**getColumns($table\_name = false){}**

**TRUE**

**-return $response (arreglo de registros con el total de campos de la tabla)**

Recibe como parámetro opcional una cadena que indica el nombre de la tabla de la cual se quiere sacar los campos.

Esta función nos retorna un arreglo de registros con la información de todos los campos (nombre, tipo de dato, longitud etc) de una tabla “x”, en caso de definir un nombre en el parámetro se retornara el arreglo de registros con todos los campos de la tabla especificada, de no definir un nombre la función retornara los campos de la tabla del modelo activo.

**searchByQuery (**String **$query){}**

**TRUE**

**-return $this (el objeto activo en memoria, ya preparado con los atributos de la tabla afectada)**

**FALSE**

**-return false (en caso de no devolver registros)**

Recibe como parámetro una cadena que indica la consulta que se va a ejecutar

Esta función nos permite ejecutar todo tipo de consulta que le pasemos de las cuales siempre nos retorne más de un registro, la función recibe una cadena con la consulta y mediante PDO realiza un fetchall() a los datos devueltos armando un arreglo general donde cada subíndice dentro del mismo corresponde a otro arreglo con la información del registro, a su vez la función se adapta de tal manera que si la consulta devuelve más de un resultado retorna un arreglo de registros de tal forma que accedamos a los valores indicando el sub índice para hacer referencia al registro “x” seguido del nombre del campo al que queremos sacar la información, en cambio sí devuelve un solo registro podemos acceder indicando el nombre de la columna solamente.

Para consultas que retornen solo un registro consulte **searchByQueryOne().**

**searchByQueryOne(**String **$query){}**

**TRUE**

**-return $this (el objeto activo en memoria, ya preparado con los atributos de la tabla afectada)**

**FALSE**

**-return false (en caso de no devolver registros)**

Recibe como parámetro una cadena que indica la consulta que se va a ejecutar

Esta función nos permite ejecutar todo tipo de consulta que le pasemos de las cuales siempre nos retorne un solo registro, la función recibe una cadena con la consulta y mediante PDO realiza un fetch() a los datos devueltos, a su vez la función instancia el modelo activo en memoria y va asignando como atributos de la misma cada columna devuelta en la consulta, de tal manera que podamos acceder a la información por medio de la clase instanciada en ese momento.

Para consultas que retornen más de un registro consulte **searchByQuery().**

**Las siguientes son funciones para el manejo dinámico de la información por medio de la abstracción de modelo, se puede jugar con todas ellas para llegar a un resultado deseado.**

**findAll(Array $condicion = []){}**

**TRUE**

**-return $registros (arreglo de registros con toda la información de la tabla del modelo activo)**

**FALSE**

**-return false (en caso de no devolver registros)**

Recibe como parámetro un arreglo que indica una condición (tener en cuenta que la condición ha de ser aplicable sobre un solo campo, ya que se concatenara un solo where donde se cumpla la condición especificada).

Esta función nos permite consultar todos los registros que haya dentro de la tabla del modelo activo, se nos retornara un arreglo de registros con toda la información disponible dentro de la tabla del modelo activo, opcionalmente se puede pasar un parámetro de tipo arreglo (["campo"=>”condición”]) aplicando una condición sobre la consulta final teniendo en cuenta que la función puede retornar 1 o varios registros aún con la condición dada.

**find (){}**

**TRUE**

**-return (arreglo de registros generales, es necesario indicar una de las funciones auxiliares all() u one() para preparar el arreglo a retornar)**

**FALSE**

**-return false (en caso de no devolver registros)**

Función que nos permite obtener bien sea todos, o un solo registro de la tabla del modelo activo, es necesario especificar la funcion all() o la funcion one() para establecer cuantos registros se retornaran.

La misma puede ser mezclada con la funcion where() para que tome todos los registros en base a una condición, a su vez esta última debe pasarse con una de las funciones auxiliares antes mencionadas para preparar el arreglo final a retornar.

**where (Array $condicion = []){}**

**TRUE**

**-return (arreglo de registros generales, es necesario indicar una de las funciones auxiliares all() u one() para preparar el arreglo a retornar)**

**FALSE**

**-return false (en caso de no devolver registros)**

Recibe como parámetro un arreglo que indica una condición (tener en cuenta que la condición ha de ser aplicable sobre un solo campo, ya que se concatenara un solo where donde se cumpla la condición especificada).

