

## Projeto: Controle de Estoque

### 1) Estrutura:

- admin (pasta)
  - gerenciar\_materiais.php
  - gerenciar\_solicitacoes.php
  - gerenciar\_usuarios.php
  - index.php
  - relatorios.php
- api (pasta)
  - get\_stock.php
- assets (pasta)
  - css (pasta)
    - style.css
- auth (pasta)
  - login.php
  - logout.php
- requisitante (pasta)
  - criar\_solicitacao.php
  - historico.php
  - index.php
- templates (pasta)
  - footer.php
  - header.php
- config.php
- Criar\_Base.txt
- gerar\_hash.php
- index.php
- ver\_solicitacao.php

### 2) Acesso para administrador (recomendação)

Usuário (E-mail): admin@escola.com

Senha: admin123

### 3) Códigos na pasta raiz do projeto:

**Criar\_Base.txt:** Comandos para execução "SQL" da base padrão

```
-- =====
-- REESTRUTURAÇÃO COMPLETA DO BANCO DE DADOS
-- Sistema: Controle de Estoque - Escola Técnica de Enfermagem
-- =====

-- Criar banco de dados
DROP DATABASE IF EXISTS `escola_enfermagem_estoque`;
CREATE DATABASE IF NOT EXISTS `escola_enfermagem_estoque`
    DEFAULT CHARACTER SET utf8mb4
    COLLATE utf8mb4_general_ci;

USE `escola_enfermagem_estoque`;
```

-----  
-- Tabela: usuarios  
-----

```
CREATE TABLE `usuarios` (  
  `id_usuario` INT(11) NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(100) NOT NULL,  
  `email` VARCHAR(100) NOT NULL,  
  `senha` VARCHAR(255) NOT NULL,  
  `perfil` ENUM('admin','requisitante') NOT NULL,  
  `status` TINYINT(1) NOT NULL DEFAULT 1,  
  `criado_em` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP(),  
  PRIMARY KEY (`id_usuario`),  
  UNIQUE KEY `email` (`email`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Inserir administrador padrão (senha: admin123)  
INSERT INTO `usuarios` (`nome`, `email`, `senha`, `perfil`, `status`) VALUES  
('Administrador Padrão', 'admin@escola.com', '\$2y\$10  
\$e.p6.jlqC2wZ6y.e0b5ZLOWW.3q.2E/n9w5t3jD/a3r2c3v4X5y6z.', 'admin', 1);

-----  
-- Tabela: materiais  
-----

```
CREATE TABLE `materiais` (  
  `id_material` INT(11) NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(100) NOT NULL,  
  `codigo_interno` VARCHAR(50) DEFAULT NULL,  
  `categoria` VARCHAR(50) NOT NULL,  
  `tipo` ENUM('consumivel','reutilizavel') NOT NULL,  
  `quantidade` INT(11) NOT NULL DEFAULT 0,  
  `quantidade_min` INT(11) NOT NULL DEFAULT 0,  
  `status` TINYINT(1) NOT NULL DEFAULT 1,  
  `criado_em` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP(),  
  PRIMARY KEY (`id_material`),  
  UNIQUE KEY `codigo_interno` (`codigo_interno`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-----  
-- Tabela: solicitacoes  
-----

```
CREATE TABLE `solicitacoes` (  
  `id_solicitacao` INT(11) NOT NULL AUTO_INCREMENT,  
  `id_usuario` INT(11) NOT NULL,  
  `data_prevista` DATE NOT NULL,  
  `hora_prevista` TIME NOT NULL,  
  `local_previsto` VARCHAR(100) NOT NULL,  
  `status` ENUM('pendente','aprovado','recusado','entregue','devolvido') NOT NULL DEFAULT  
'pendente',  
  `observacoes` TEXT DEFAULT NULL,  
  `criado_em` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP(),
```

```
`atualizado_em` DATETIME DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP(),  
PRIMARY KEY (`id_solicitacao`),  
KEY `id_usuario` (`id_usuario`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
-- Tabela: solicitacao_itens
```

```
CREATE TABLE `solicitacao_itens` (  
  `id_item` INT(11) NOT NULL AUTO_INCREMENT,  
  `id_solicitacao` INT(11) NOT NULL,  
  `id_material` INT(11) NOT NULL,  
  `quantidade_solic` INT(11) NOT NULL,  
  `quantidade_entreg` INT(11) DEFAULT 0,  
  `quantidade_dev` INT(11) DEFAULT 0,  
  PRIMARY KEY (`id_item`),  
  KEY `id_solicitacao` (`id_solicitacao`),  
  KEY `id_material` (`id_material`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
-- Tabela: logs_auditoria
```

```
CREATE TABLE `logs_auditoria` (  
  `id_log` INT(11) NOT NULL AUTO_INCREMENT,  
  `id_usuario` INT(11) NOT NULL,  
  `acao` VARCHAR(100) NOT NULL,  
  `detalhes` TEXT DEFAULT NULL,  
  `data_hora` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP(),  
  PRIMARY KEY (`id_log`),  
  KEY `id_usuario` (`id_usuario`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
-- Chaves Estrangeiras
```

```
ALTER TABLE `solicitacoes`  
  ADD CONSTRAINT `solicitacoes_ibfk_1` FOREIGN KEY (`id_usuario`) REFERENCES `usuarios`  
  (`id_usuario`) ON DELETE CASCADE;
```

```
ALTER TABLE `solicitacao_itens`  
  ADD CONSTRAINT `solicitacao_itens_ibfk_1` FOREIGN KEY (`id_solicitacao`) REFERENCES  
  `solicitacoes` (`id_solicitacao`) ON DELETE CASCADE,  
  ADD CONSTRAINT `solicitacao_itens_ibfk_2` FOREIGN KEY (`id_material`) REFERENCES `materiais`  
  (`id_material`) ON DELETE CASCADE;
```

```
ALTER TABLE `logs_auditoria`  
  ADD CONSTRAINT `logs_auditoria_ibfk_1` FOREIGN KEY (`id_usuario`) REFERENCES `usuarios`  
  (`id_usuario`) ON DELETE CASCADE;
```

