3º Bimestre - PW II

por Rafael SP

Cronograma

Aula 1 - 05/08

- Apresentar Projeto Cats
- Apresentar Instrumentos Avaliativos
- Construção de Páginas
- Login
- Cadastre-se

Aula 2 -12/08

- Introduzir conceitos de:
- Sessão
- Cookies
- Construir Páginas:
- Feed
- Revisar Foreach
- Implementar Sessão

Aula 3 - 19/08

- Construir Paginas:
- Post
- Inserir Post
- Atualizar Post

Aula 4 - 26/08

- Termino do Projéto
- Explicação do Projeto avaliativo do Semestre

Aula 5 - 02/09

- Prova

Cronograma

Aula 6 - 09/09

Projeto Avaliativo

Aula 7 -16/09

Projeto Avaliativo

Aula 8 - 23/05

Entrega do Projeto Avaliativo

25/05

Último dia de Entrega de Menções

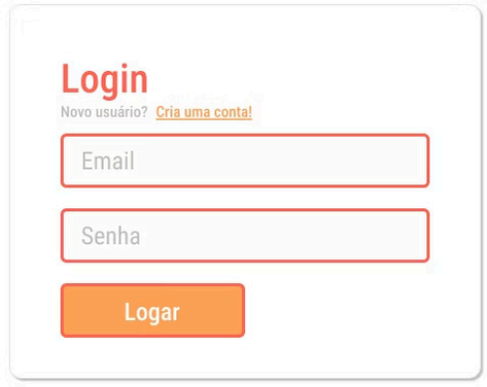
Projeto Guiado Cats

Objetivo

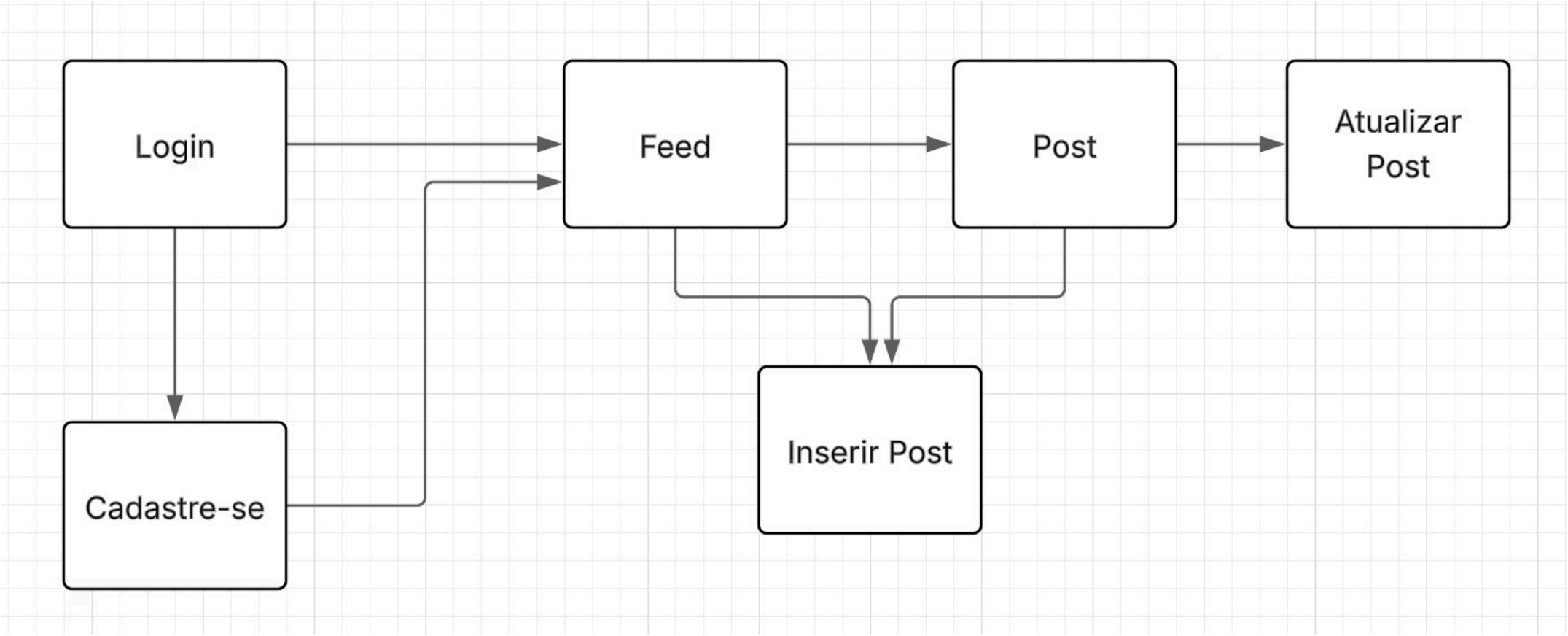
> Trabalhar os conceitos de Sessões, Cookies, Validação de Formulários e estruturas de controle a partir de um projeto guiado.

