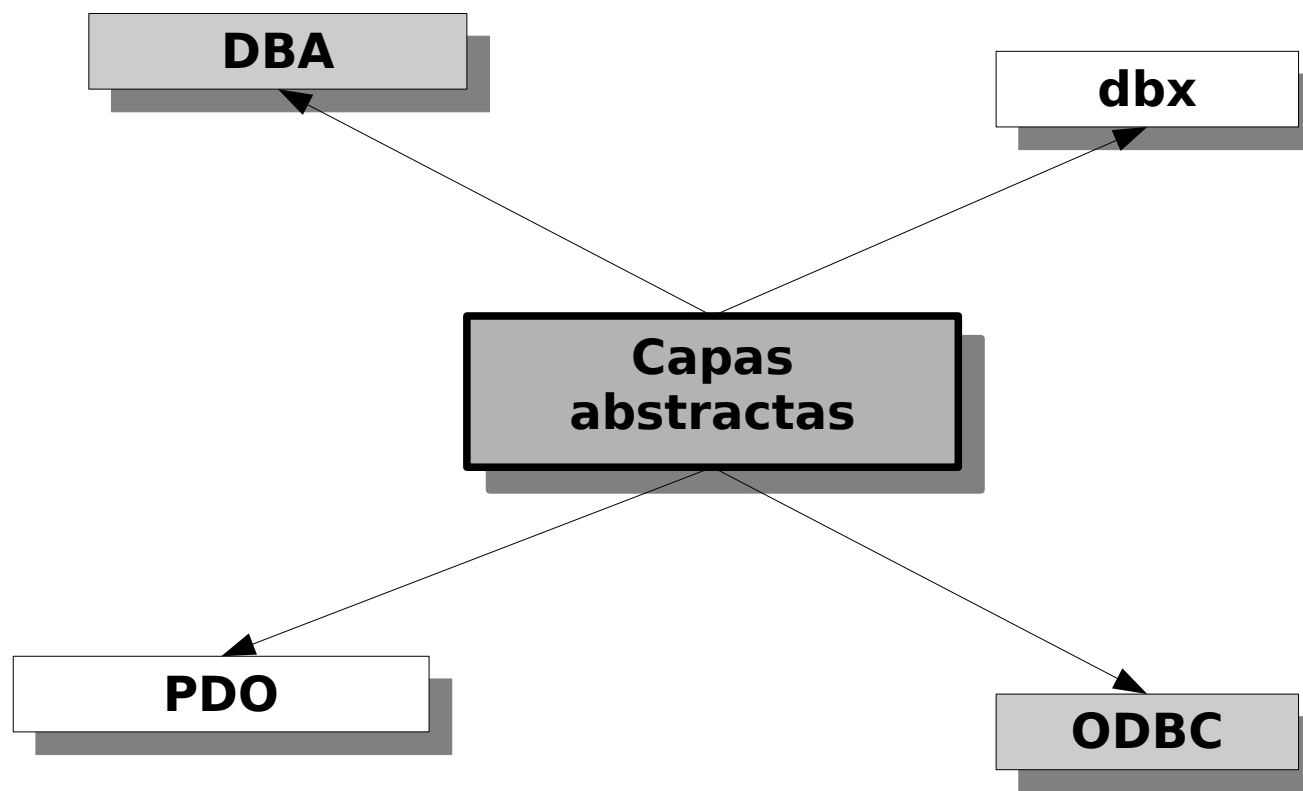
## Extensiones específicas de proveedores de DB





### Clase 7: Introducción a SQL, MySQL y PDO

## Extensiones y capas de abstracción para DB





## Clase 7: Introducción a SQL, MySQL y PDO

# Extensiones y capas de abstracción para DB

### DBA

Estas funciones permiten acceder a las bases de datos estilo Berkeley DB. Es una capa de abstracción genérica que accede a bases de datos basadas en ficheros.

### dbx

El módulo dbx es una capa de abstracción (db 'x', donde 'x' es una base de datos soportada). Las funciones de dbx permiten acceder a todas las bases de datos soportadas usando una simple convención de llamada.

### ODBC

Open DataBase Connectivity (ODBC) es un estándar de acceso a bases de datos que hace posible el acceso a cualquier dato desde cualquier aplicación, sin importar qué sistema de gestión de bases de datos almacene los datos,