**gerar\_hash.php:** Se a pasta do projeto for "sistema\_estoque" executar da seguinte forma:  
[http://localhost/sistema\\_estoque/gerar\\_hash.php](http://localhost/sistema_estoque/gerar_hash.php)

```
<?php
// Defina a senha que você deseja usar
$senha_para_testar = 'admin123';

// Gera o hash seguro
$hash = password_hash($senha_para_testar, PASSWORD_DEFAULT);

// Exibe o hash
echo "<h1>Hash Gerado</h1>";
echo "<p>Sua senha '<strong>' . htmlspecialchars($senha_para_testar) . "</strong>' gerou o
seguinte hash:</p>";
echo "<textarea rows='3' cols='80' readonly>" . htmlspecialchars($hash) . "</textarea>";
echo "<p>Copie o código acima e siga as instruções para atualizar o banco de dados.</p>";
?>
```

### **config.php**

```
<?php
// Configurações do Banco de Dados
define('DB_HOST', 'localhost');
define('DB_PORT', '7306'); // <-- ADICIONE ESTA LINHA PARA A PORTA
define('DB_USER', 'root');
define('DB_PASS', '');
define('DB_NAME', 'escola_enfermagem_estoque');

// Habilitar exibição de erros para depuração
ini_set('display_errors', 1);
error_reporting(E_ALL);

// Iniciar a sessão
if (session_status() == PHP_SESSION_NONE) {
    session_start();
}

// Conexão com o banco de dados
try {
    // MODIFIQUE A LINHA ABAIXO para incluir a porta na string de conexão
    $dsn = "mysql:host=" . DB_HOST . ";port=" . DB_PORT . ";dbname=" . DB_NAME . ";charset=utf8";
    $pdo = new PDO($dsn, DB_USER, DB_PASS);

    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("ERRO: Não foi possível conectar ao banco de dados. " . $e->getMessage());
}

/**
 * Registra uma ação no log de auditoria.
 * @param PDO $pdo Objeto de conexão com o banco.
```

```

* @param int $id_usuario ID do usuário que realizou a ação.
* @param string $acao Descrição da ação.
* @param string $detalhes Detalhes adicionais.
*/
function registrar_log($pdo, $id_usuario, $acao, $detalhes = '')
{
    try {
        $sql = "INSERT INTO logs_auditoria (id_usuario, acao, detalhes) VALUES (?, ?, ?)";
        $stmt = $pdo->prepare($sql);
        $stmt->execute([$id_usuario, $acao, $detalhes]);
    } catch (PDOException $e) {
        error_log('Erro ao registrar log de auditoria: ' . $e->getMessage());
    }
}

// Função de verificação de login
function check_login($required_profile) {
    if (!isset($_SESSION['id_usuario']) || $_SESSION['perfil'] !== $required_profile) {
        // Registrar tentativa de acesso indevido
        if (isset($_SESSION['id_usuario'])) {
            global $pdo;
            registrar_log($pdo, $_SESSION['id_usuario'], 'Acesso Negado', 'Tentativa de acesso à área: ' .
            $required_profile);
        }
        header("Location: /sistema_estoque/auth/login.php");
        exit();
    }
}
?>

```

### index.php

```

<?php
require_once 'config.php';

if (!isset($_SESSION['id_usuario'])) {
    header("Location: auth/login.php");
    exit();
}

// Redireciona com base no perfil do usuário
if ($_SESSION['perfil'] === 'admin') {
    header("Location: admin/index.php");
    exit();
} elseif ($_SESSION['perfil'] === 'requisitante') {
    header("Location: requisitante/index.php");
    exit();
} else {
    // Se o perfil for desconhecido, faz logout por segurança
    header("Location: auth/logout.php");
    exit();
}

```

```
}  
?>
```

### **ver\_solicitacao.php**

```
<?php  
require_once 'templates/header.php';  
  
// Verificar se o usuário está logado  
if (!isset($_SESSION['id_usuario'])) {  
    header("Location: /sistema_estoque/auth/login.php");  
    exit();  
}  
  
$id_solicitacao = filter_input(INPUT_GET, 'id', FILTER_VALIDATE_INT);  
if (!$id_solicitacao) {  
    echo "<p class='error'>ID de solicitação inválido.</p>";  
    require_once 'templates/footer.php';  
    exit();  
}  
  
try {  
    // Buscar dados da solicitação e nome do requisitante  
    $stmt_solicitacao = $pdo->prepare("  
        SELECT s.*, u.nome AS nome_usuario, u.email AS email_usuario  
        FROM solicitacoes s  
        JOIN usuarios u ON s.id_usuario = u.id_usuario  
        WHERE s.id_solicitacao = ?  
    ");  
    $stmt_solicitacao->execute([$id_solicitacao]);  
    $solicitacao = $stmt_solicitacao->fetch(PDO::FETCH_ASSOC);  
  
    if (!$solicitacao) {  
        throw new Exception("Solicitação não encontrada.");  
    }  
  
    // VERIFICAÇÃO DE PERMISSÃO:  
    // Um requisitante só pode ver suas próprias solicitações. Um admin pode ver todas.  
    if ($_SESSION['perfil'] === 'requisitante' && $solicitacao['id_usuario'] != $_SESSION['id_usuario']) {  
        registrar_log($pdo, $_SESSION['id_usuario'], 'Acesso Negado', "Tentativa de ver solicitação  
#{ $id_solicitacao } de outro usuário.");  
        throw new Exception("Acesso negado. Você não tem permissão para ver esta solicitação.");  
    }  
  
    // Buscar itens da solicitação  
    $stmt_itens = $pdo->prepare("  
        SELECT si.*, m.nome AS nome_material, m.codigo_interno  
        FROM solicitacao_itens si  
        JOIN materiais m ON si.id_material = m.id_material  
        WHERE si.id_solicitacao = ?  
    ");
```

```

$stmt_iteens->execute([$id_solicitacao]);
$iteens = $stmt_iteens->fetchAll(PDO::FETCH_ASSOC);

} catch (Exception $e) {
    echo "<p class='error'>Erro: " . htmlspecialchars($e->getMessage()) . "</p>";
    require_once 'templates/footer.php';
    exit();
}
?>

<h2>Detalhes da Solicitação Nº <?php echo $solicitacao['id_solicitacao']; ?></h2>

<div class="solicitacao-details">
    <div><strong>Status:</strong> <span class="status-badge status-<?php echo
htmlspecialchars($solicitacao['status']); ?>"><?php echo
htmlspecialchars(ucfirst($solicitacao['status'])); ?></span></div>
    <div><strong>Requisitante:</strong> <?php echo
htmlspecialchars($solicitacao['nome_usuario']); ?> (<?php echo
htmlspecialchars($solicitacao['email_usuario']); ?>)</div>
    <div><strong>Data de Criação:</strong> <?php echo date('d/m/Y H:i',
strtotime($solicitacao['criado_em'])); ?></div>
    <div><strong>Data Prevista para Uso:</strong> <?php echo date('d/m/Y',
strtotime($solicitacao['data_prevista'])); ?> às <?php echo date('H:i',
strtotime($solicitacao['hora_prevista'])); ?></div>
    <div><strong>Local Previsto para Uso:</strong> <?php echo
htmlspecialchars($solicitacao['local_previsto']); ?></div>
    <?php if (!empty($solicitacao['observacoes'])): ?>
        <div><strong>Observações:</strong> <?php echo
nl2br(htmlspecialchars($solicitacao['observacoes'])); ?></div>
    <?php endif; ?>
</div>

<h3>Itens Solicitados</h3>
<div class="table-wrapper">
    <table class="responsive-table">
        <thead>
            <tr>
                <th>Material</th>
                <th>Código Interno</th>
                <th>Qtd. Solicitada</th>
                <th>Qtd. Entregue</th>
                <th>Qtd. Devolvida</th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($iteens as $item): ?>
                <tr>
                    <td><?php echo htmlspecialchars($item['nome_material']); ?></td>
                    <td><?php echo htmlspecialchars($item['codigo_interno']); ?></td>
                    <td><?php echo $item['quantidade_solic']; ?></td>
                    <td><?php echo $item['quantidade_entreg'] ?? 'N/D'; ?></td>
                </tr>
            </tbody>
        </table>
    </div>