Cats

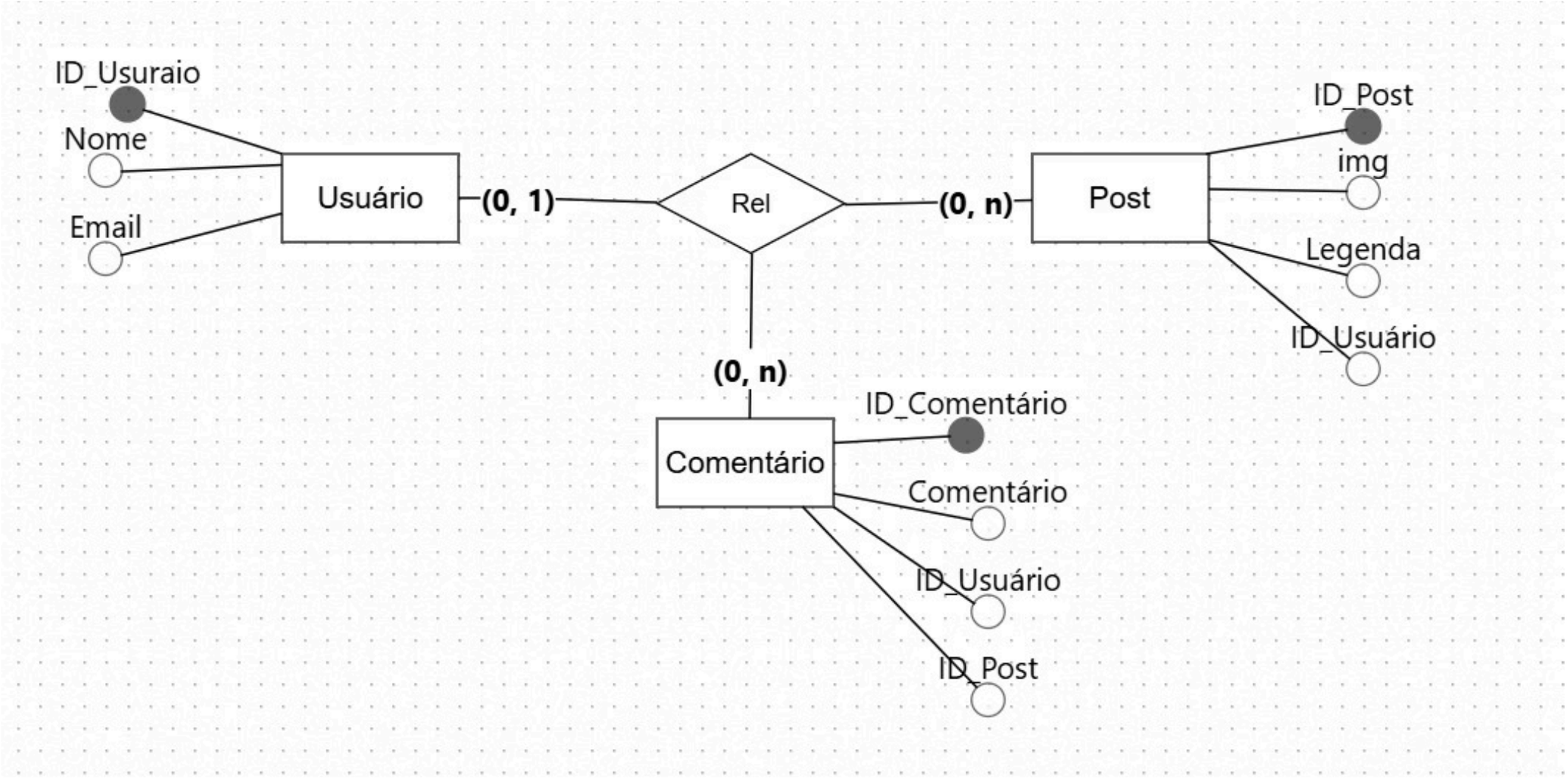
Cats é uma rede social para gatos. Onde os gatos publicam, curtem e compartilham.



