

chatUp
A rede social do momento!

Afinal o quê é o chatUp?

No contexto do Projeto Integrador, desenvolvemos o chatUp, uma rede social moderna inspirada em plataformas populares como o Instagram, com o objetivo de aplicar e integrar conhecimentos adquiridos ao longo do curso.

O chatUp foi pensado como um ambiente digital dinâmico, onde os usuários podem:

- Compartilhar fotos, vídeos ;
- Seguir perfis e interagir com publicações;
- Enviar mensagens diretas.

Este projeto visa não apenas a criação de uma rede social funcional, mas também a prática de conceitos essenciais de programação web, banco de dados, experiência do usuário (UX), segurança e organização de software em equipe.

Estrutura do programa

Estrutura de cadastro o que foi usado?

O sistema de cadastro foi desenvolvido com PHP e MySQL e conta com diversas validações para garantir a integridade e segurança dos dados dos usuários.

O que ele faz?

- Recebe os dados de cadastro via formulário (nome, nome de usuário, e-mail, senha e data de nascimento);
- Realiza validações (como verificar senhas iguais, formato de e-mail e idade válida);
- Protege os dados com medidas como:
- Sanitização (limpeza dos dados para evitar SQL Injection);
- Hash da senha com `password_hash()` (criptografia segura);
- Insere os dados no banco de dados usando prepared statements (mais segurança);
- Exibe mensagens de erro ou sucesso para o usuário.

chatUp

Cadastro de Usuário

Nome Completo:

Nome de Usuário:

E-mail:

Senha:

Confirme a Senha:

Data de Nascimento:

Cadastrar

Sistema de formulários

```
<form method="POST" action="/chatUp/login/cadastro.php">
  <div class="form-group">
    <label for="nome_cadastrado">Nome Completo:</label>
    <input type="text" id="nome_cadastrado" name="nome_cadastrado" required>
  </div>

  <div class="form-group">
    <label for="nome_usuario">Nome de Usuário:</label>
    <input type="text" id="nome_usuario" name="nome_usuario" required>
  </div>

  <div class="form-group">
    <label for="email_cadastro">E-mail:</label>
    <input type="email" id="email_cadastro" name="email_cadastro" required>
  </div>

  <div class="form-group">
    <label for="senha_cadastro">Senha:</label>
    <input type="password" id="senha_cadastro" name="senha_cadastro" required>
  </div>

  <div class="form-group">
    <label for="senha_confirma">Confirme a Senha:</label>
    <input type="password" id="senha_confirma" name="senha_confirma" required>
  </div>

  <div class="form-group">
    <label for="data_nascimento">Data de Nascimento:</label>
    <input type="date" id="data_nascimento" name="data_nascimento" required>
  </div>

  <button type="submit" class="btn-cadastrar">Cadastrar</button>
</form>
</body>
```

Sistema de validações

- Senha com no mínimo 8 caracteres;
- E-mails únicos e válidos;
- Nome de usuário único;
- Data de nascimento válida (não pode ser futura):
- Verificação de senhas coincidentes.

```
return $conn->real_escape_string(trim($item));
}, $rawData);

// Validações
$erros = [];

if ( @var string[] $erros {
    $erros[] = "Nome obrigatório";
} elseif (strlen($dados['nome']) > 100) {
    $erros[] = "Nome máximo 100 caracteres";
}

if (empty($dados['nome_usuario'])) {
    $erros[] = "Nome de usuário obrigatório";
}

if (empty($dados['email'])) {
    $erros[] = "Email obrigatório";
} elseif (!filter_var($dados['email'], FILTER_VALIDATE_EMAIL)) {
    $erros[] = "Formato de email inválido";
}

if (strlen($dados['senha']) < 8) {
    $erros[] = "Senha deve ter no mínimo 8 caracteres";
}

if ($dados['senha'] !== $dados['senha_confirma']) {
    $erros[] = "As senhas não coincidem";
}

if (empty($dados['data_nascimento'])) {
    $erros[] = "Data de nascimento obrigatória";
} elseif (!preg_match('/^\d{4}-\d{2}-\d{2}$/', $dados['data_nascimento'])) {
    $erros[] = "Formato de data inválido (AAAA-MM-DD)";
} else {
    $dataNascimento = DateTime::createFromFormat('Y-m-d', $dados['data_nascimento']);
    $dataAtual = new DateTime();
    if ($dataNascimento >= $dataAtual) {
        $erros[] = "Você não pode colocar essa data de nascimento";
    }
}
```