Esta función nos permite obtener un arreglo bien sea con todos o un solo registro de la tabla del modelo activo basándose en una condición, adicionalmente se puede mezclar con la función and() para concatenar un and al final del where.

Es necesario indicar mediante una de las funciones auxiliares all() u one(), de qué forma se preparara el arreglo a retornar.

**and (Array $condicion = []){}**

**TRUE**

**-return (arreglo de registros generales, es necesario indicar la función where() antes de llamar esta función)**

**FALSE**

**-return false (en caso de no devolver registros)**

Recibe como parámetro un arreglo que indica una condición (tener en cuenta que la condición ha de ser aplicable sobre un solo campo, ya que se concatenara un solo and donde se cumpla la condición especificada).

Esta función sirve como auxiliar de la función where() (donde es necesario llamar esta última primero) y lo que hace es concatenar un “and” para especificar aún más la consulta.

**all (){}**

**TRUE**

**-return (arreglo de todos los registros del modelo activo)**

**FALSE**

**-return false (en caso de no devolver registros)**

Función auxiliar que permite preparar los registros devueltos, simplemente la función capta la consulta que se pasa desde cualquiera de las funciones anteriormente mencionadas aplicandole un fetchall() y retornando un arreglo con todos los registros de la tabla del modelo activo.

**one (){}**

**TRUE**

**-return (un objeto del registro, donde cada atributo se corresponde con la columna dentro de la tabla)**

**FALSE**

**-return false (en caso de no devolver registros)**

Función auxiliar que permite preparar los registros devueltos, esta simplemente capta la consulta que es preparada por las funciones anteriormente mencionadas, y aplica un fetch() para devolver un solo registro, posteriormente lo transforma en un objeto para poder acceder a la información mediante un puntero a su atributo.

Tanto all() como one() captan las consultas preparadas según sea el caso del uso de las funciones anteriores, en un supuesto caso de usar la funcion find() seguido de la funcion one() esta última aplicara un select \* from tabla limit 1 para retornar un solo registro y volverlo un objeto, y en el caso de all() aplicara un select \* from tabla para devolver un arreglo de registros.

La razón de hacer uso de las funciones auxiliares all y one es para tener una forma de obtener acceso a la información, ya que si se usa un find() solo este devolverá un objeto con arreglos de todos los registros y la información será inaccesible, allí es donde entra en juego las funciones auxiliares para preparar el retorno de la información en forma de arreglo o un objeto.

**requestPost(){}**

**TRUE**

**-return (un objeto del modelo activo, con los atributos establecidos en el formulario)**

**FALSE**

**-return false (en caso de que algún atributo en el formulario no se encuentre dentro de la tabla del modelo activo)**

requestPost() es una función que recibe la información enviada por post, y crea un objeto del modelo activo, valida cada atributo definido dentro del formulario y devuelve un objeto donde los atributos son los definidos en el formulario y sus respectivos valores

De esta forma una vez que se llama esta función podremos obtener acceso rápido a la información que introdujo el usuario en “x” campo del formulario.

**save(){}**

**TRUE**

**-return true**

**FALSE**

**-return false (en caso de que haya ocurrido un error al ejecutar la sentencia)**

save() nos permite guardar los registros que son enviamos por post desde el formulario, la misma recoge los datos del formulario a través de un modelo activo, de esta manera nos permite alterar la información en un momento dado, también es importante saber que la función hace un llamado a otra función estática dentro de la misma clase llamada insertCommand() quien es la encargada de preparar la sentencia, y enviar los valores hacia la base de datos.

**update(int $id){}**

**TRUE**

**-return true**

**FALSE**

**-return false (en caso de que haya ocurrido un error al ejecutar la sentencia)**

update() nos permite actualizar los valores que son enviados a través del formulario, esta función actualiza los valores que son enviados a través del modelo, permitiéndonos en un momento dado alterar los valores y enviar los nuevos a actualizar.

Esta función recibe como parámetro un numero entero indicando el id del registro que se desea modificar, también hace un llamado a la función estática updateCommand() la cual prepara la sentencia para actualizar el registro.

**findModel(int $id){}**

**TRUE**

**-return true (retorna un arreglo con la información del registro solicitado)**

**FALSE**

**-return false (en caso de que haya ocurrido un error al ejecutar la sentencia)**

Recibe como parámetro un número entero el cual nos permitirá filtrar la información dentro de la tabla.

Esta función nos permite hacer una búsqueda dentro del modelo activo, es decir buscar la información de un registro “x” solicitado, y nos devuelve un arreglo con todos los campos y la información del mismo.