Paginas



Banco de Dados



Arquivos que criaremos

```
/projeto/
  │
  ├─── db.php             -> Conexão com o banco
  ├─── login.php           -> Tela de login
  ├─── cadastrar.php       -> Tela de cadastro
  ├─── processa_login.php  -> Processa login
  └─── processa_cadastro.php -> Processa cadastro
```

📦 SQL para criar a tabela no MySQL

Execute isso no phpMyAdmin (no banco `meu_banco`):

```
create database catsdb;
use catsdb;

CREATE TABLE usuario (
    id INT AUTO_INCREMENT PRIMARY KEY,
    email VARCHAR(255) NOT NULL UNIQUE,
    senha VARCHAR(255) NOT NULL
);
```

🔌 db.php — Conexão com o MySQL

```
<?php
$host = 'localhost';
$user = 'root';
$pass = ''; // Padrão do XAMPP
$dbname = 'catsdb';

$conn = new mysqli($host, $user, $pass, $dbname);

if ($conn->connect_error) {
    die("Erro de conexão: " . $conn->connect_error);
}else{
    echo "Deu bom!";
}
?>
```

> Certifique-se de que você criou um banco chamado `cats_DB` no phpMyAdmin.

📥 login.php — Tela de Login

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Login</title>
</head>
<body>
    <h2>Login</h2>
    <form action="processa_login.php" method="POST">
        <label>Email:</label>
        <input type="email" name="email" required><br><br>

        <label>Senha:</label>
        <input type="password" name="senha" required><br><br>

        <button type="submit">Logar</button>
    </form>

    <p><a href="cadastrar.php">Ainda não tem conta? Cadastre-se</a></p>
</body>
</html>
```

📝 `cadastrar.php` — Tela de Cadastro

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Cadastro</title>
</head>
<body>
    <h2>Cadastre-se</h2>
    <form action="processa_cadastro.php" method="POST">
        <label>Email:</label>
        <input type="email" name="email" required><br><br>

        <label>Senha:</label>
        <input type="password" name="senha" required><br><br>

        <label>Confirme a senha:</label>
        <input type="password" name="confirmar_senha" required><br><br>

        <button type="submit">Cadastrar</button>
    </form>

    <p><a href="login.php">Já tem conta? Faça login</a></p>
</body>
</html>
```