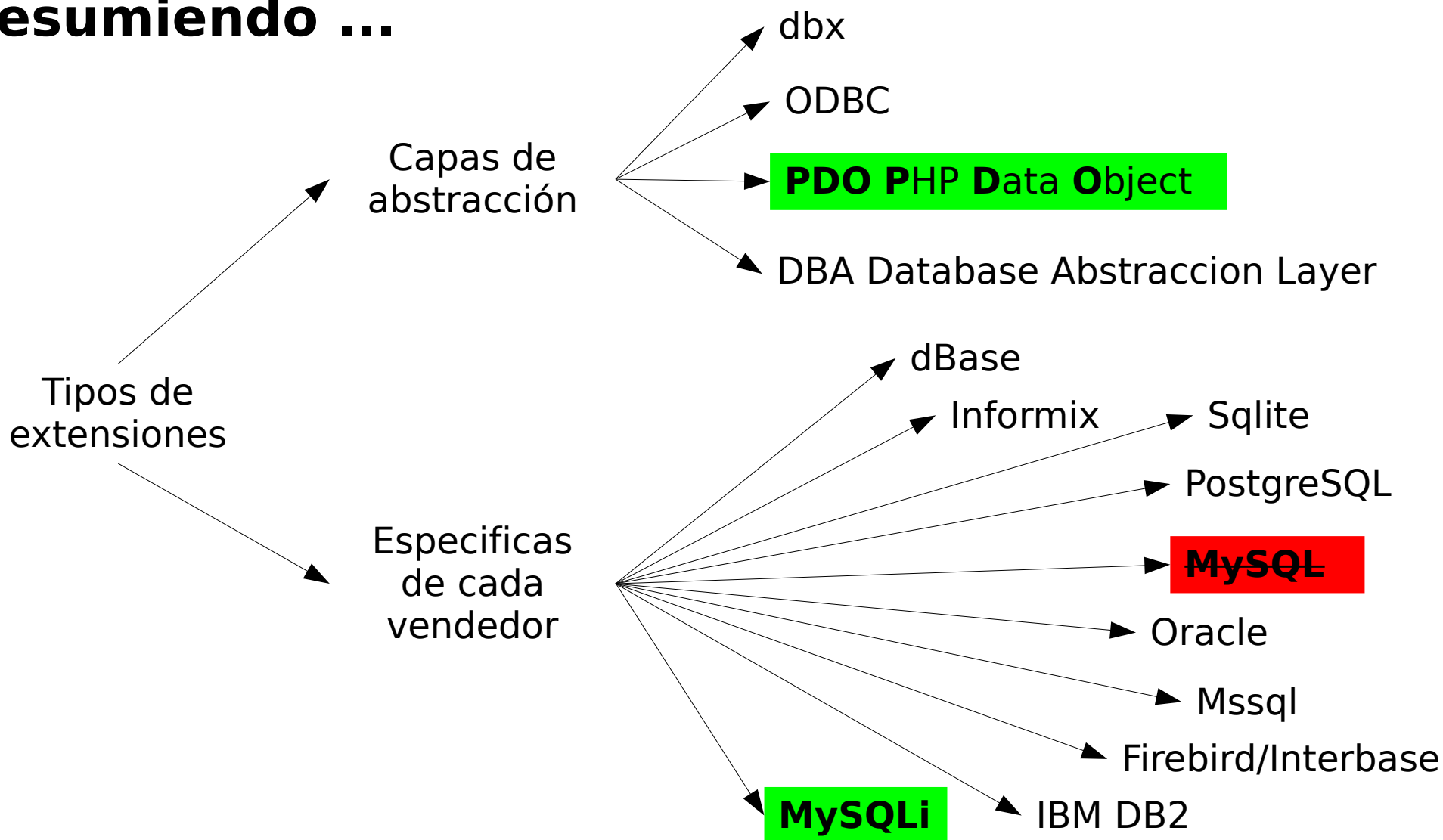
### PDO

La extensión PHP Data Objects (PDO) define un interfaz ligera, para tener acceso a bases de datos en PHP.



### Clase 7: Introducción a SQL, MySQL y PDO

## Resumiendo ...







## Capítulo 4: Programación de base de datos

~~Extensiones de base de datos en PHP~~

**Introducción a MySQL**

**SQL como lenguaje de consultas**

**Introducción a PDO**

**Usando métodos de la clase PDO**



## Clase 7: Introducción a SQL, MySQL y PDO

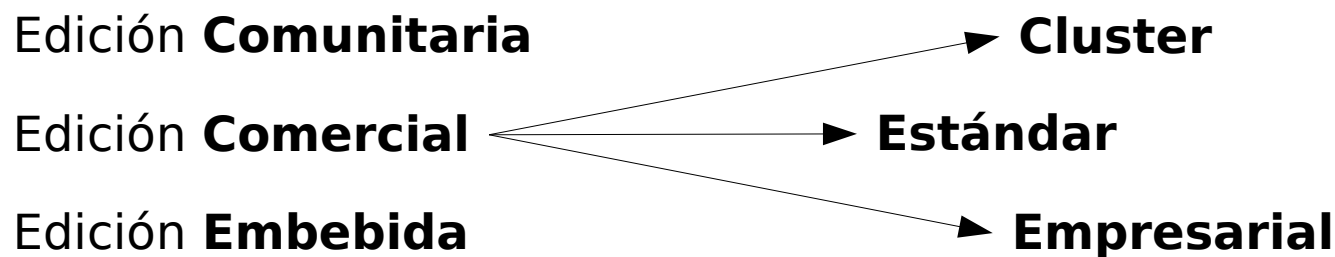
### ¿Qué es MySQL?

Es un sistema de gestión de base de datos relacionales (**SGBD**), multihilo, multiplataforma y multiusuario; con un sistema de licenciamiento dual.

En otras palabras, MySQL es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos.

### Ediciones

Posee tres líneas principales:





## Clase 7: Introducción a SQL, MySQL y PDO

### Características de MySQL

- Multiplataforma.
- Soporta varios formatos de tablas.
- Seguridad.
- Posee estabilidad.
- De simple administración.
- Soporte técnico.
- Licencia Libre o comercial.
- Manuales y ayuda en varios idiomas.
- Posee una gran comunidad.