```

```

                <td><?php echo $item['quantidade_dev'] ?? 'N/D'; ?></td>
            </tr>
        <?php endforeach; ?>
    </tbody>
</table>
</div>

```

```

<a href="javascript:history.back()" class="btn btn-secondary" style="margin-top: 20px;">Voltar</a>

```

```

<style>
.solicitacao-details {
    background-color: #f8f9fa;
    border: 1px solid #dee2e6;
    border-radius: 8px;
    padding: 20px;
    margin-bottom: 30px;
    display: flex;
    flex-direction: column;
    gap: 10px;
}
.status-badge {
    padding: 3px 8px;
    border-radius: 12px;
    color: white;
    font-weight: bold;
    font-size: 0.9em;
}
.status-pendente { background-color: #ffc107; color: #333; }
.status-aprovado { background-color: #28a745; }
.status-recusado { background-color: #dc3545; }
.status-entregue { background-color: #17a2b8; }
.status-devolvido { background-color: #6c757d; }
</style>

```

```

<?php require_once 'templates/footer.php'; ?>

```

## PASTA admin:

### gerenciar\_materiais.php

```

<?php
require_once '../templates/header.php';
check_login('admin');

$success_message = "";
$error_message = "";
$edit_material = null;

// Lógica para ADICIONAR ou EDITAR material
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['form_submit'])) {
    // Sanitização e validação

```



```

$id_material = filter_input(INPUT_POST, 'id_material', FILTER_VALIDATE_INT);
$nome = trim(filter_input(INPUT_POST, 'nome', FILTER_SANITIZE_STRING));
$codigo_interno = trim(filter_input(INPUT_POST, 'codigo_interno', FILTER_SANITIZE_STRING));
$categoria = trim(filter_input(INPUT_POST, 'categoria', FILTER_SANITIZE_STRING));
$tipo = filter_input(INPUT_POST, 'tipo', FILTER_SANITIZE_STRING);
$quantidade = filter_input(INPUT_POST, 'quantidade', FILTER_VALIDATE_INT);
$quantidade_min = filter_input(INPUT_POST, 'quantidade_min', FILTER_VALIDATE_INT);
$status = filter_input(INPUT_POST, 'status', FILTER_VALIDATE_INT);

try {
    if ($id_material) { // ATUALIZAÇÃO
        $sql = "UPDATE materiais SET nome=?, codigo_interno=?, categoria=?, tipo=?, quantidade=?,
quantidade_min=?, status=? WHERE id_material=?";
        $stmt = $pdo->prepare($sql);
        $stmt->execute([$nome, $codigo_interno, $categoria, $tipo, $quantidade, $quantidade_min,
$status, $id_material]);
        registrar_log($pdo, $_SESSION['id_usuario'], 'Atualização de Material', "ID: {$id_material}");
        $success_message = 'Material atualizado com sucesso!';
    } else { // CRIAÇÃO
        $sql = "INSERT INTO materiais (nome, codigo_interno, categoria, tipo, quantidade,
quantidade_min, status) VALUES (?, ?, ?, ?, ?, ?, ?)";
        $stmt = $pdo->prepare($sql);
        $stmt->execute([$nome, $codigo_interno, $categoria, $tipo, $quantidade, $quantidade_min,
$status]);
        $new_id = $pdo->lastInsertId();
        registrar_log($pdo, $_SESSION['id_usuario'], 'Criação de Material', "ID: {$new_id}");
        $success_message = 'Material criado com sucesso!';
    }
} catch (PDOException $e) {
    $error_message = 'Erro ao salvar material: ' . $e->getMessage();
}
}

// Carregar dados para edição
if (isset($_GET['edit'])) {
    $id_edit = filter_input(INPUT_GET, 'edit', FILTER_VALIDATE_INT);
    $stmt = $pdo->prepare("SELECT * FROM materiais WHERE id_material = ?");
    $stmt->execute([$id_edit]);
    $edit_material = $stmt->fetch(PDO::FETCH_ASSOC);
}

// Buscar todos os materiais
$materiais = $pdo->query("SELECT * FROM materiais ORDER BY nome")->
fetchAll(PDO::FETCH_ASSOC);
?>