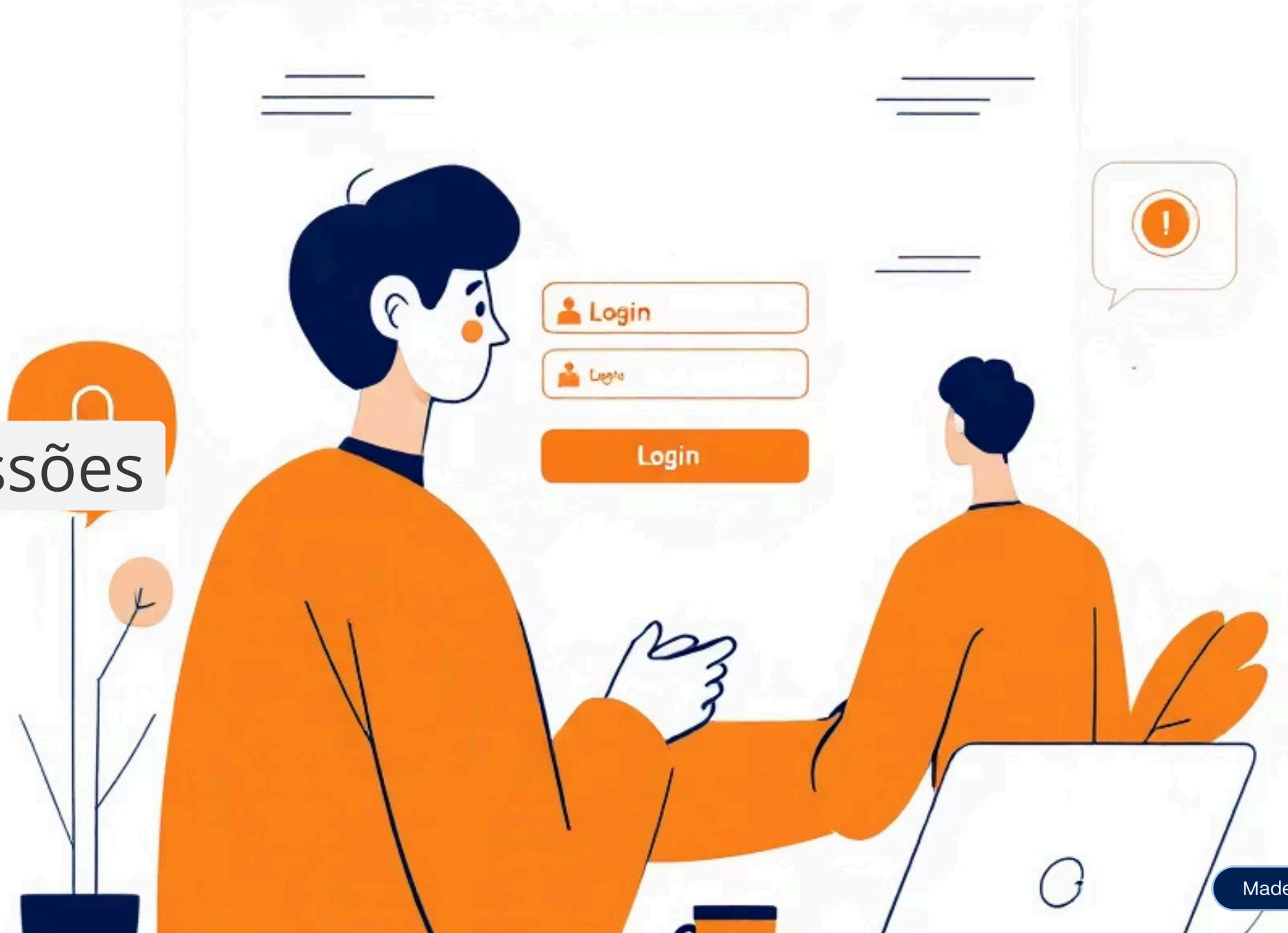
✅ processa_cadastro.php — Processa Cadastro

```
<?php
include 'db.php';

$email = $_POST['email'];
$senha = $_POST['senha'];
$confirmar = $_POST['confirmar_senha'];

if ($senha !== $confirmar) {
    die("As senhas não conferem.");
    // die = exit
    // Ela encerra a execução do código.
}

$senha_hash = password_hash($senha, PASSWORD_DEFAULT);

$sql = "INSERT INTO usuario (email, senha) VALUES (?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("ss", $email, $senha_hash);

if ($stmt->execute()) {
    echo "Cadastro realizado com sucesso. <a href='login.php'>Fazer login</a>";
} else {
    echo "Erro ao cadastrar: " . $stmt->error;
}

$conn->close();
?>
```