## Clase 7: Introducción a SQL, MySQL y PDO

### Tipos de datos en MySQL

- **Cadenas de caracteres:**

CHAR  
VARCHAR  
BLOB  
TEXT  
ENUM  
SET

- **Fecha y hora:**

DATETIME  
DATE  
TIMESTAMP  
TIME  
YEAR

- **Numéricos:**

DECIMAL  
INTEGER  
TINYINT  
BIT  
BOOL  
MEDIUMINT  
BIGINT  
SMALLINT  
FLOAT  
DOUBLE





## Clase 7: Introducción a SQL, MySQL y PDO

### Tipos de tablas en MySQL

- ISAM→almacenamiento antiguo. Las tablas no podían moverse de un S.O. a otro.
- MyISAM→ reemplazo de ISAM. Las tablas son independientes al S.O.
- MERGE→ permite tratar tablas idénticas como si fueran una sola tabla.
- HEAP→ crea tablas en memoria. El reiniciarse mysql se borrarán dichas tablas.
- InnoDB→ trabaja con transacciones. Admite claves foráneas. Estándar ACID
- BerkeleyDB→ capaces de realizar COMMIT y ROLLBACK en transacciones.

**A.C.I.D.:** Atomicidad, Consistencia, Aislamiento y Durabilidad





## Capítulo 4: Programación de base de datos

~~Extensiones de base de datos en PHP~~

~~Introducción a MySQL~~

**SQL como lenguaje de consultas**

**Introducción a PDO**

**Usando métodos de la clase PDO**



## Clase 7: Introducción a SQL, MySQL y PDO

### SQL

- Es el lenguaje de manipulación más utilizado en bases de datos relacionales.
- Desarrollado entre 1974 y 1975 en IBM Research y llamado SEQUEL (Structured English QUery Language).
- Se convierte en un estándar ANSI en 1986 y en 1987 en un estándar ISO.
- Introducido por primera vez en una base de datos comercial en el año 1979 por Oracle.
- SQL se utiliza para definir, consultar y actualizar la base de datos.
- Conceptualmente es un lenguaje de definición de datos (LDD), un lenguaje de definiciones de vistas (LDV) y un lenguaje de manipulación de datos (LMD).





### Clase 7: Introducción a SQL, MySQL y PDO

## SQL - CREATE DATABASE - sintaxis

Creando una bases de datos

***CREATE DATABASE*** <NOMBRE>

Ejemplo:

**CREATE DATABASE** personal;







### Clase 7: Introducción a SQL, MySQL y PDO

## SQL - CREATE TABLE - sintaxis

Creando Tablas

```
CREATE TABLE <nombre> (  
    <columna1> <tipo> [<atributos>],  
    <columna..n> <tipo> [<atributos>]  
)
```

Ejemplo:

```
CREATE TABLE personas (  
    idpersona INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    documento INT,  
    apellido VARCHAR(255),  
    nombres VARCHAR(255),  
    direccion VARCHAR(255)  
)
```





## Clase 7: Introducción a SQL, MySQL y PDO

### SQL - CREATE INDEX - sintaxis

Creando Índices

```
CREATE INDEX <nombre_indice>  
ON <nombre_tabla>  
(<nombre_columna1>, <nombre_columna..n>)
```

Ejemplo:

```
CREATE INDEX idx_ndocu ON personas (documento);
```

Otro ejemplo con varios campos:

```
CREATE INDEX idx_documento ON personas  
(tipo_documento, documento);
```





### Clase 7: Introducción a SQL, MySQL y PDO

## SQL - DROP SCHEMA - DROP TABLE - sintaxis

Borrar una base de datos

***DROP SCHEMA*** <NOMBRE>

Ejemplo:

***DROP SCHEMA*** <personal>

Borrar una tabla

**DROP TABLE** <NOMBRE>

Ejemplo:

**DROP TABLE** personas





### Clase 7: Introducción a SQL, MySQL y PDO

## SQL - INSERT - sintaxis

Agregar datos

**INSERT INTO** <nombreTabla> **VALUES** (<valor1>, ..., valor..n>)

**INSERT INTO** <nombreTabla>(<columna1>, ..., <columna..n>)  
**VALUES** (<valor1>, ..., <valor..n>)

Ejemplo:

```
CREATE TABLE personas (  
    idpersona INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    documento INT,  
    apellido VARCHAR(255),  
    nombres VARCHAR(255),  
    direccion VARCHAR(255));
```



**INSERT INTO** personas **VALUES** (22555333,'Gómez','Francisco','Rivadavia');

**INSERT INTO** personas (documento,apellido) **VALUES** (22555333,'Gómez');



### Clase 7: Introducción a SQL, MySQL y PDO

## SQL - UPDATE - sintaxis

Modificar datos

**UPDATE** *<nombreTabla>* **SET** *columna= <nuevoValor>;*

Ejemplo:

**UPDATE** personas **SET** nombres= 'antonio';

Otro ejemplo filtrando filas:

**UPDATE** personas **SET** nombres= 'antonio' WHERE documento=22555333;





### Clase 7: Introducción a SQL, MySQL y PDO

## SQL - DELETE - sintaxis

Borrar datos

**DELETE FROM** *<nombreTabla>*;

Ejemplo:

**DELETE FROM** personas;

Otro ejemplo filtrando filas:

**DELETE FROM** personas **WHERE** documento=22555333;





### Clase 7: Introducción a SQL, MySQL y PDO

## SQL - SELECT - sintaxis

Realizando Consultas

**SELECT** \* **FROM** <nombreTabla>

Ejemplo:

**SELECT** \* **FROM** personas;

Otro ejemplo filtrando algunos campos:

**SELECT** documento, apellido **FROM** personas;

Último ejemplo filtrando algunos campos y por fila:

**SELECT** documento, apellido **FROM** personas  
**WHERE** documento=22555333;





## Capítulo 4: Programación de base de datos

~~Extensiones de base de datos en PHP~~

~~Introducción a MySQL~~

~~SQL como lenguaje de consultas~~

**Introducción a PDO**

**Usando métodos de la clase PDO**





## Clase 7: Introducción a SQL, MySQL y PDO

### Introducción a PDO

#### ¿Qué es PDO?

**P**HP **D**ata **O**bject es una extensión del lenguaje que provee una capa de abstracción para acceder a base de datos en PHP 5.

