

Sistema de posts para WordPress

O modelo de dados do sistema consta em duas entidades: usuário e post. Como pode-se ver abaixo um usuário possui quatro ações disponíveis:

- Criar post;
- Editar post;
- Excluir post;
- Publicar post (enviar ao WordPress).

Modelo de dados

Um usuário pode ter nenhum ou vários posts.
Já um post pertence a um usuário apenas.

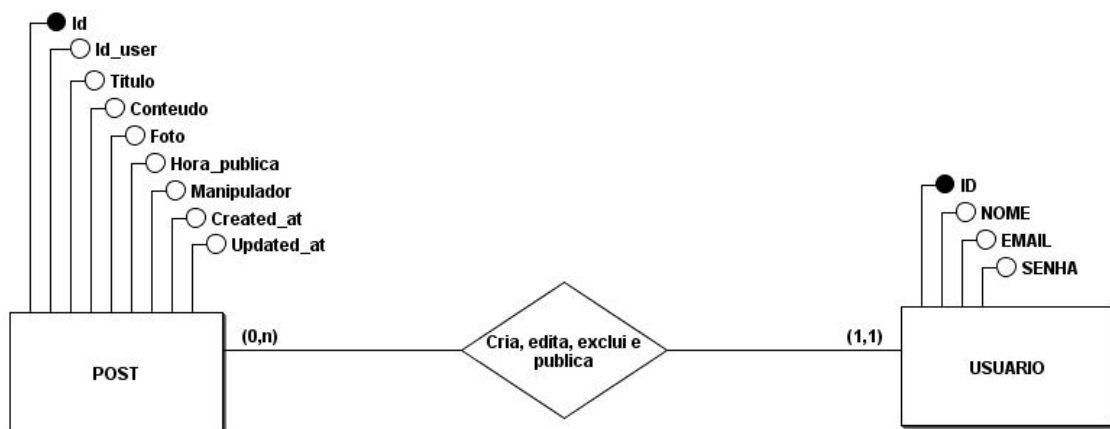


Figura 1. Modelo conceitual de dados do sistema

A entidade usuário tem os seguintes atributos:

- Id (chave primária);
- Nome;
- E-mail;
- Senha.

A entidade Post tem os seguintes atributos:

- Id;
- Id_user (chave estrangeira que corresponde ao id do usuário);
- Título;
- Conteúdo;
- Foto;
- Hora_publica;
- Manipulador;
- Created_at;
- Updated_at.

Abaixo, mais detalhadamente, pode-se ver o modelo lógico de dados do sistema:

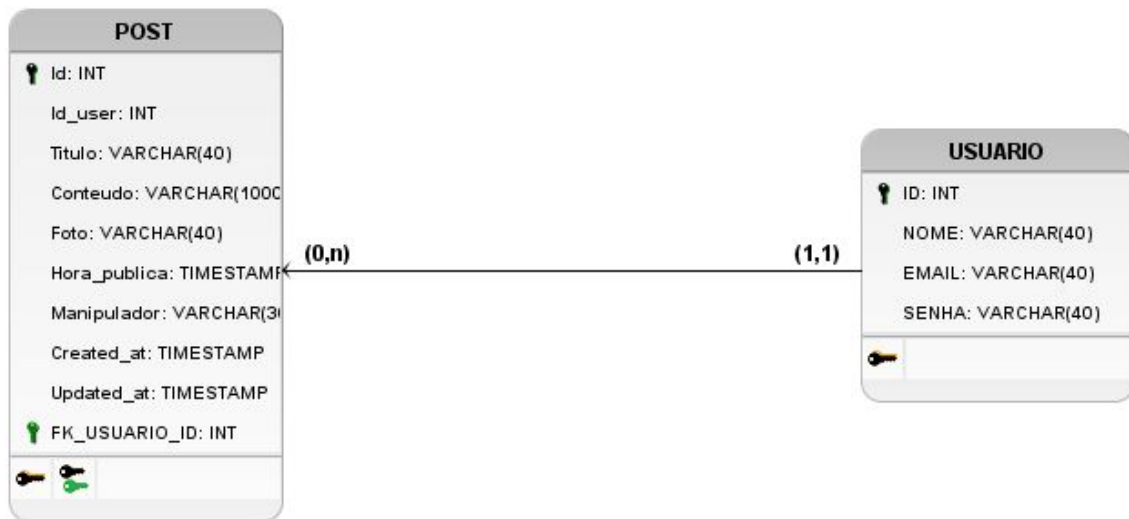


Figura 2. Modelo lógico de dados do sistema

Configuração/ instalação do sistema

Após baixar o projeto, será preciso ter instalado o wordpress na máquina que o utilizará.