```

<h2>Gerenciar Materiais</h2>

```

<?php if ($success_message): ?><p class="success"><?php echo
htmlspecialchars($success_message); ?></p><?php endif; ?>
<?php if ($error_message): ?><p class="error"><?php echo htmlspecialchars($error_message); ?>

```

</p><?php endif; ?>

<div class="form-container">

<h3><?php echo \$edit\_material ? 'Editar Material' : 'Adicionar Novo Material'; ?></h3>

<form action="gerenciar\_materiais.php" method="POST">

<input type="hidden" name="form\_submit" value="1">

<input type="hidden" name="id\_material" value="<?php echo \$edit\_material['id\_material'] ?? '' ?>" required>

<div class="form-group">

<label for="nome">Nome do Material</label>

<input type="text" name="nome" value="<?php echo htmlspecialchars(\$edit\_material['nome'] ?? ''); ?>" required>

</div>

<!-- Outros campos do formulário (código, categoria, etc.) -->

<div class="form-group">

<label>Código Interno</label>

<input type="text" name="codigo\_interno" value="<?php echo htmlspecialchars(\$edit\_material['codigo\_interno'] ?? ''); ?>" required>

</div>

<div class="form-group">

<label>Categoria</label>

<input type="text" name="categoria" value="<?php echo htmlspecialchars(\$edit\_material['categoria'] ?? ''); ?>" required>

</div>

<div class="form-group">

<label>Tipo</label>

<select name="tipo" required>  
<option value="consumivel" <?php echo (isset(\$edit\_material['tipo']) && \$edit\_material['tipo'] == 'consumivel') ? 'selected' : '' ?>>Consumível</option>  
<option value="reutilizavel" <?php echo (isset(\$edit\_material['tipo']) && \$edit\_material['tipo'] == 'reutilizavel') ? 'selected' : '' ?>>Reutilizável</option>  
</select>

</div>

<div class="form-group">

<label>Quantidade em Estoque</label>

<input type="number" name="quantidade" value="<?php echo \$edit\_material['quantidade'] ?? 0; ?>" required>

</div>

<div class="form-group">

<label>Estoque Mínimo</label>

<input type="number" name="quantidade\_min" value="<?php echo \$edit\_material['quantidade\_min'] ?? 0; ?>" required>

</div>

<div class="form-group">

<label>Status</label>

<select name="status" required>  
<option value="1" <?php echo (isset(\$edit\_material['status']) && \$edit\_material['status'] == 1) ? 'selected' : '' ?>>Ativo</option>  
<option value="0" <?php echo (isset(\$edit\_material['status']) && \$edit\_material['status'] == 0) ? 'selected' : '' ?>>Inativo</option>

```

        </select>
    </div>

    <button type="submit"><?php echo $edit_material ? 'Atualizar Material' : 'Salvar Material'; ?>
</button>
    <?php if ($edit_material): ?>
        <a href="gerenciar_materiais.php" class="btn btn-secondary">Cancelar Edição</a>
    <?php endif; ?>
</form>
</div>

```

<h3>Materiais em Estoque</h3>

<table class="responsive-table">

<thead>

<tr>

<th>Nome</th>

<th>Categoria</th>

<th>Tipo</th>

<th>Qtd.</th>

<th>Status</th>

<th>Ações</th>

</tr>

</thead>

<tbody>

<?php foreach (\$materiais as \$material): ?>

<tr>

<td><?php echo htmlspecialchars(\$material['nome']); ?></td>

<td><?php echo htmlspecialchars(\$material['categoria']); ?></td>

<td><?php echo htmlspecialchars(ucfirst(\$material['tipo'])); ?></td>

<td><?php echo \$material['quantidade']; ?></td>

<td><?php echo \$material['status'] ? 'Ativo' : 'Inativo'; ?></td>

<td><a href="gerenciar\_materiais.php?edit=<?php echo \$material['id\_material']; ?>" class="btn">Editar</a></td>

</tr>

<?php endforeach; ?>

</tbody>

</table>

<?php require\_once '../templates/footer.php'; ?>

### gerenciar\_solicitacoes.php

<?php

require\_once '../templates/header.php';

check\_login('admin');

\$success\_message = '';

\$error\_message = '';

// Bloco de código para processar a atualização de status via formulário POST

if (\$\_SERVER['REQUEST\_METHOD'] === 'POST' && isset(\$\_POST['id\_solicitacao'])) {

```

$id_solicitacao = filter_input(INPUT_POST, 'id_solicitacao', FILTER_VALIDATE_INT);
$novo_status = filter_input(INPUT_POST, 'status', FILTER_SANITIZE_STRING);

if ($id_solicitacao && in_array($novo_status, ['pendente', 'aprovado', 'recusado', 'entregue', 'devolvido'])) {
    try {
        $pdo->beginTransaction();

        // Busca o status atual da solicitação para validação
        $stmt_old = $pdo->prepare("SELECT status FROM solicitacoes WHERE id_solicitacao = ? FOR UPDATE");
        $stmt_old->execute([$id_solicitacao]);
        $status_antigo = $stmt_old->fetchColumn();

        // Evita processamento desnecessário se o status for o mesmo
        if ($status_antigo === $novo_status) {
            throw new Exception("A solicitação já está com o status '{$novo_status}'. Nenhuma alteração foi feita.");
        }

        // Define as regras de transição de status permitidas (a "esteira")
        $allowed_transitions = [
            'pendente' => ['aprovado', 'recusado', 'entregue'],
            'aprovado' => ['entregue', 'recusado'],
            'recusado' => [], // De 'recusado' não se pode ir para nenhum outro estado
            'entregue' => ['devolvido'],
            'devolvido' => [] // De 'devolvido' também não
        ];

        // Valida se a mudança de status é permitida
        if (!in_array($novo_status, $allowed_transitions[$status_antigo])) {
            throw new Exception("Ação inválida: não é possível alterar o status de '{$status_antigo}' para '{$novo_status}'.");
        }

        // Lógica para quando o status muda para "Entregue"
        if ($novo_status === 'entregue' && $status_antigo !== 'entregue') {
            $itens_stmt = $pdo->prepare("
                SELECT si.id_material, si.quantidade_solic, m.nome AS nome_material, m.quantidade AS
quantidade_atual
                FROM solicitacao_itens si JOIN materiais m ON si.id_material = m.id_material
                WHERE si.id_solicitacao = ?
            ");
            $itens_stmt->execute([$id_solicitacao]);
            $itens_para_entrega = $itens_stmt->fetchAll(PDO::FETCH_ASSOC);

            // VERIFICAÇÃO DE ESTOQUE CRÍTICA NO LADO DO SERVIDOR
            foreach ($itens_para_entrega as $item) {
                if ($item['quantidade_solic'] > $item['quantidade_atual']) {
                    throw new Exception(
                        "Estoque insuficiente para o item '<strong>' .