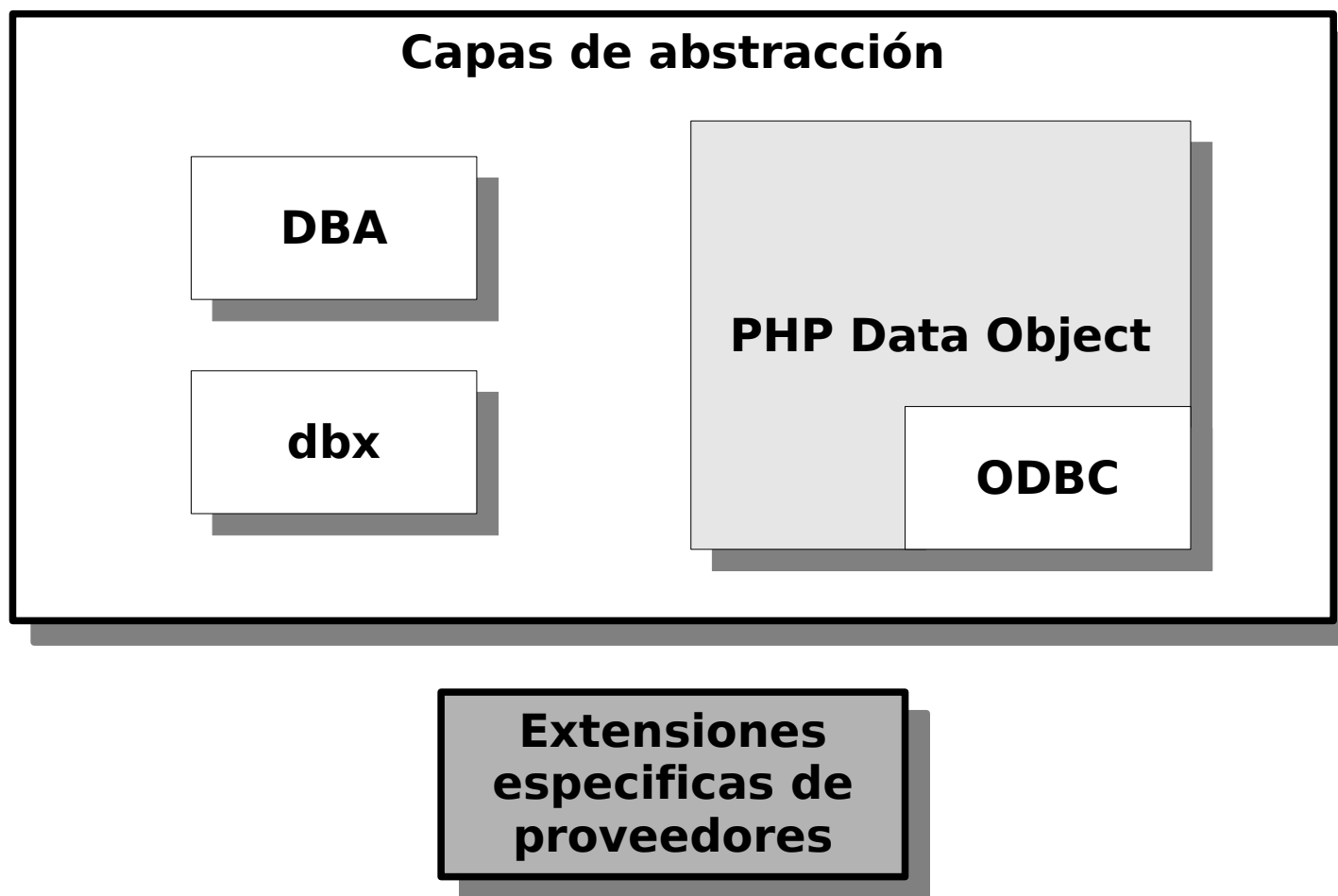
#### Historia de PDO

PDO tiene origen en el año 2003 y sale de forma experimental en la versión 5.0 de PHP como extensión PECL.  
En PHP 5.1, PDO aparece como una extensión estable que se incluye por defecto.