🔐 `processa_login.php` — Processa Login

```
<?php
include 'db.php';

$email = $_POST['email'];
$senha = $_POST['senha'];

$sql = "SELECT * FROM usuario WHERE email = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("s", $email);
$stmt->execute();
$result = $stmt->get_result();

if ($usuario = $result->fetch_assoc()) {
    if (password_verify($senha, $usuario['senha'])) {
        echo "Login bem-sucedido. Bem-vindo, " . htmlspecialchars($usuario['email']) . "!";
    } else {
        echo "Senha incorreta.";
    }
} else {
    echo "Usuário não encontrado.";
}

$conn->close();
?>
```

Sessões



Nosso Banco até agora

```
create database catsdb;
use catsdb;

CREATE TABLE usuario (
    id INT AUTO_INCREMENT PRIMARY KEY,
    email VARCHAR(255) NOT NULL UNIQUE,
    senha VARCHAR(255) NOT NULL
);

CREATE TABLE posts (
    id INT AUTO_INCREMENT PRIMARY KEY,
    usuario_id INT NOT NULL,
    imagem VARCHAR(255) NOT NULL,
    legenda TEXT NOT NULL,
    data_post DATETIME NOT NULL,
    FOREIGN KEY (usuario_id) REFERENCES usuario(id)
);
```

1️⃣ O que é uma sessão?

Uma **sessão** é uma forma do servidor “lembrar” informações sobre o usuário **entre diferentes páginas** do site.

- **Sem sessão** → cada vez que você abre uma página, o servidor não sabe quem você é.
- **Com sessão** → o servidor guarda informações (como `id` do usuário) e consegue reconhecê-lo de página em página.

> Pense na sessão como um armário com seu nome no servidor: você coloca informações lá (ex.: seu login) e pode pegar depois, enquanto a sessão estiver ativa.

2️⃣ Como o PHP faz isso?

O PHP usa um **identificador de sessão** (um código único), normalmente salvo em um **cookie** no navegador chamado `PHPSESSID`.

1. Quando você executa `session_start()`, o PHP:
   - Gera um identificador de sessão (se for a primeira vez);
   - Ou recupera o identificador existente (se o usuário já tiver uma sessão ativa).
2. Esse identificador é usado para acessar um **arquivo no servidor** onde ficam os dados da sessão.

3️⃣ Criando e lendo variáveis de sessão

As variáveis de sessão ficam no array **superglobal** `$_SESSION`.

Exemplo:

```
session_start();

// Criar variável de sessão
$_SESSION['usuario_id'] = 123;

// Acessar variável de sessão
echo $_SESSION['usuario_id']; // 123
```

4️⃣ Quando a sessão começa e termina

- **Começa**: quando você chama `session_start()` e define dados em `$_SESSION`.
- **Termina**:
  - Quando você **fecha o navegador** (se o cookie for temporário);
  - Quando você executa `session_destroy()`;
  - Ou quando o servidor expira a sessão (configuração `php.ini` → `session.gc_maxlifetime`).