```

```

htmlspecialchars($item['nome_material']) . "</strong>". " .
        "Solicitado: {$item['quantidade_solic']}, Disponível: {$item['quantidade_atual']}.
Operação cancelada."
    );
}
}

// Se passou na verificação, prossegue com a baixa de estoque
foreach ($itens_para_entrega as $item) {
    $pdo->prepare("UPDATE materiais SET quantidade = quantidade - ? WHERE id_material
= ?")->execute([$item['quantidade_solic'], $item['id_material']]);
    $pdo->prepare("UPDATE solicitacao_itens SET quantidade_entreg = ? WHERE
id_solicitacao = ? AND id_material = ?")->execute([$item['quantidade_solic'], $id_solicitacao,
$item['id_material']]);
}
    registrar_log($pdo, $_SESSION['id_usuario'], 'Entrega de Material', "Solicitação ID:
{$id_solicitacao}");
}

// Lógica para quando o status muda para "Devolvido"
elseif ($novo_status === 'devolvido' && $status_antigo === 'entregue') {
    $itens_stmt = $pdo->prepare("
        SELECT si.id_material, si.quantidade_entreg
        FROM solicitacao_itens si JOIN materiais m ON si.id_material = m.id_material
        WHERE si.id_solicitacao = ? AND m.tipo = 'reutilizavel'
    ");
    $itens_stmt->execute([$id_solicitacao]);
    $itens_reutilizaveis = $itens_stmt->fetchAll(PDO::FETCH_ASSOC);

    foreach ($itens_reutilizaveis as $item) {
        $pdo->prepare("UPDATE materiais SET quantidade = quantidade + ? WHERE id_material
= ?")->execute([$item['quantidade_entreg'], $item['id_material']]);
        $pdo->prepare("UPDATE solicitacao_itens SET quantidade_dev = ? WHERE id_solicitacao
= ? AND id_material = ?")->execute([$item['quantidade_entreg'], $id_solicitacao,
$item['id_material']]);
    }
    registrar_log($pdo, $_SESSION['id_usuario'], 'Devolução de Material', "Solicitação ID:
{$id_solicitacao}");
}

// Atualiza o status da solicitação no banco
    $pdo->prepare("UPDATE solicitacoes SET status = ? WHERE id_solicitacao = ?")->
execute([$novo_status, $id_solicitacao]);
    registrar_log($pdo, $_SESSION['id_usuario'], 'Mudança de Status de Solicitação', "ID:
{$id_solicitacao}, de '{$status_antigo}' para '{$novo_status}'");

    $pdo->commit();
    $success_message = "Status da solicitação #{$id_solicitacao} atualizado com sucesso!";

} catch (Exception $e) {
    $pdo->rollBack();
}

```

```

        $error_message = $e->getMessage();
    }
}
}

```

```

// Bloco para buscar os dados para exibição na página
// CORREÇÃO: Usando LEFT JOIN para garantir que todas as solicitações sejam listadas
$stmt_solicitacoes = $pdo->query("
    SELECT s.*, u.nome as nome_usuario
    FROM solicitacoes s
    LEFT JOIN usuarios u ON s.id_usuario = u.id_usuario
    ORDER BY s.criado_em DESC
");
$solicitacoes = $stmt_solicitacoes->fetchAll(PDO::FETCH_ASSOC);
?>

```

<h2>Gerenciar Solicitações</h2>

```

<?php if ($success_message): ?><p class="success"><?php echo
htmlspecialchars($success_message); ?></p><?php endif; ?>
<?php if ($error_message): ?><p class="error"><?php echo $error_message; // Permite HTML como
<strong> na msg de erro ?></p><?php endif; ?>

```

```

<div class="table-wrapper">
    <table class="responsive-table">
        <thead>
            <tr>
                <th>ID</th>
                <th>Requisitante</th>
                <th>Data Prevista</th>
                <th>Status</th>
                <th>Ações</th>
            </tr>
        </thead>
        <tbody>
            <?php if (empty($solicitacoes)): ?>
                <tr>
                    <td colspan="5" style="text-align: center; padding: 20px;">Nenhuma solicitação
encontrada.</td>
                </tr>
            <?php else: ?>
                <?php foreach ($solicitacoes as $s): ?>
                    <tr>
                        <td><?php echo $s['id_solicitacao']; ?></td>
                        <td>
                            <?php echo htmlspecialchars($s['nome_usuario'] ?? 'Usuário Removido'); ?>
                        </td>
                        <td><?php echo date('d/m/Y', strtotime($s['data_prevista'])); ?></td>
                        <td><strong class="status-badge status-<?php echo htmlspecialchars($s['status']); ?>">
<?php echo htmlspecialchars(ucfirst($s['status'])); ?></strong></td>
                        <td style="display: flex; flex-wrap: wrap; gap: 5px; align-items: center;">