Segurança nas inserções de dados no banco

- Uso de password_hash para senhas;
- Uso de mysqli_prepare e bind_param para evitar SQL Injection;
- Sistema de sessão para manter as mensagens entre páginas (usando \$_SESSION).

```
try {  
    // Hash da senha  
    $senha_hash = password_hash($dados['senha'], PASSWORD_BCRYPT);  
  
    // Início da transação  
    $conn->begin_transaction();  
  
    $stmt = $conn->prepare("INSERT INTO users (nome, nome_usuario, email, senha, data_nascimento) VALUES (?, ?, ?, ?, ?)");  
    if (!$stmt) {  
        throw new Exception("Preparação falhou: " . $conn->error);  
    }  
  
    $stmt->bind_param(  
        "sssss",  
        $dados['nome'],  
        $dados['nome_usuario'],  
        $dados['email'],  
        $senha_hash,  
        $dados['data_nascimento']  
    );  
  
    if (!$stmt->execute()) {  
        throw new mysqli_sql_exception($stmt->error, $stmt->errno);  
    }  
  
    $conn->commit();  
    $_SESSION['sucesso'] = "Cadastro realizado com sucesso!";  
    header("Location: login.php");  
    exit;  
}
```

Estrutura de login, o quê foi usado?

O sistema de login permite que usuários cadastrados acessem sua conta com segurança, utilizando email e senha. O processo foi implementado com foco em eficiência, validação e proteção contra acessos indevidos.

O que ele faz?

- Recebe os dados digitados (e-mail e senha);
- Verifica se o e-mail existe no banco de dados;
- Compara a senha informada com a senha salva (hash) usando `password_verify`;
- Cria uma sessão segura para manter o usuário logado;
- Exibe mensagens de erro se os dados estiverem incorretos.

The image shows a login interface for a service named 'chatUp'. The top section has a blue background with the 'chatUp' logo in white. Below the logo, on the right, is the text 'Divirta-se no chatUp'. The main form area is white and contains two input fields: 'Email' and 'Senha'. Below these fields is a blue button labeled 'Entrar'. At the bottom of the form, there is a link that says 'Não tem conta? Cadastre-se'.

chatUp

Divirta-se no chatUp

Email

Senha

Entrar

Não tem conta? [Cadastre-se](#)

Sistema de formulários

```
<body>
  <div class="container-login">
    <div class="sidebar">
      <h1 class="logo-text">chatUp</h1>
      <div class="frases">
        <span class="frase ativa">Publique fotos e vídeos</span>
        <span class="frase">Troque mensagens com seus amigos</span>
        <span class="frase">Divirta-se no chatUp</span>
      </div>
    </div>

    <div class="form-area">
      <?php if (isset($_SESSION['erro'])): ?>
        <div class="erro"><?= $_SESSION['erro'] ?></div>
        <?php unset($_SESSION['erro']); ?>
      <?php endif; ?>

      <form action="/chatUp/login/login.php" method="post">
        <div>
          <label for="email_login">Email</label>
          <input type="email" name="email_login" id="email_login" required>
        </div>
        <div>
          <label for="senha_login">Senha</label>
          <input type="password" name="senha_login" id="senha_login" required>
        </div>
        <div>
          <input type="submit" value="Entrar">
        </div>
      </form>

      <p>Não tem conta? <a href="/chatUp/login/cadastro.php">Cadastre-se</a></p>
    </div>
  </div>
```


