Una Interfaz de Programación de Aplicaciones, o **API**, define las clases, métodos, funciones y variables que necesitará llamar una aplicación para llevar a cabo una tarea determinada.

En el caso de las **aplicaciones de PHP que necesitan comunicarse con un servidor de bases de datos**, **las APIs** necesarias **se ofrecen** generalmente **en forma de extensiones de PHP**.

Las **APIs** pueden ser **procedimentales** u **orientadas a objetos**.

- Con una API procedimental se invocan funciones para llevar a cabo las tareas,
- Con una API orientada a objetos se instancian clases, y luego se invocan a métodos de los objetos creados.

Entre ambas opciones, la segunda es generalmente la recomendada, ya que es más actualizada y conlleva una mejor organización de código.

PHP ofrece tres API´s para conectar a un servidor de base de datos MySQL

Extensión MySQL

- Es la extensión original que permitia a aplicaciones PHP interactuar con bd MySQL
- Proporciona una interfaz procedural
- Pensada para usar sólo con versiones anteriores a MySQL 4.1.3
- Se puede usar también con versiones posteriores, pero no estan disponibles todas las funcionalidades del servidor MySQL.

Extensión mysqli

- Es la extensión de MySQL *mejorada*
- Desarrollada para MySQL 4.1.3 o posterior.
- Incluida en versiones PHP 5 y posteriores.
- Proporciona una interfaz dual: Procedural y Orientada a Objetos
- Mejoras que presenta frente a la extensión *mysql* son:
  - Interfaz orientada a objetos
  - Soporte para Declaraciones Preparadas
  - Soporte para Múltiples Declaraciones
  - Soporte para Transacciones

Objetos de Datos de PHP (**PDO**)

- Son una capa de abstracción de bases de datos específicas para aplicaciones PHP.
- Ofrece una API homogénea, en teoria , independientemente del tipo de servidor de bases de datos con el que se vaya a conectar la aplicación.
- Proporciona una interfaz orientada a objeto
- Su mayor inconveniente es que no permite utilizar todas las funcionalidades ofrecidas por el servidor MySQL.

Comparación de funcionalidades de los tres métodos para conectar a MySQL desde PHP

	Extension MySQL	Extension mysqli	PDO
Con version PHP	Antes 4.0	5.0	5.0
Incluido en PHP 5.x	Si	Si	Si
Estado de desarrollo	Solo se mantiene	Activo	Activo desde 5.3
Utilización	No	Si (Recomendada)	Si
Soporte sentencias preparadas	No	Si	Si
Soporte procedimientos almacenados	No	Si	Si
Soporte sentecias multiples	No	Si	Mayormente

Interfaz dual

La extensión mysqli ofrece soporte para programación procedimental y programación orientada a objetos.

La interfaz procedimental es similar a la de la extensión mysql, y en la mayoría de los casos, los nombres de funciones difieren únicamente por el prefijo.

Algunas funciones de mysqli requieren como primer argumento un identificador de conexión, mientras que en las funciones similares de la interfaz mysql este dato era el último argumento y además opcional.

	Interfaz Procedimental		
Fase	Extension mysql		Extension **Mysqli**
Conexion	`$Id = mysql_connect("localhost", "user", "key");` `$ok = mysql_select_db("test");`		`$Id = mysqli_connect("127.0.0.1", "user", "key","test");`
Consulta	`$accion = "select nif from T1" ;` `$res = mysql_query( $accion , $Id );`		`$accion = "select nif from T1" ;` `$res = mysqli_query( $Id , $accion );`

Acceso a resultado	`echo mysql_num_rows( $res );`
	`$fila = mysql_fetch_assoc($res);`
	`echo $fila['nif'];`
			`echo mysqli_num_rows( $res );`
			`$fila = mysqli_fetch_assoc($res);`
			`echo $fila['nif'];`

Cierre conexión	`$ok = mysql_free_result($res);` `mysql_close($Id);`		`$ok = mysqli_free_result($res);` `mysqli_close($Id);`

	Interfaz Orientada a Objeto
Fase	Extension **Mysqli**
Conexion	`$Id = new mysqli("127.0.0.1", "user", "key", "test");` // Crea objeto $Id de la clase mysqli `if ( $Id->connect_errno ) {` // Sí el código de error es distinto de 0 `echo "Fallo al conectar a MySQL: " . $Id->connect_error ;` // muestra mensaje del error `}`
Consulta	`$accion = "select nif from T1" ;` // Establece mandato a ejecutar en la BD `$res = $Id->query( $accion ) ;` // Ejecuta el mandato
Acceso a resultado	`echo $Id->affected_rows ;` // Muestra nº de filas afectadas por el mandato `$res->data_seek(0);` // Apunta a fila 0 del resultado obtenido `$fila = $res->fetch_assoc();` // Extrae la fila apuntada y la guarda en $fila `echo $fila['nif'];` // Muestra valor
Cierre conexión	`$Id->close()` // cierra la conexión

Ejemplo: Muestra datos de una tabla

```php
<?php
  $conex1 = new mysqli("localhost", "mi_usuario", "mi_contraseña", "empresa");  // Abre una conexión
  if (mysqli_connect_errno()) {               // Comprueba conexión
     printf("Conexión fallida: %s\n", mysqli_connect_error());
     exit();
    }
 $query = "SELECT Apellido, Oficio  FROM Empleados ORDER BY Apellido";
 if ($result  =  $conex1->query( $query)) {         // Sí hay resultados
      $result->data_seek(0);                        // Apunta  a la primera fila
      while ( $fila = $result->fetch_assoc()) {     // Extrae fila apuntada  y apunta a la siguiente
          printf ("Apellido: %s  Oficio: %s\n", $fila['apellido'], $fila['oficio'] );    // Muestra sus datos
       }
   //  Extrae filas del resultado en orden inverso
      for ( $num_fila  = $result->num_rows  -  1; $num_fila >= 0; $num_fila--) {
          $result->data_seek($num_fila);
          $row = $result->fetch_row();
          printf ("Apellido: %s  Oficio: %s\n", $row[0], $fila[1] );
       }

      $result->close();    // libera  recursos
    }
  $conex1->close();  // cierra conexión
?>
```