```

```

        <form action="gerenciar_solicitacoes.php" method="POST" style="display:inline-flex;
gap: 5px; margin: 0;">
        <input type="hidden" name="id_solicitacao" value="<?php echo
        $s['id_solicitacao']; ?>">
        <select name="status">
            <option value="pendente" <?php if($s['status'] == 'pendente') echo 'selected'; ?>>
Pendente</option>
            <option value="aprovado" <?php if($s['status'] == 'aprovado') echo 'selected'; ?>>
Aprovado</option>
            <option value="recusado" <?php if($s['status'] == 'recusado') echo 'selected'; ?>>
Recusado</option>
            <option value="entregue" <?php if($s['status'] == 'entregue') echo 'selected'; ?>>
Entregue</option>
            <option value="devolvido" <?php if($s['status'] == 'devolvido') echo 'selected'; ?>>
Devolvido</option>
        </select>
        <button type="submit" class="btn btn-secondary" style="padding: 8px 12px;">
Alterar</button>
        </form>
        <a href=" ../ver_solicitacao.php?id=<?php echo $s['id_solicitacao']; ?>" class="btn">Ver
Detalhes</a>
        </td>
    </tr>
    <?php endforeach; ?>
    <?php endif; ?>
</tbody>
</table>
</div>

<?php
require_once '../templates/footer.php';
?>

```

### gerenciar\_usuarios.php

```

<?php
require_once '../templates/header.php';
check_login('admin');

$success_message = "";
$error_message = "";
$edit_user = null;

// Lógica para ADICIONAR ou EDITAR usuário
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $id_usuario = filter_input(INPUT_POST, 'id_usuario', FILTER_VALIDATE_INT);
    $nome = trim(filter_input(INPUT_POST, 'nome', FILTER_SANITIZE_STRING));
    $email = trim(filter_input(INPUT_POST, 'email', FILTER_VALIDATE_EMAIL));
    $perfil = filter_input(INPUT_POST, 'perfil', FILTER_SANITIZE_STRING);
    $status = filter_input(INPUT_POST, 'status', FILTER_VALIDATE_INT);
    $senha = $_POST['senha']; // Senha não é filtrada para permitir caracteres especiais
}

```

```

try {
    if ($id_usuario) { // ATUALIZAÇÃO
        if (!empty($senha)) {
            $hash_senha = password_hash($senha, PASSWORD_DEFAULT);
            $stmt = $pdo->prepare("UPDATE usuarios SET nome=?, email=?, perfil=?, status=?,
senha=? WHERE id_usuario=?");
            $stmt->execute([$nome, $email, $perfil, $status, $hash_senha, $id_usuario]);
        } else {
            $stmt = $pdo->prepare("UPDATE usuarios SET nome=?, email=?, perfil=?, status=? WHERE
id_usuario=?");
            $stmt->execute([$nome, $email, $perfil, $status, $id_usuario]);
        }
        registrar_log($pdo, $_SESSION['id_usuario'], 'Atualização de Usuário', "ID: {$id_usuario}");
        $success_message = 'Usuário atualizado com sucesso!';
    } else { // CRIAÇÃO
        if (empty($senha)) {
            throw new Exception("O campo senha é obrigatório para novos usuários.");
        }
        $hash_senha = password_hash($senha, PASSWORD_DEFAULT);
        $stmt = $pdo->prepare("INSERT INTO usuarios (nome, email, senha, perfil, status) VALUES
(?, ?, ?, ?, ?)");
        $stmt->execute([$nome, $email, $hash_senha, $perfil, $status]);
        $new_id = $pdo->lastInsertId();
        registrar_log($pdo, $_SESSION['id_usuario'], 'Criação de Usuário', "ID: {$new_id}");
        $success_message = 'Usuário criado com sucesso!';
    }
} catch (Exception $e) {
    $error_message = 'Erro: ' . $e->getMessage();
}
}

```

// Lógica para carregar dados de um usuário para edição

```

if (isset($_GET['edit'])) {
    $id_edit = filter_input(INPUT_GET, 'edit', FILTER_VALIDATE_INT);
    $stmt = $pdo->prepare("SELECT * FROM usuarios WHERE id_usuario = ?");
    $stmt->execute([$id_edit]);
    $edit_user = $stmt->fetch(PDO::FETCH_ASSOC);
}

```

// Buscar todos os usuários para listar

```

$usuarios = $pdo->query("SELECT id_usuario, nome, email, perfil, status FROM usuarios ORDER BY
nome")->fetchAll(PDO::FETCH_ASSOC);
?>

```

<h2>Gerenciar Usuários</h2>

```

<?php if ($success_message): ?><p class="success"><?php echo
htmlspecialchars($success_message); ?></p><?php endif; ?>
<?php if ($error_message): ?><p class="error"><?php echo htmlspecialchars($error_message); ?>
</p><?php endif; ?>

```



```

<!-- Formulário de Criação/Edição -->
<div class="form-container">
  <h3><?php echo $edit_user ? 'Editar Usuário' : 'Adicionar Novo Usuário'; ?></h3>
  <form action="gerenciar_usuarios.php" method="POST">
    <input type="hidden" name="id_usuario" value="<?php echo $edit_user['id_usuario'] ?? ''; ?>">

    <div class="form-group">
      <label for="nome">Nome Completo</label>
      <input type="text" id="nome" name="nome" value="<?php echo
htmlspecialchars($edit_user['nome'] ?? ''); ?>" required>
    </div>

    <div class="form-group">
      <label for="email">E-mail</label>
      <input type="email" id="email" name="email" value="<?php echo
htmlspecialchars($edit_user['email'] ?? ''); ?>" required>
    </div>

    <div class="form-group">
      <label for="senha">Senha</label>
      <input type="password" id="senha" name="senha" placeholder="<?php echo $edit_user ?
'Deixe em branco para não alterar' : ''; ?>">
    </div>

    <div class="form-group">
      <label for="perfil">Perfil</label>
      <select id="perfil" name="perfil" required>
        <option value="requisitante" <?php echo (isset($edit_user['perfil']) && $edit_user['perfil']
== 'requisitante') ? 'selected' : ''; ?>>Requisitante</option>
        <option value="admin" <?php echo (isset($edit_user['perfil']) && $edit_user['perfil'] ==
'admin') ? 'selected' : ''; ?>>Administrador</option>
      </select>
    </div>

    <div class="form-group">
      <label for="status">Status</label>
      <select id="status" name="status" required>
        <option value="1" <?php echo (!isset($edit_user['status']) || $edit_user['status'] == 1) ?
'selected' : ''; ?>>Ativo</option>
        <option value="0" <?php echo (isset($edit_user['status']) && $edit_user['status'] == 0) ?
'selected' : ''; ?>>Inativo</option>
      </select>
    </div>

    <button type="submit"><?php echo $edit_user ? 'Atualizar Usuário' : 'Salvar Usuário'; ?>
  </button>
  <?php if ($edit_user): ?>
    <a href="gerenciar_usuarios.php" class="btn btn-secondary">Cancelar Edição</a>
  <?php endif; ?>
</form>