Sistema de validações

- Uso de password_verify() para comparar senhas de forma segura;
- Uso de sessões (\$_SESSION) para manter o usuário autenticado;
- Em caso de falha, uma mensagem é exibida sem revelar se o erro foi no e-mail ou senha (boa prática de segurança).

```
function validar_campos(string $email, string $senha): void
{
    if (empty($email) || empty($senha)) {
        $_SESSION['erro'] = 'Preencha todos os campos';
        header("Location: /chatUp/login/login.php");
        exit;
    }
}

/**
 * @param mysqli $conn,
 * @param string $email.
 */
function validar_existencia_usuario(mysqli $conn, string $email): ?array
{
    $sql = $conn->prepare("SELECT id, nome_usuario, senha FROM users WHERE email = ?");
    $sql->bind_param("s", $email);
    if (!$sql->execute()) {
        throw new RuntimeException("Erro ao buscar usuário");
    }

    $resultado = $sql->get_result();
    return $resultado->fetch_assoc();
}

/**
 * @param string $senha,
 * @param ?array $usuario
 */
function validar_senha(string $senha, ?array $usuario): void
{
    if (!$usuario || !password_verify($senha, $usuario['senha'])) {
        $_SESSION['erro'] = 'Credenciais inválidas';
        header("Location: /chatUp/login/login.php");
        exit;
    }

    $_SESSION['usuario_id'] = $usuario['id'];
    $_SESSION['nome_usuario'] = $usuario['nome_usuario'];
    header("Location: /chatUp/main.php");
    exit;
}
```

Processar login

- Usuário digita o e-mail e a senha e envia o formulário;
- O back-end (arquivo PHP + função `processar_login`) verifica os dados;
- Se estiver correto, cria uma sessão e redireciona para o feed;
- Se estiver errado, exibe uma mensagem de erro amigável.

```
*/
*/
function processar_login(mysqli $conn)
{
    if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['email_login'], $_POST['senha_login'])) {
        try {
            $email = $_POST['email_login'];
            $senha = $_POST['senha_login'];

            validar_campos($email, $senha);
            $usuario = validar_existencia_usuario($conn, $email);
            validar_senha($senha, $usuario);

        } catch (RuntimeException $e) {
            $_SESSION['erro'] = "Erro: " . $e->getMessage();
            header("Location: /chatUp/login/login.php");
            exit;
        }
    }
}
```

Depois da tempestade vem a calmaria, este é o nosso **Feed**.

A página principal é o coração da rede social chatUp, onde os usuários autenticados visualizam os posts, comentários e interações da rede. Esse sistema foi estruturado para garantir segurança, desempenho e organização no carregamento de conteúdo.

O que ela faz?

- Valida a autenticação do usuário (somente usuários logados têm acesso);
- Carrega os dados do perfil logado;
- Busca todos os posts públicos/visíveis no banco de dados;
- Para cada post, carrega os comentários associados;



Mas e meu Perfil?

A funcionalidade de edição de perfil no chatUp permite que os usuários atualizem seu nome, biografia e foto de perfil (avatar). Essa funcionalidade foi projetada com foco em validação, segurança e experiência do usuário.


O que ela faz?

- Permite ao usuário alterar seu nome, bio e avatar;
- Realiza validações para garantir que os dados sejam válidos;
- Realiza upload da imagem com verificação de tipo e tamanho;
- Atualiza o banco de dados de forma segura com transações.