5️⃣ Ciclo de uso típico em um sistema com login

1. Usuário envia email/senha → sistema valida no banco.
2. Se válido, cria variáveis de sessão:

```
$_SESSION['usuario_id'] = $dados['id'];
$_SESSION['nome'] = $dados['nome'];
```

1. Em cada página restrita:

```
session_start();
if (!isset($_SESSION['usuario_id'])) {
    header("Location: login.php");
    exit;
}
```

1. Para logout:

```
session_start();
session_unset();    // Remove variáveis da sessão
session_destroy();  // Destrói a sessão
```

6️⃣ Vantagens e cuidados

Vantagens:

- Fácil de implementar;
- Dados não ficam no navegador, mas no servidor (mais seguro que cookies para dados sensíveis);
- Funciona automaticamente entre páginas.

Cuidados:

- Sempre chamar `session_start()` antes de qualquer saída HTML;
- Não armazenar dados muito grandes ou arquivos na sessão;
- Regenerar o ID de sessão após login (`session_regenerate_id(true)`) para evitar *session fixation*;
- Usar HTTPS para proteger o cookie da sessão.

Inserindo uma sessão no processa_login.php

```
<?php
session_start();
include 'db.php';
$email = $_POST['email'];
$senha = $_POST['senha'];

$sql = "SELECT * FROM usuario WHERE email = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("s", $email);
$stmt->execute();
$result = $stmt->get_result();

if ($usuario = $result->fetch_assoc()) {
    if (password_verify($senha, $usuario['senha'])) {
        // Cria a sessão com o ID do usuário
        $_SESSION['usuario_id'] = $usuario['id'];

        // Redireciona para o feed
        header("Location: novo_post.php");
        exit;
    } else {
        echo "Senha incorreta.";
    }
} else {
    echo "Usuário não encontrado.";
}

$conn->close();
?>
```

2. Criando `novo_post.php`

```
<?php
session_start();

// Verifica se está logado
if (!isset($_SESSION['usuario_id'])) {
    header("Location: login.php");
    exit;
}
?>
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Novo Post</title>
</head>
<body>

    <form action="salvar_post.php" method="POST" enctype="multipart/form-data">
        <h2>Criar novo post</h2>
        <label>Imagem:</label>
        <input type="file" name="imagem" accept="image/*" required><br><br>

        <label>Legenda:</label>
        <textarea name="legenda" rows="4" cols="40" required></textarea><br><br>

        <button type="submit">Postar</button>
    </form>
</body>
</html>
```

salvar_post.php

```
<?php
session_start();
require 'db.php'; // conexão ao banco

// Verifica se o usuário está logado
if (!isset($_SESSION['usuario_id'])) {
    header("Location: login.php");
    exit;
}

$usuario_id = $_SESSION['usuario_id'];
$legenda = $_POST['legenda'];
$data_post = date('Y-m-d H:i:s'); // data e hora atuais

// Diretório para salvar imagens
$diretorio = "uploads/";
if (!is_dir($diretorio)) {
    mkdir($diretorio, 0777, true);
}

// Verifica se o upload foi feito
if (isset($_FILES['imagem']) && $_FILES['imagem']['error'] == 0) {
    $nome_arquivo = uniqid() . "_" . basename($_FILES['imagem']['name']);
    $caminho_arquivo = $diretorio . $nome_arquivo;

    // Move o arquivo para o diretório de uploads
    if (move_uploaded_file($_FILES['imagem']['tmp_name'], $caminho_arquivo)) {
        // Salva no banco
        $sql = "INSERT INTO posts (usuario_id, imagem, legenda, data_post) VALUES (?, ?, ?, ?)";
        $stmt = $conn->prepare($sql);
        $stmt->bind_param("isss", $usuario_id, $caminho_arquivo, $legenda, $data_post);

        if ($stmt->execute()) {
            echo "Post publicado com sucesso! <a href='feed.php'>Voltar ao feed</a>";
        } else {
            echo "Erro ao publicar post: " . $stmt->error;
        }
    } else {
        echo "Erro ao salvar imagem!";
    }
} else {
    echo "Nenhuma imagem enviada!";
}

$conn->close();
?>
```

