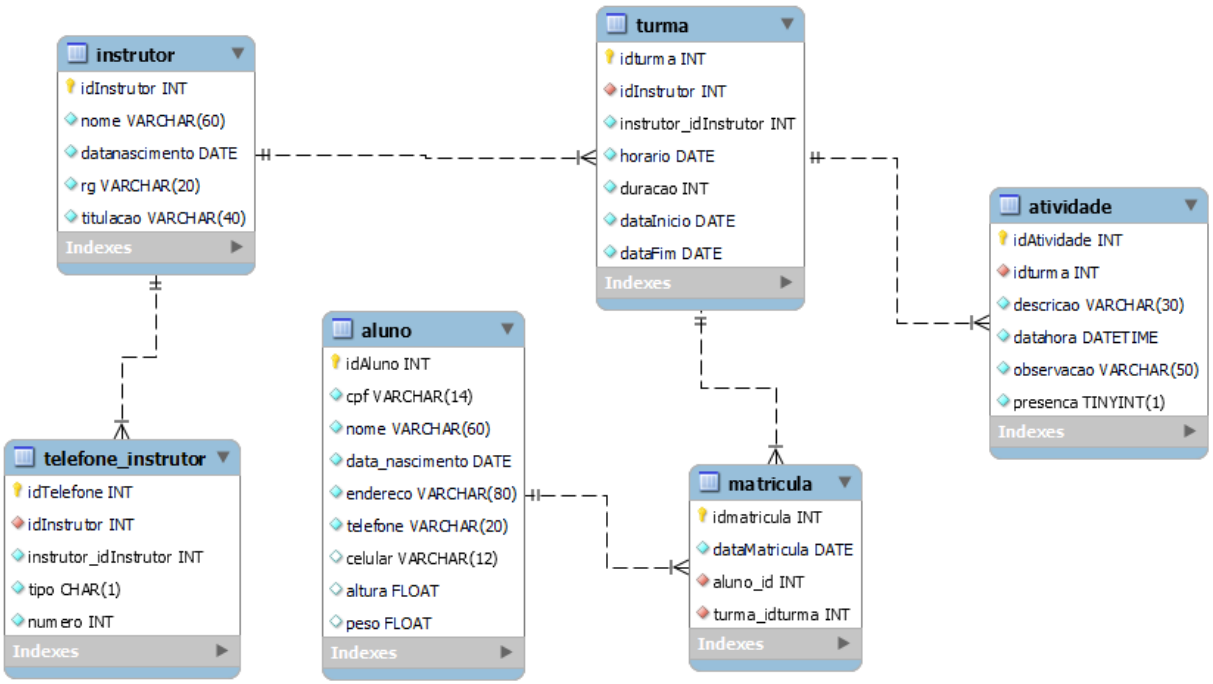
CRUD com Boostrap, PHP e MySQL

A maior parte dos sistemas em PHP é feito com um banco de dados no back end, e normalmente, esse banco de dados é o MySQL. Por isso, é importante criar essa camada de acesso a dados de uma forma bem genérica e independente, para que depois ela possa ser reaproveitada em todo o sistema.

Neste tutorial, você vai ver como deve ser esse banco de dados para o nosso sistema de CRUD e como criá-lo, executando o SQL pelo phpMyAdmin. Depois disso, vamos criar um arquivo de configurações para usar em todo o sistema. E, por fim, vamos criar o script de conexão com esse banco de dados, usando PHP.

Passo 1: Crie as seguintes tabelas no banco de dados

Para este tutorial, vamos criar um banco de dados bem simples, com algumas tabelas, conforme o modelo:



Passo 2: Crie o Arquivo de Configurações do Sistema

Agora, vamos criar um arquivo de configurações, que vai guardar todos os dados que vão ser usados em todos os outros scripts PHP do sistema.

Crie um arquivo chamado config.php, na pasta principal, e coloque o código a seguir:

```
<?php

    /** O nome do banco de dados */
    define('DB_NAME', 'nome_do_banco');

    /** suário do banco de dados MySQL */
```

```
define('DB_USER', 'nome_do_usuário');

/** Senha do banco de dados MySQL */
define('DB_PASSWORD', 'senha_do_banco');
/** Nome do host **/
define('DB_HOST', 'caminho_do_host');

/** Caminho absoluto para a pasta do sistema **/
if ( !defined('ABSPATH') )
        define('ABSPATH', dirname(__FILE__) . '/');

/** Caminho do servidor para o sistema **/
if ( !defined('BASEURL') )
        define('BASEURL', '/crud/');

/** Caminho do arquivo do banco de dados **/
if ( !defined('DBAPI') )
        define('DBAPI', ABSPATH . 'inc/database.php');

/** Seta o cabeçalho e rodapé
define('HEADER_TEMPLATE', ABSPATH . 'inc/header.php');
('FOOTER_TEMPLATE', ABSPATH . 'inc/footer.php');
?>
```

Passo 3: Implemente o script de Conexão com o Banco de Dados

Vamos criar um arquivo que vai ter todas as funções de acesso ao banco de dados. Assim, podemos reaproveitar código nas outras partes do CRUD.