[Voltar](#)

Editar Perfil

Foto do Perfil:



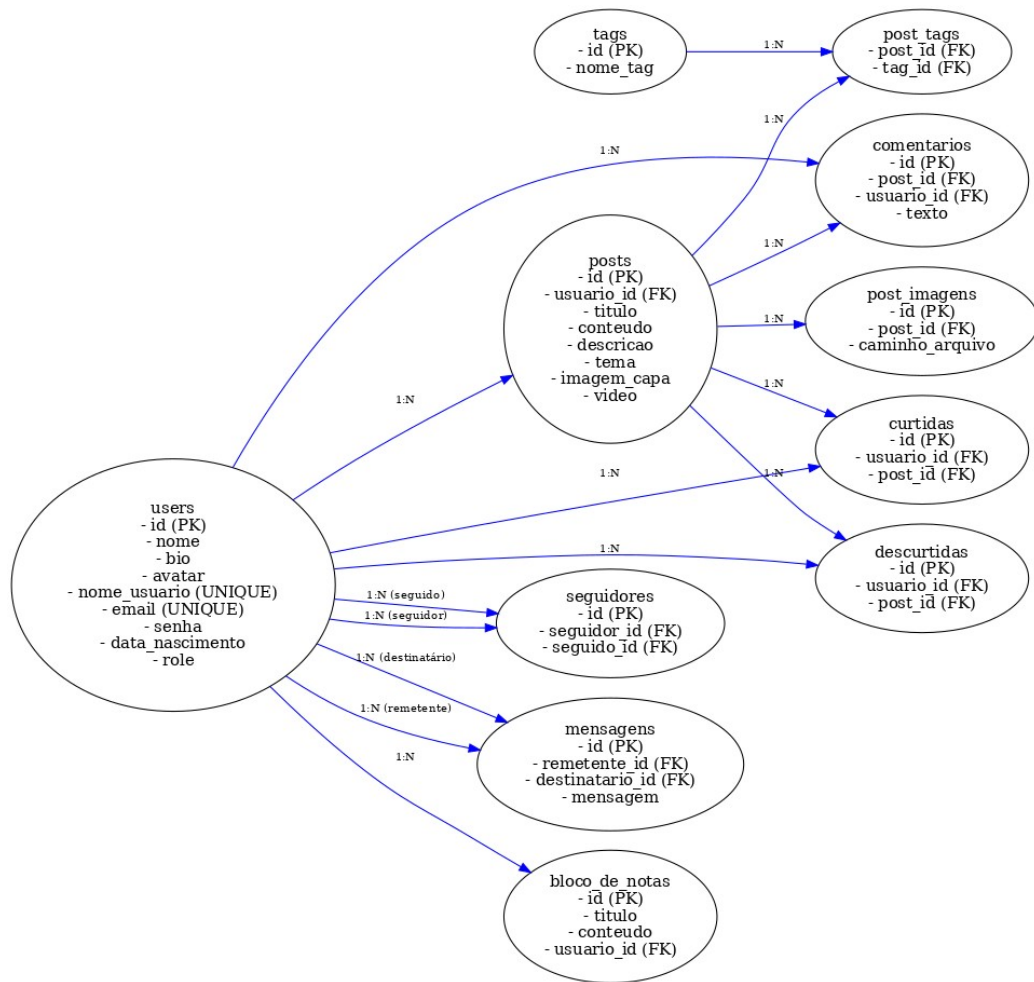
Nenhum arquivo escolhido

Nome:

Bio:

Salvar Alterações

Como meus dados são estruturados no banco de dados?



Quero interagir com meus colegas, cadê o Chat?

O chatUp conta com um sistema completo de mensagens diretas (DMs) entre usuários, funcionando em tempo real com AJAX e exibindo apenas contatos que o usuário segue.

O que ele faz?

- Exibe lista de usuários que você segue, permitindo iniciar uma conversa;
- Carrega o histórico da conversa entre os dois usuários;
- Permite enviar e receber mensagens instantaneamente;
- Atualiza o chat automaticamente a cada 2 segundos com AJAX (setInterval + fetch);
- Integra com o sistema de perfis e imagens de avatar.

