

Classe FormacaoAcad

Nesta aula vamos continuar com o desenvolvimento do nosso projeto completo no padrão MVC com PHP Orientado a Objetos.

Agora vamos modelar a classe FormacaoAcad que será responsável por gerenciar os dados da formação acadêmica do usuário. Crie o arquivo PHP com o nome "FormacaoAcad", novamente dentro da pasta Model. Não se esquecendo de colocar os delimitadores php "<?php ... ?>". O desenvolvimento desta classe será baseado no diagrama a seguir:

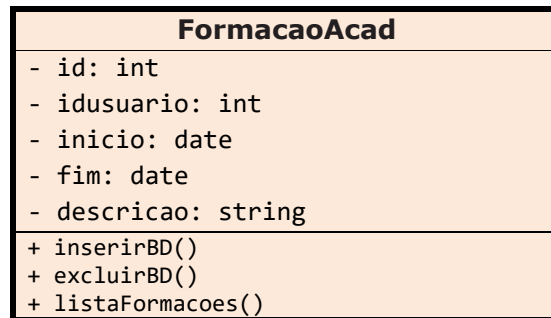


Diagrama da Classe FormacaoAcad

Logo após os delimitadores, devemos criar a classe e seus atributos, sabendo que:

- id é o código único de cada registro de formação acadêmica;
- idusuario é o código do usuário a quem pertence essa formação;
- inicio é a data do início da formação;
- fim é a data do fim da formação;
- descrição é a descrição da formação acadêmica.

Como resultado, devemos obter o seguinte código:

```
class FormacaoAcad
{
    private $id;
    private $idusuario;
    private $inicio;
    private $fim;
    private $descricao;
}
```

Assim como na classe Usuário, o próximo passo é criar todos os métodos *getters* e *setters* de cada um desses atributos. Eles darão acesso aos atributos privados desta classe. Seguindo os padrões de projeto codifica-se dessa forma:

```
//ID
public function setID($id)
{
    $this->id = $id;
}
public function getID()
{
    return $this->id;
}
//idusuario
public function setIdUsuario($idusuario)
{
    $this->idusuario = $idusuario;
}
public function getIdUsuario()
{
    return $this->idusuario;
}
```

```
//inicio
public function setInicio($inicio)
{
    $this->inicio = $inicio;
}
public function getInicio()
{
    return $this->inicio;
}
//fim
public function setFim($fim)
{
    $this->fim = $fim;
}
public function getFim()
{
    return $this->fim;
}
//Descrição
public function setDescription($descricao)
{
    $this->descricao = $descricao;
}
public function getDescription()
{
    return $this->descricao;
}
```

Agora vamos partir para a codificação dos métodos específicos, que são três: *inserirBD*, *excluirBD* e *listaFormacoes*. Todos estão relacionados a operações com o Banco de Dados. Para a criação desses métodos, siga o exemplo da codificação dos métodos da classe Usuário. O primeiro método, "*inserirBD*", é exatamente como o desenvolvido para a classe Usuário, apenas mudando para a tabela (INSERT INTO). Para codificar, basta seguir o mesmo padrão.

```
public function inserirBD()
{
    require_once 'ConexaoBD.php';
    $con = new ConexaoBD();
    $conn = $con->conectar();
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    $sql = "INSERT INTO formacaoAcad (idusuario, inicio, fim, descricao)
VALUES ('".$this->idusuario."', '".$this->inicio."', '".$this->fim."', '".
    $this->descricao."')";
    if ($conn->query($sql) === true) {
        $this->id = mysqli_insert_id($conn);
        $conn->close();
        return true;
    }
    else {
        $conn->close();
        return false;
    }
}
```

Vamos para o próximo método: *excluirBD*.

```
public function excluirBD($id)
{
    require_once 'ConexaoBD.php';
    $con = new ConexaoBD();
    $conn = $con->conectar();
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    $sql = "DELETE FROM formacaoAcad WHERE idformacaoAcademica = '".$id.'" ";
    if ($conn->query($sql) === true) {
        $conn->close();
        return true;
    }
    else {
        $conn->close();
        return false;
    }
}
```

Este método segue o mesmo padrão:

- Inclusão da classe ConexaoBD;
- Instância do objeto da Classe ConexaoBD;
- Conexão ao Banco de Dados, com verificação do sucesso ou não;
- Confeção da sentença SQL;
- Execução da sentença com verificação do sucesso ou não.