```

</div>

<h3>Usuários Cadastrados</h3>

<table class="responsive-table">

<thead>

<tr>

<th>Nome</th>

<th>E-mail</th>

<th>Perfil</th>

<th>Status</th>

<th>Ações</th>

</tr>

</thead>

<tbody>

<?php foreach (\$usuarios as \$usuario): ?>

<tr>

<td><?php echo htmlspecialchars(\$usuario['nome']); ?></td>

<td><?php echo htmlspecialchars(\$usuario['email']); ?></td>

<td><?php echo htmlspecialchars(ucfirst(\$usuario['perfil'])); ?></td>

<td><?php echo \$usuario['status'] ? 'Ativo' : 'Inativo'; ?></td>

<td><a href="gerenciar\_usuarios.php?edit=<?php echo \$usuario['id\_usuario']; ?>"

class="btn">Editar</a></td>

</tr>

<?php endforeach; ?>

</tbody>

</table>

<?php require\_once '../templates/footer.php'; ?>

#### index.php

<?php

require\_once '../templates/header.php';

check\_login('admin');

try {

// Contagem de solicitações pendentes

\$stmt\_pendentes = \$pdo->query("SELECT COUNT(\*) FROM solicitacoes WHERE status = 'pendente'");

\$solicitacoes\_pendentes = \$stmt\_pendentes->fetchColumn();

// Contagem de materiais com estoque baixo

\$stmt\_estoque\_baixo = \$pdo->query("SELECT COUNT(\*) FROM materiais WHERE quantidade < quantidade\_min AND status = 1");

\$estoque\_baixo = \$stmt\_estoque\_baixo->fetchColumn();

// Contagem de usuários ativos

\$stmt\_usuarios = \$pdo->query("SELECT COUNT(\*) FROM usuarios WHERE status = 1");

\$usuarios\_ativos = \$stmt\_usuarios->fetchColumn();

} catch (PDOException \$e) {

```

    echo "<p class='error'>Erro ao carregar dados do painel: " . htmlspecialchars($e->getMessage()) .
    "</p>";
    // Definir valores padrão em caso de erro
    $solicitacoes_pendentes = $estoque_baixo = $usuarios_ativos = 'N/A';
}
?>

```

```

<h2>Painel do Administrador</h2>

```

```

<p>Bem-vindo(a), <?php echo htmlspecialchars($_SESSION['nome']); ?>!</p>

```

```

<div class="dashboard-widgets">
    <div class="widget">
        <h3>Solicitações Pendentes</h3>
        <p class="widget-value"><?php echo $solicitacoes_pendentes; ?></p>
        <a href="gerenciar_solicitacoes.php" class="widget-link">Ver Solicitações</a>
    </div>
    <div class="widget">
        <h3>Itens com Estoque Baixo</h3>
        <p class="widget-value"><?php echo $estoque_baixo; ?></p>
        <a href="gerenciar_materiais.php" class="widget-link">Ver Materiais</a>
    </div>
    <div class="widget">
        <h3>Usuários Ativos</h3>
        <p class="widget-value"><?php echo $usuarios_ativos; ?></p>
        <a href="gerenciar_usuarios.php" class="widget-link">Ver Usuários</a>
    </div>
</div>

```

```

<style>
.dashboard-widgets {
    display: flex;
    gap: 20px;
    flex-wrap: wrap;
    margin-top: 20px;
}
.widget {
    background-color: #f8f9fa;
    border: 1px solid #dee2e6;
    border-radius: 8px;
    padding: 20px;
    flex-grow: 1;
    min-width: 250px;
    text-align: center;
}
.widget h3 {
    margin-top: 0;
}
.widget-value {
    font-size: 2.5rem;
    font-weight: bold;
    color: #007bff;
}

```

```

        margin: 10px 0;
    }
    .widget-link {
        text-decoration: none;
        color: #007bff;
        font-weight: bold;
    }
</style>

<?php require_once '../templates/footer.php'; ?>

```

## relatorios.php

```

<?php
require_once '../config.php';
check_login('admin');

if (isset($_GET['gerar_csv'])) {
    header('Content-Type: text/csv; charset=utf-8');
    header('Content-Disposition: attachment; filename=relatorio_materiais.csv');

    $output = fopen('php://output', 'w');
    fputcsv($output, ['ID', 'Nome', 'Categoria', 'Quantidade em Estoque', 'Status']);

    $stmt = $pdo->query("SELECT id_material, nome, categoria, quantidade, status FROM materiais
ORDER BY nome");
    while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
        $row['status'] = $row['status'] ? 'Ativo' : 'Inativo';
        fputcsv($output, $row);
    }
    fclose($output);
    exit();
}

require_once '../templates/header.php';
?>

<h2>Relatórios</h2>
<p>Selecione um tipo de relatório para gerar.</p>

<div class="relatorio-item">
    <h3>Relatório de Materiais em Estoque</h3>
    <p>Gera um arquivo CSV com a lista completa de materiais e suas quantidades atuais.</p>
    <a href="relatorios.php?gerar_csv=true" class="btn">Gerar Relatório CSV</a>
</div>

<!-- Outros relatórios podem ser adicionados aqui -->

<?php require_once '../templates/footer.php'; ?>