Crie um arquivo chamado database.php, na pasta inc do seu projeto, e coloque o código a seguir:

```
<?php
    mysqli_report(MYSQLI_REPORT_STRICT);
    function open_database() {
        try {
            $conn = new mysqli(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
            return $conn;
        } catch (Exception $e) {
            echo $e->getMessage();
            return null;
        }
    }

    function close_database($conn) {
        try {
            mysqli_close($conn);
        } catch (Exception $e) {
            echo $e->getMessage();
        }
    }
```

```
        function find( $table = null, $id = null ) {

             $database = open_database();
             $found = null;

            try {
                if ($id) {
                    $sql = "SELECT * FROM " . $table . " WHERE id = " . $id;
                    $result = $database->query($sql);

                    if ($result->num_rows > 0) {
                        $found = $result->fetch_assoc();
                    }

                } else
                        {
                             $sql = "SELECT * FROM " . $table;
                             $result = $database->query($sql);

                             if ($result->num_rows > 0) {
                                 $found = $result->fetch_all(MYSQLI_ASSOC);
                             }
                         }
                } catch (Exception $e) {
                        $_SESSION['message'] = $e->GetMessage();
                        $_SESSION['type'] = 'danger';
                }

                close_database($database);
                return $found;
            }

    ?>

<?php
        /**
        *  Pesquisa Todos os Registros de uma Tabela
        */

        function find_all( $table ) {
            return find($table);
        }
 ?>

<?php

        function save($table = null, $data = null) {
            $database = open_database();
            $columns = null;
            $values = null;
            //print_r($data);
```

```
        foreach ($data as $key => $value) {
            $columns .= trim($key, "'") . ",";
            $values .= "'$value',";
        }

       // remove a ultima virgula
            $columns = rtrim($columns, ',');
            $values = rtrim($values, ',');

            $sql = "INSERT INTO " . $table . "($columns)" . " VALUES " . "($values);";
            try {
                $database->query($sql);
                $_SESSION['message'] = 'Registro cadastrado com sucesso.';
                $_SESSION['type'] = 'success';

            } catch (Exception $e) {

                $_SESSION['message'] = 'Nao foi possivel realizar a operacao.';
                $_SESSION['type'] = 'danger';
            }
            close_database($database);
        }
      ?>

<?php

function update($table = null, $id = 0, $data = null) {
  $database = open_database();
  $items = null;
  foreach ($data as $key => $value) {
    $items .= trim($key, "'") . "='$value',";
  }
  // remove a ultima virgula
  $items = rtrim($items, ',');
  $sql  = "UPDATE " . $table;
  $sql .= " SET $items";
  $sql .= " WHERE id=" . $id . ";";
  try {
    $database->query($sql);
    $_SESSION['message'] = 'Registro atualizado com sucesso.';
    $_SESSION['type'] = 'success';
  } catch (Exception $e) {
    $_SESSION['message'] = 'Nao foi possivel realizar a operacao.';
    $_SESSION['type'] = 'danger';
  }
  close_database($database);
}
?>

<?php

function remove( $table = null, $id = null ) {
```

```
    $database = open_database();

    try {
        if ($id) {
            $sql = "DELETE FROM " . $table . " WHERE id = " . $id;
            $result = $database->query($sql);
         if ($result = $database->query($sql)) {
             $_SESSION['message'] = "Registro Removido com Sucesso.";
             $_SESSION['type'] = 'success';
         }
      }
   }    catch (Exception $e) {
          $_SESSION['message'] = $e->GetMessage();
          $_SESSION['type'] = 'danger';
   }
   close_database($database);
}
?>
```

Este será um arquivo de funções do banco de dados. Tudo que for relativo ao BD estará nele.

Vamos entender esses códigos…

Primeiramente, na linha 3, configuramos o MySQL para avisar sobre erros graves, usando a função mysqli_report(). Depois, temos duas funções.

A primeira função – open_database() – abre a conexão com a base de dados, e retorna a conexão realizada, se der tudo certo. Se houver algum erro, a função dispara uma exceção, que pode ser exibida ao usuário.

Já a segunda função – close_database($conn) – fecha a conexão que for passada. Se houver qualquer erro, a função dispara uma exceção, também.

Passo 4: Teste a Conexão

Agora sim, vamos ver se o banco de dados está conectado ao CRUD.

Crie um arquivo chamado index.php, na pasta principal, e coloque o código a seguir:

```
<?php require_once 'config.php'; ?>
<?php require_once DBAPI; ?>

<?php
            $db = open_database();

            if ($db) {
                 echo '<h1>Banco de Dados Conectado!</h1>';
            } else {
                 echo '<h1>ERRO: Não foi possível Conectar!</h1>';
            }
?>
```