Aqui está um link de um tutorial para instalar o WordPress, caso ainda não o tenham instalado:
<https://www.youtube.com/watch?v=YC9RazT2a5c>

Além disso, as alterações a seguir precisam ser feitas:

1. Alterar nome do banco de dados utilizado pelo seu WordPress:
Dentro da pasta do projeto, haverá uma pasta com o nome **wordpress**.
Nesta pasta, *procure por wp-config.php*.
Este arquivo contém as configurações de banco de dados do wordpress.
Procure por `define('DB_NAME', 'wp_pradig');`.
Esta função define o nome do banco de dados utilizado pelo wordpress. Altere somente o segundo parâmetro para o mesmo nome que utilizar em seu WordPress.

Caso esta breve orientação acima não seja o suficiente, neste link também explica como alterar o nome do banco de dados wordpress:

<https://premium.wpmudev.org/blog/change-database-name/>

2. Alterar dados da conexão com o banco de dados:
Na pasta principal do projeto há o arquivo **conexao.php**.
Este arquivo contém os dados de conexão com o banco de dados que estiver utilizando.

Por padrão elas virão como:

```
host = localhost;  
nome do banco = faculdade;  
usuário = 'root';  
senha = ''.
```

Altere para o seu ambiente para que a conexão funcione.

Fluxo de funcionamento do sistema

Cadastro/ Login:

O cadastro de novos usuários é feito com nome de usuário, e-mail e senha como campos de autenticação.

Não é permitido cadastrar dois usuários com emails iguais.

O login é feito informando e-mail e senha.

Post

Após o cadastro ser realizado, o usuário terá o direito de criar novas postagens, ver todas as postagens que ele criou, bem como editá-las, excluí-las, ou ainda, selecionar um post para ser publicado no WordPress.

Na criação do post/ postagem o usuário deverá inserir os seguintes campos:

- Título
- Conteúdo

Estes campos são obrigatórios.

Além disso, o usuário poderá inserir uma foto. O campo Foto da postagem é opcional.

Na página “Todas as postagens” o usuário terá as seguintes ações disponíveis para cada post criado:

- Editar (botão azul);
- Excluir (botão vermelho);
- Publicar (botão verde).

Publicação

Ao selecionar a opção de publicar o post, o sistema vai se conectar ao *post_controller*. O *post_controller* tem diversos (else if) manipulando a variável *\$acao*. A ação correspondente ao botão de publicar é *\$acao == 'publicar'*. Veja o trecho de código abaixo:

```
else if($acao == 'publicar'){
    $post = new Post(); // Instância da classe POST(presente em
post_model.php)
    $hora = date('Y-m-d H:i:s'); // Variável que recupera a data e
hora atual
    $post->__set('id', $_POST['id']); // O id da publicação recuperada
do formulário que vem do front ao selecionar o botão de publicar.
    $post->__set('manipulador', $hora); // Setando a variável que contém
a hora atual no atributo manipulador (presente em post_model.php)
    $conexao = new Conexao();
    $postService = new PostService($conexao, $post);
    $postService->publicar();
    header('location:
http://localhost/posts/wordpress/wp-post/original'); // Após todas as
ações do controller, vai direcionar para o script que faz a publicação
do wordpress
}
```

Como funciona a conexão da aplicação com o script de publicação dos dados?

Dentro da pasta wp-post do WordPress, existem três arquivos fundamentais:

- WordpressController.php;
- WordService.php;
- Wordpost.php.

Na própria aplicação PHP, ao ser pressionado o botão de publicar, a coluna do post chamada (manipulador) recebe a data e hora atuais.

O método recuperar do arquivo WordService manipula essa coluna, pois esse método foi criado para buscar sempre o post mais recente possuindo a data e horário como condições. Veja o código abaixo:

```
public function recuperar()
{
    $query = 'select * from posts ORDER BY manipulador DESC LIMIT 1';
    $stmt = $this->banco->prepare($query);
    $stmt->execute();
    return $stmt->fetchAll(PDO::FETCH_OBJ);
}
```

Dessa forma, O *WordpressController.php* vai buscar esse método *recuperar* que possui o post mais recente. Esse método chamado *recuperar* vai ficar armazenado em uma variável chamada *\$arrayposts*. Veja o trecho de código abaixo

```
$arrayPosts = $wordservice->recuperar();
```

Finalmente temos os dados do post mais recente na variável *\$arrayPost*, e podemos manipular ela no script que faz a publicação. Basta fazer um *require* de *WordpressController.php* no script desejado.

Esse script que faz a publicação no WordPress é o *Original.php*.

Ao fazer o *require* de *Wordpresscontroller.php* nessa página, podemos usar a variável que contém o post mais recente e fazer um *foreach* para popular os dados da postagem que será enviada ao WordPress.

Veja o trecho de código abaixo:

```
foreach ($arrayPosts as $key => $valores) {
    $newPost = array(
        'post_title' => $valores->title,
        'post_content' => $valores->content,
        'post_status' => $postStatus,
        'post_date' => $timeStamp,
        'post_author' => $userID,
        'post_type' => $postType,
        'post_category' => array($categoryID)
    );
}
```

Pronto. Agora o array que será enviado ao WordPress já possui os dados vindos do sistema, e a publicação no wordpress será feita!