A diferença é que precisa de um id (parâmetro) para realizar a exclusão do registro correto. Então, em caso positivo, será excluído o registro do Banco de Dados e em seguida será fechada a conexão, retornando a TRUE. Caso negativo, fecha-se a conexão e retorna a FALSE.

Por fim, codifique o método *listaFormacoes*:

```
public function listaFormacoes($idusuario)
{
    require_once 'ConexaoBD.php';
    $con = new ConexaoBD();
    $conn = $con->conectar();
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    $sql = "SELECT * FROM formacaoAcademica WHERE idusuario = '".$idusuario.'" ";
    $re = $conn->query($sql);
    $conn->close();
    return $re;
}
```

Este método segue inicialmente o mesmo Padrão:

- Inclusão da classe ConexaoBD;
- Instância do objeto da Classe ConexaoBD;
- Conexão ao Banco de Dados, com verificação do sucesso ou não;
- Confeção da sentença SQL.

Porém, além de receber o id do usuário por meio de passagem de parâmetro, esse método retorna um ou mais registros como resultado da consulta do Banco de Dados.

```
$re = $conn->query($sql);
$conn->close();
return $re;
```

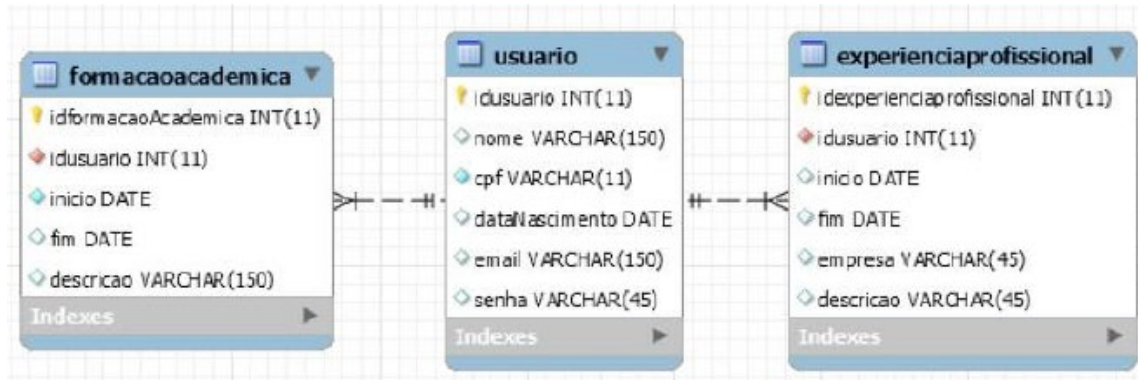


Diagrama Lógico de Banco de dados

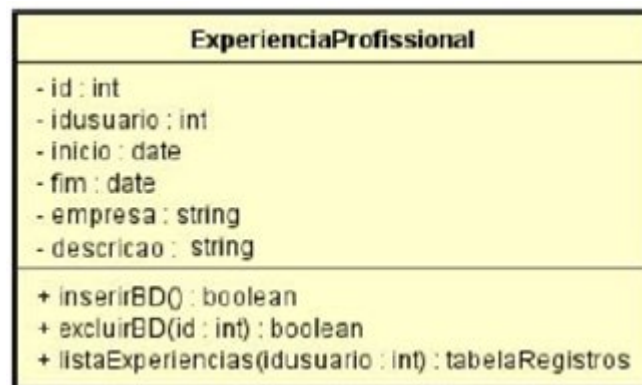


Diagrama de classe Experiência Profissional

Script SQL para criação da tabela *ExperienciaProfissional*:

```

CREATE TABLE `projeto_final`.`experienciaprofissional` (
  `idexperienciaprofissional` INT NOT NULL AUTO_INCREMENT,
  `idusuario` INT NOT NULL,
  `inicio` DATE NULL,
  `fim` DATE NULL,
  `empresa` VARCHAR(45) NULL,
  `descricao` VARCHAR(45) NULL,
  PRIMARY KEY (`idexperienciaprofissional`),
  INDEX `idUser_idx` (`idusuario` ASC),
  CONSTRAINT `idUser`
  FOREIGN KEY (`idusuario`)
  REFERENCES `projeto_final`.`usuario` (`idusuario`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION);
  
```

Classe ExperiênciaProfissional

Com o nome "*ExperiênciaProfissional*", novamente dentro da pasta Model, crie o arquivo PHP. Não se esqueça de colocar os delimitadores php "<?php e ?>".

O desenvolvimento dessa classe deve ser baseado no diagrama de Classe *ExperienciaProfissional*. A partir dos dados desse diagrama crie a classe e seus atributos com seus respectivos métodos *getters* e *setters*:

```

class ExperienciaProfissional
{
    private $id;
    private $idusuario;
    private $inicio;
    private $fim;
    private $empresa;
    private $descricao;
  }
  
```

```