Explicação Linha a Linha de `salvar_post.php`

▼ Linha a Linha

1. `<?php` Abre a tag PHP — tudo que vem depois será interpretado como código PHP.
2. `session_start();` Inicia (ou retoma) a sessão do PHP. Precisa ser chamado **antes** de qualquer saída HTML. É necessário para acessar `$_SESSION` (onde guardamos o `usuario_id`).
3. `require 'conexao.php'; // conexão ao banco` Inclui o arquivo `conexao.php` que deve criar a conexão com o banco (normalmente definindo `$conn` como objeto `mysqli` ou `PDO`). Se o arquivo não existir, o script para com erro.
4. `// Verifica se o usuário está logado` Comentário explicativo — não executa nada, só documenta.
5. `if (!isset($_SESSION['usuario_id'])) {` Verifica se a variável de sessão `usuario_id` existe; se não existir, o usuário não está autenticado.
6. `header("Location: login.php");` Se não estiver logado, redireciona o navegador para `login.php`.
7. `exit;` Interrompe a execução do script imediatamente (importante após `header("Location: ...")`).
8. `}` Fecha o `if`.
9. `$usuario_id = $_SESSION['usuario_id'];` Recupera o `id` do usuário logado da sessão e guarda em `$usuario_id` para usar depois (associar o post ao usuário).
10. `$legenda = $_POST['legenda'];` Pega o valor do campo `legenda` enviado pelo formulário via POST. **Observação:** aqui pegamos direto; para evitar XSS depois, é melhor escapar ao exibir (ex.: `htmlspecialchars`) — preparado evita SQL injection, mas não XSS.
11. `$data_post = date('Y-m-d H:i:s'); // data e hora atuais` Gera a data e hora atual no formato `YYYY-MM-DD HH:MM:SS`, padrão para campos `DATETIME` no MySQL.
12. `// Diretório para salvar imagens` Outro comentário.
13. `$diretorio = "uploads/";` Define a pasta onde as imagens serão salvas (relativa ao diretório do script).
14. `if (!is_dir($diretorio)) {` Verifica se o diretório existe.
15. `mkdir($diretorio, 0777, true);` Se não existir, cria o diretório. `0777` dá permissão total (leitura/escrita/execução) — **recomendo** usar `0755` em produção por segurança. `true` permite criar diretórios recursivamente.
16. `}` Fecha o `if` do diretório.
17. `// Verifica se o upload foi feito` Comentário.
18. `if (isset($_FILES['imagem']) && $_FILES['imagem']['error'] == 0) {` Verifica se o arquivo `imagem` foi enviado e se não houve erro no upload (erro 0 = OK).
19. `$nome_arquivo = uniqid() . "_" . basename($_FILES['imagem']['name']);` Cria um nome único para o arquivo usando `uniqid()` concatenado com o nome original (via `basename` para remover possíveis diretórios). Isso reduz risco de sobrescrever arquivos existentes. Ainda é bom sanitizar/remover caracteres problemáticos.
20. `$caminho_arquivo = $diretorio . $nome_arquivo;` Monta o caminho completo onde a imagem será salva (ex.: `uploads/5f2a3_arquivo.jpg`).
21. `// Move o arquivo para o diretório de uploads` Comentário.
22. `if (move_uploaded_file($_FILES['imagem']['tmp_name'], $caminho_arquivo)) {` Move o arquivo do local temporário (`tmp_name`) para o destino final. `move_uploaded_file` é a forma segura para tratar uploads.
23. `// Salva no banco` Comentário.
24. `$sql = "INSERT INTO posts (usuario_id, imagem, legenda, data_post) VALUES (?, ?, ?, ?)";` Query SQL com placeholders `?` para evitar injeção de SQL (prepared statement).
25. `$stmt = $conn->prepare($sql);` Prepara a query no banco (retorna um statement).
26. `$stmt->bind_param("isss", $usuario_id, $caminho_arquivo, $legenda, $data_post);` Faz o bind dos parâmetros ao statement. `"isss"` indica os tipos: `i` = integer para `$usuario_id`, e `s` = string para os outros três (caminho, legenda, data).
27. `if ($stmt->execute()) {` Executa o statement; se retornar `true`, a inserção foi bem-sucedida.
28. `echo "Post publicado com sucesso! <a href='feed.php'>Voltar ao feed</a>";` Mensagem de sucesso ao usuário com link para o feed (nota: é better redirecionar ao invés de apenas echo).
29. `} else {` Caso haja erro na execução do statement...
30. `echo "Erro ao publicar post: " . $stmt->error;` Exibe a mensagem de erro retornada pelo statement (útil em desenvolvimento; retire/registre no log em produção).
31. `}` Fecha o `if` do execute.
32. `} else {` Se `move_uploaded_file` retornou falso...
33. `echo "Erro ao salvar imagem!";` Mensagem de erro ao salvar o arquivo no servidor.
34. `}` Fecha o `if` que checa `move_uploaded_file`.
35. `} else {` Fecha o `if` que checa `$_FILES['imagem']` — este é o `else` (caso não tenha arquivo enviado ou erro no upload).
36. `echo "Nenhuma imagem enviada!";` Informa que não houve upload.
37. `}` Fecha o `if` principal de upload.
38. `$conn->close();` Encerra a conexão com o banco (bom liberar recursos).
39. `?>` Fecha a tag PHP (opcional no fim de arquivos PHP que só contenham PHP; pode ser omitida para evitar espaços em branco indesejados).