### Clase 7: Introducción a SQL, MySQL y PDO

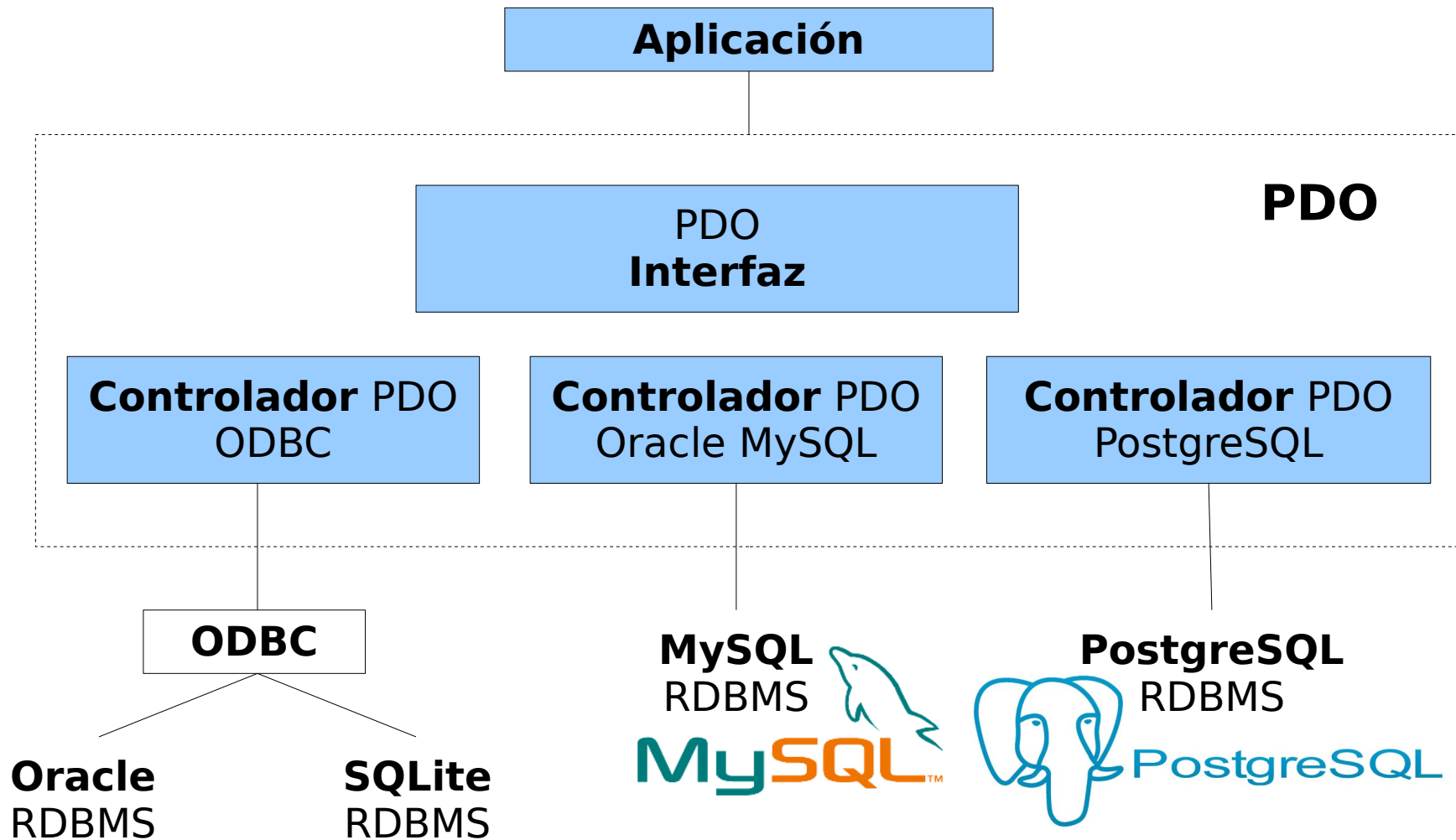
## Introducción a PDO - Extensiones





### Clase 7: Introducción a SQL, MySQL y PDO

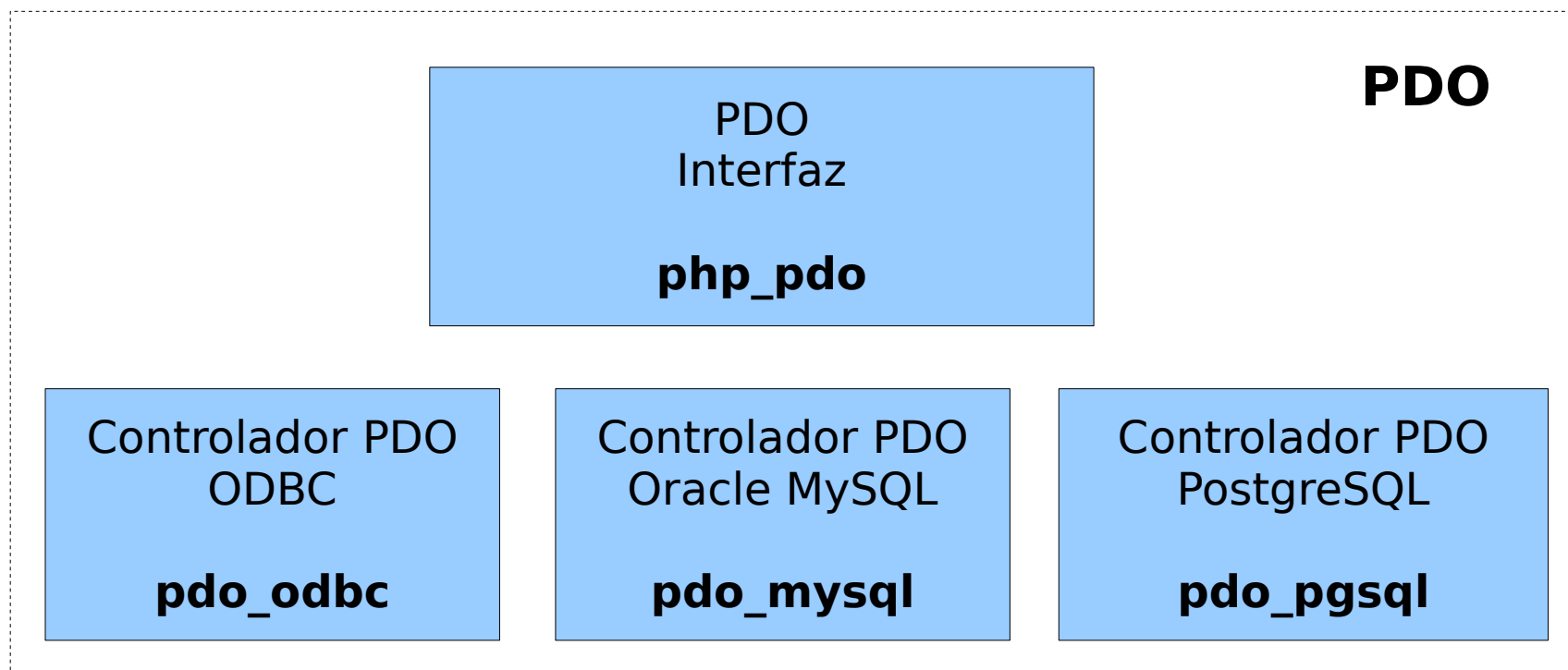
## Introducción a PDO - Arquitectura





### Clase 7: Introducción a SQL, MySQL y PDO

## Introducción a PDO - Arquitectura





## Clase 7: Introducción a SQL, MySQL y PDO

### Introducción a PDO - Controladores

<b>PDO_DBLIB</b>	FreeTDS / Microsoft SQL Server / Sybase
<b>PDO_FIREBIRD</b>	Firebird / Interbase 6
<b>PDO_IBM</b>	IBM DB2
<b>PDO_INFORMIX</b>	IBM Informix Dynamic Server
<b>PDO_MYSQL</b>	Oracle MySQL 3.x / 4.x / 5.x
<b>PDO_OCI</b>	Oracle Call Interface
<b>PDO_ODBC</b>	ODBC v3 (IBM DB2, unixODBC and win32 ODBC)
<b>PDO_PGSQL</b>	PostgreSQL
<b>PDO_SQLITE</b>	SQLite 3 and SQLite 2
<b>PDO_4D</b>	4D



### Clase 7: Introducción a SQL, MySQL y PDO

## Introducción a PDO - Clases y métodos

### Clase PDO

Representa una conexión entre PHP y un servidor de bases de datos.

### Clase PDOStatement

Representa una instrucción preparada y después que la instrucción es ejecutada, un conjunto de resultados (result set).

### Clase PDOException

Representa un error lanzado por PDO.



## Clase 7: Introducción a SQL, MySQL y PDO

# Introducción a PDO - Clases PDO

## Clase PDO y sus métodos

**PDO::beginTransaction()**

Inicializa una transacción.

**PDO::commit()**

Ejecuta una transacción.

**PDO::\_\_construct()**

Crea una instancia PDO que representa una conexión a una base de datos.

**PDO::errorCode()**

Recupera el SQLSTATE asociado con la última operación en la base de datos.