//ID
public function setID($id)
{
    $this->id = $id;
}
public function getID()
{
    return $this->id;
}
//idusuario
public function setIdUsuario($idusuario)
{
    $this->idusuario = $idusuario;
}
public function getIdUsuario()
{
    return $this->idusuario;
}
//inicio
public function setInicio($inicio)
{
    $this->inicio = $inicio;
}
public function getInicio()
{
    return $this->inicio;
}
//fim
public function setFim($fim)
{
    $this->fim = $fim;
}
public function getFim()
{
    return $this->fim;
}
//Empresa
public function setEmpresa($empresa)
{
    $this->empresa = $empresa;
}
public function getEmpresa()
{
    return $this->empresa;
}
//Descrição
public function setDescricao($descricao)
{
    $this->descricao = $descricao;
}
public function getDescricao()
{
    return $this->descricao;
}
}
```

Veja a construção do Método *inserirBD*:

```
public function inserirBD()
{
    require_once 'ConexaoBD.php';
    $con = new ConexaoBD();
    $conn = $con->conectar();
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    $sql = "INSERT INTO experienciaprofissional (idusuario, inicio, fim, empresa,
        descricao) VALUES ('".$this->idusuario."', '".$this->inicio."', '".$this->fim."', '".$this->empresa."', '".$this->descricao."')";
    if ($conn->query($sql) === true) {
        $this->id = mysqli_insert_id($conn);
        $conn->close();
        return true;
    }
    else {
        $conn->close();
        return false;
    }
}
```

Veja a construção do Método *excluirBD*:

```
public function excluirBD($id)
{
    require_once 'ConexaoBD.php';
    $con = new ConexaoBD();
    $conn = $con->conectar();
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    $sql = "DELETE FROM experienciaprofissional WHERE idexperienciaprofissional = '".$id."'";
    if ($conn->query($sql) === true) {
        $conn->close();
        return true;
    }
    else {
        $conn->close();
        return false;
    }
}
```

Veja a construção do Método *listarExperiências*:

```
public function listaExperiencias($idusuario)
{
    require_once 'ConexaoBD.php';
    $con = new ConexaoBD();
    $conn = $con->conectar();
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    $sql = "SELECT * FROM experienciaProfissional WHERE idusuario = '".$idusuario."'";
    $re = $conn->query($sql);
    $conn->close();
    return $re;
}
```

Na próxima aula daremos sequência ao desenvolvimento deste projeto.
Até lá! :)