Feed.php

```
<?php
session_start();
require 'db.php';

// Verifica se está logado
if (!isset($_SESSION['usuario_id'])) {
    header("Location: login.php");
    exit;
}

// Consulta que pega posts junto com dados do usuário
$sql = "
    SELECT p.imagem, p.legenda, p.data_post, u.email
    FROM posts p
    JOIN usuario u ON p.usuario_id = u.id
    ORDER BY p.data_post DESC
";
$result = $conn->query($sql);

// Se não houver posts, $result será uma lista vazia
$posts = [];
if ($result && $result->num_rows > 0) {
    $posts = $result->fetch_all(MYSQLI_ASSOC);
}

$conn->close();
?>
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Feed</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background: #fafafa;
            margin: 0;
            padding: 20px;
        }
        .post {
            background: white;
            border: 1px solid #ddd;
            margin-bottom: 20px;
            padding: 15px;
            border-radius: 8px;
            max-width: 500px;
        }
        .post img {
            max-width: 100%;
            border-radius: 8px;
        }
        .autor {
            color: #555;
            font-size: 14px;
            margin-bottom: 8px;
        }
        .data {
            color: #888;
            font-size: 12px;
            margin-top: 5px;
        }
        .legenda {
            margin-top: 10px;
            font-size: 15px;
        }
    </style>
</head>
<body>

<h1>Feed de Posts</h1>
<a href="novo_post.php">
 Novo Post</a>
<hr>

<?php if (empty($posts)): ?>
    <p>Nenhum post encontrado.</p>
<?php else: ?>
    <?php foreach ($posts as $post): ?>
        <div class="post">
            <div class="autor">
                Publicado por: <?= htmlspecialchars($post['email'], ENT_QUOTES, 'UTF-8') ?>
            </div>
            <img src="<?= htmlspecialchars($post['imagem'], ENT_QUOTES, 'UTF-8') ?>" alt="Imagem do post">
            <div class="legenda">
                <?= nl2br(htmlspecialchars($post['legenda'], ENT_QUOTES, 'UTF-8')) ?>
            </div>
            <div class="data">
                <?= date('d/m/Y H:i', strtotime($post['data_post'])) ?>
            </div>
        </div>
    <?php endforeach; ?>
<?php endif; ?>

</body>
</html>
```