**PDO::errorInfo()**

Recupera información extendida del error asociado con la ultima operación en la base de datos.



## Clase 7: Introducción a SQL, MySQL y PDO

# Introducción a PDO - Clases PDO

## Clase PDO y sus métodos

### **PDO::exec()**

Ejecuta una instrucción SQL y retorna el número de filas afectadas.

### **PDO::getAttribute()**

Recupera un atributo de conexión a base de datos.

### **PDO::getAvailableDrivers()**

Retorna un array (arreglo) de los drivers disponibles en la extensión PDO.

### **PDO::lastInsertId()**

Retorna el ID (identificador) de la última fila insertada o secuencia de valores.

### **PDO::prepare()**

Prepara una instrucción para ejecución y retorna un objeto de tipo PDOStatement.





## Clase 7: Introducción a SQL, MySQL y PDO

# Introducción a PDO - Clases PDO

## Clase PDO y sus métodos

### **PDO::query()**

Ejecuta una instrucción SQL, retornando un conjunto de resultados (resul set) como un objeto de tipo PDOStatement.

### **PDO::quote()**

Quotes (Pone entre comillas simples un string) para uso en una query (consulta).

### **PDO::rollBack()**

Vuelve atrás una transacción (Rolls back).

### **PDO::setAttribute()**

Modifica un atributo.



### Clase 7: Introducción a SQL, MySQL y PDO

## Introducción a PDO - Instalación y configuración

En sistemas operativos **Windows** existe una instalación por defecto de la implementación **PDO** y el driver de **PDO\_SQLite activado**.

En sistemas operativos **Linux** también existe una instalación por defecto de la implementación **PDO** y el driver de **PDO\_SQLite activado**.

Para agregar cualquier otro controlador de base de datos a PDO, simplemente se debe activar el mismo en el archivo **php.ini**. Para activar el **pdo** de **mysql** se debe descomentar la siguiente línea:

**#extension=php\_pdo\_mysql.dll** (En Windows)

O

**#extension=php\_pdo\_mysql.so** (En GNU/Linux)



## Clase 7: Introducción a SQL, MySQL y PDO

### Constantes predefinidas

PDO::PARAM\_BOOL (integer)  
PDO::PARAM\_NULL (integer)  
PDO::PARAM\_INT (integer)  
PDO::PARAM\_STR (integer)  
PDO::PARAM\_LOB (integer)  
PDO::PARAM\_STMT (integer)  
PDO::PARAM\_INPUT\_OUTPUT (integer)  
PDO::FETCH\_LAZY (integer)  
PDO::FETCH\_ASSOC (integer)  
PDO::FETCH\_NAMED (integer)  
PDO::FETCH\_NUM (integer)  
PDO::FETCH\_BOTH (integer)  
PDO::FETCH\_OBJ (integer)  
PDO::FETCH\_BOUND (integer)  
PDO::FETCH\_COLUMN (integer)  
PDO::FETCH\_CLASS (integer)  
PDO::FETCH\_INTO (integer)  
PDO::FETCH\_FUNC (integer)

PDO::FETCH\_GROUP (integer)  
PDO::FETCH\_UNIQUE (integer)  
PDO::FETCH\_KEY\_PAIR (integer)  
PDO::FETCH\_CLASSTYPE (integer)  
PDO::FETCH\_SERIALIZE (integer)  
PDO::FETCH\_PROPS\_LATE (integer)  
PDO::ATTR\_AUTOCOMMIT (integer)  
PDO::ATTR\_PREFETCH (integer)  
PDO::ATTR\_TIMEOUT (integer)  
PDO::ATTR\_ERRMODE (integer)  
PDO::ATTR\_SERVER\_VERSION  
(integer)  
PDO::ATTR\_CLIENT\_VERSION (integer)  
PDO::ATTR\_SERVER\_INFO (integer)  
PDO::ATTR\_CONNECTION\_STATUS  
(integer)  
PDO::ATTR\_CASE (integer)  
PDO::ATTR\_CURSOR\_NAME (integer)



### Clase 7: Introducción a SQL, MySQL y PDO

## Constantes predefinidas

PDO::ATTR\_CURSOR (integer)  
PDO::ATTR\_DRIVER\_NAME (string)  
PDO::ATTR\_ORACLE\_NULLS (integer)  
PDO::**ATTR\_PERSISTENT** (integer)  
PDO::ATTR\_STATEMENT\_CLASS (integer)  
PDO::ATTR\_FETCH\_CATALOG\_NAMES (integer)  
PDO::ATTR\_FETCH\_TABLE\_NAMES (integer)  
PDO::ATTR\_STRINGIFY\_FETCHES (integer)  
PDO::ATTR\_MAX\_COLUMN\_LEN (integer)  
PDO::ATTR\_DEFAULT\_FETCH\_MODE (integer)  
PDO::ATTR\_EMULATE\_PREPARES (integer)  
PDO::ERRMODE\_SILENT (integer)  
PDO::ERRMODE\_WARNING (integer)  
PDO::ERRMODE\_EXCEPTION (integer)  
PDO::CASE\_NATURAL (integer)  
PDO::CASE\_LOWER (integer)  
PDO::CASE\_UPPER (integer)

y más ...



## Capítulo 4: Programación de base de datos

~~Extensiones de base de datos en PHP~~

~~Introducción a MySQL~~

~~SQL como lenguaje de consultas~~

~~Introducción a PDO~~

**Usando métodos de la clase PDO**



## Clase 7: Introducción a SQL, MySQL y PDO

### Usando métodos de la clase PDO

**PDO::\_\_construct** — Crea una instancia de la clase que representa una conexión.

#### Sintaxis

**PDO::\_\_construct()** ( string `$dsn` [, string `$nombreUsuario` [, string `$contrasenia` [, array `$opciones_control` ]]] )

#### Descripción

Este método crea una instancia de la clase PDO que representa una conexión a base de datos. Retorna un objeto PDO si la conexión fue exitosa y en caso contrario se lanza una excepción PDO (PDOException).





### Capítulo 4: Programación de base de datos

## Cadena de conexión DSN para PDO

**{dbtype}:dbname={dbname};host={dbhost};port={dbport}**

**mysql:dbname=phpn1\_db\_clase7;host=127.0.0.1;port=3306**







## Clase 7: Introducción a SQL, MySQL y PDO

### Métodos de la clase PDO

**PDO::exec** — Ejecuta una declaración SQL y retorna el número de filas afectadas

#### Sintaxis

int **PDO::exec** ( string \$SqlDeclarada )

#### Descripción

Este método ejecuta una SQL declarada y retorna el número de filas afectadas, en caso de error devuelve false o cero 0 o "". Este método no devuelve resultados para la sentencia SELECT de SQL.







## Clase 7: Introducción a SQL, MySQL y PDO

### Métodos de la clase PDO

**PDO::getAttribute** — Obtiene el valor de un atributo de la conexión a la base de datos.

#### Sintaxis

mixed **PDO::getAttribute** ( int \$atributo )

#### Descripción

Este método devuelve el valor de una atributo del manejador de la base de datos. Se recomienda utilizar las constantes asociadas a la conexión de la base de datos, estas siempre comienzan de la forma **PDO::ATTR\_\***.





## Clase 7: Introducción a SQL, MySQL y PDO

### Métodos de la clase PDO

**PDO::setAttribute** — Establece el valor de un atributo de la conexión a la base de datos.

#### Sintaxis

bool **PDO::setAttribute** ( int \$atributo, mixed \$valor )

#### Descripción

Este método establecer un nuevo valor a un atributo del manejador de la base de datos. Se recomienda utilizar las constantes asociadas a la conexión de la base de datos, estas siempre comienzan de la forma **PDO::ATTR\_\***. Si pudo establecer el valor devuelve *true* y sino *false*.





## Clase 7: Introducción a SQL, MySQL y PDO

### Métodos de la clase PDO

**PDO::errorCode** — Obtiene el SQLSTATE asociado con la última operación sobre el manejador de la base de datos

#### Sintaxis

mixed **PDO::errorCode** ( void )

#### Descripción

Este método retorna un SQLSTATE asociado a la última operación sobre el manejador de la DB, identificado con 5 caracteres alfanuméricos según lo define el estándar ANSI SQL-92. Retorna NULL cuando no se ha ejecutado un método.





## Clase 7: Introducción a SQL, MySQL y PDO

### Métodos de la clase PDO

**PDO::errorInfo** — Obtiene información extendida del error asociado con la última operación sobre el manejador de la base de datos

#### Sintaxis

array **PDO::errorInfo** ( void )

#### Descripción

Este método un arreglo con información acerca del error ocurrido en la última operación sobre el manejador de la base de datos. El arreglo consiste de los siguientes campos:

- 0 **SQLSTATE** código de error (ANSI 92)
- 1 **Código de error** específico del controlador de base de datos.
- 2 **Mensaje de error** específico del controlador de la base de datos.





## Clase 7: Introducción a SQL, MySQL y PDO

### Ejemplo de uso

```
<?php

$dsn = 'mysql:host=127.0.0.1;port=3306';
$user = 'root';
$password = '';

$dbh = new PDO($dsn, $user, $password);

$driversPDO = $dbh->getAvailableDrivers();

print_r($driversPDO);

$dbh = null;
```





### Clase 7: Introducción a SQL, MySQL y PDO

## Ejemplo de uso

```
<?php

$dsn = 'mysql:host=127.0.0.1;port=3306';
$user = 'root';
$password = '';

$dbh = new PDO($dsn, $user, $password);

// Muestra información del estado del servidor de base de datos
echo $dbh->getAtributtes(PDO::ATTR_SERVER_INFO);

echo "<br/>";

// Muestra la versión del gestor de la base de datos
echo $dbh->getAtributtes(PDO::ATTR_SERVER_VERSION);

$dbh = null;
```





## Clase 7: Introducción a SQL, MySQL y PDO

### Ejemplo de uso

```
<?php

$dsn = 'mysql:host=127.0.0.1;port=3306';
$user = 'root';
$password = '';

$dbh = new PDO($dsn, $user, $password);

$result = $dbh->exec('CREATE DATABASE phpn1_db_clase7;');

if (!$result) {
    print_r($dbh->errorInfo());
}

$result = $dbh->exec('USE phpn1_db_clase7;');

$dbh = null;
```







### Clase 7: Introducción a SQL, MySQL y PDO

## Ejemplo de uso

```
<?php
```

```
$dsn = 'mysql:dbname=phpn1_db_clase7;host=127.0.0.1;port=3306';  
$user = 'root';  
$password = '';
```

```
$declaracionSQL = "CREATE TABLE persona (id_persona INT NOT NULL  
AUTO_INCREMENT PRIMARY KEY, apellidos VARCHAR(100) NOT NULL, domicilio  
VARCHAR(200));";
```

```
$dbh = new PDO($dsn, $user, $password);
```

```
$result = $dbh->exec($declaracionSQL);
```

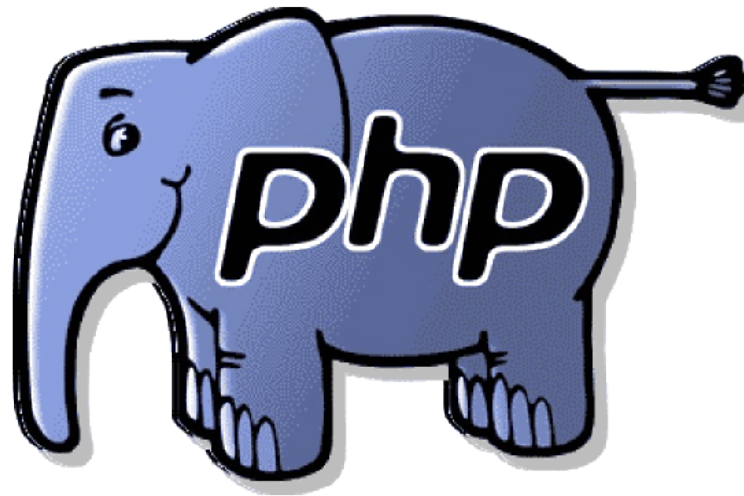
```
If (!$result)  
    print_r($dbh->errorInfo());
```

```
$dbh = null;
```





**¿Dudas?**



**¿Consultas?**



## Información de contacto

### Web:

<http://www.gugler.com.ar>

<http://campusvirtual.gugler.com.ar>

<http://www.facebook.com/gugler.com.ar>

<http://www.twitter.com/cgugler>

### Mail:

[contacto@gugler.com.ar](mailto:contacto@gugler.com.ar)

[academica@gugler.com.ar](mailto:academica@gugler.com.ar)

[administracion@gugler.com.ar](mailto:administracion@gugler.com.ar)