```

**PASTA api:**

## get\_stock.php

```
<?php
header('Content-Type: application/json');
require_once '../config.php';

// Apenas permitir acesso se o usuário estiver logado
if (!isset($_SESSION['id_usuario'])) {
    echo json_encode(['error' => 'Acesso não autorizado']);
    exit();
}

$id_material = filter_input(INPUT_GET, 'id', FILTER_VALIDATE_INT);

if (!$id_material) {
    echo json_encode(['error' => 'ID de material inválido']);
    exit();
}

try {
    $stmt = $pdo->prepare("SELECT quantidade FROM materiais WHERE id_material = ?");
    $stmt->execute([$id_material]);
    $material = $stmt->fetch(PDO::FETCH_ASSOC);

    if ($material) {
        echo json_encode(['quantidade' => $material['quantidade']]);
    } else {
        echo json_encode(['quantidade' => 0]);
    }
} catch (PDOException $e) {
    echo json_encode(['error' => 'Erro ao consultar o banco de dados.']);
}
?>
```

## PASTA assets -> SUBPASTA css

### style.css

```
/* --- GERAL E RESET --- */
*, *::before, *::after {
    box-sizing: border-box;
}

body, html {
    margin: 0;
    padding: 0;
    font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue", Arial,
    sans-serif;
    background-color: #f4f7f6;
    color: #333;
```

```
    line-height: 1.6;
}

body {
    display: flex;
    flex-direction: column;
    min-height: 100vh;
}

main {
    flex-grow: 1; /* Garante que o conteúdo principal ocupe o espaço disponível */
}

.container {
    width: 95%;
    max-width: 1200px;
    margin: 20px auto; /* Centraliza o container e adiciona margem superior/inferior */
    padding: 20px;
    background: #fff;
    box-shadow: 0 2px 5px rgba(0,0,0,0.05);
    border-radius: 8px;
}

/* --- HEADER E NAVEGAÇÃO --- */
header {
    background-color: #0056b3; /* Tom de azul mais profissional */
    color: white;
    padding: 1rem 5%;
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);
    width: 100%;
}

.header-container {
    display: flex;
    justify-content: space-between;
    align-items: center;
    width: 100%;
    max-width: 1200px;
    margin: 0 auto;
}

header h1 {
    margin: 0;
    font-size: 1.5rem;
}

header h1 a {
    color: white;
    text-decoration: none;
}
```

```
header nav ul {
  list-style: none;
  margin: 0;
  padding: 0;
  display: flex;
  gap: 20px; /* Espaçamento entre os itens do menu */
}
```

```
header nav ul li a {
  color: white;
  text-decoration: none;
  font-weight: bold;
  padding: 5px 10px;
  border-radius: 4px;
  transition: background-color 0.3s;
}
```

```
header nav ul li a:hover {
  background-color: rgba(255, 255, 255, 0.2);
}
```

```
.menu-toggle {
  display: none; /* Escondido em telas grandes */
  background: none;
  border: none;
  color: white;
  font-size: 2rem;
  cursor: pointer;
}
```

```
/* --- FORMULÁRIOS E BOTÕES --- */
```

```
.form-container, form {
  display: flex;
  flex-direction: column;
  gap: 15px; /* Espaçamento entre os campos */
}
```

```
.form-container {
  background-color: #f8f9fa;
  padding: 20px;
  border-radius: 8px;
  margin-bottom: 30px;
}
```

```
label {
  font-weight: bold;
  margin-bottom: 5px;
}
```

```
input[type="text"], input[type="email"], input[type="password"],
input[type="number"], input[type="date"], input[type="time"],
select, textarea {
  width: 100%;
  padding: 12px;
```

```

    border: 1px solid #ccc;
    border-radius: 4px;
}
button, .btn {
    background-color: #007bff;
    color: white;
    padding: 12px 20px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    text-align: center;
    text-decoration: none;
    font-size: 1rem;
    transition: background-color 0.3s;
    align-self: flex-start; /* Alinha o botão à esquerda */
}
button:hover, .btn:hover {
    background-color: #0056b3;
}
.btn-danger { background-color: #dc3545; }
.btn-secondary { background-color: #6c757d; }

/* --- TABELAS --- */
.table-wrapper {
    overflow-x: auto; /* Permite scroll horizontal em tabelas largas em mobile */
}
.responsive-table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
}
.responsive-table th, .responsive-table td {
    border: 1px solid #ddd;
    padding: 12px;
    text-align: left;
}
.responsive-table th {
    background-color: #f2f2f2;
    font-weight: bold;
}
.responsive-table tr:nth-child(even) { background-color: #f9f9f9; }
.responsive-table tr:hover { background-color: #f1f1f1; }

/* --- PÁGINA DE LOGIN --- */
.login-body {
    display: flex; justify-content: center; align-items: center;
    height: 100vh; background-color: #f4f7f6;
}
.login-container {
    background: white; padding: 40px; border-radius: 8px;
    box-shadow: 0 4px 15px rgba(0,0,0,0.1); text-align: center;

```



```

    width: 90%; max-width: 400px;
}

/* --- MENSAGENS E ALERTAS --- */
.success { color: #155724; background-color: #d4edda; border: 1px solid #c3e6cb; padding: 10px;
border-radius: 4px; }
.error { color: #721c24; background-color: #f8d7da; border: 1px solid #f5c6cb; padding: 10px;
border-radius: 4px; }

/* --- FOOTER --- */
footer {
    text-align: center;
    padding: 20px;
    margin-top: 30px;
    background-color: #e9ecef;
    border-top: 1px solid #dee2e6;
    color: #6c757d;
}

/* --- RESPONSIVIDADE --- */
@media (max-width: 820px) {
    .menu-toggle {
        display: block; /* Mostra o botão hamburger */
    }

    header nav {
        position: absolute;
        top: 70px; /* Altura do header */
        left: 0;
        width: 100%;
        background-color: #004a99; /* Fundo um pouco diferente para destacar */
        display: none; /* Escondido por padrão */
    }

    header nav.active {
        display: block; /* Mostra o menu quando a classe 'active' é adicionada */
    }

    header nav ul {
        flex-direction: column; /* Itens do menu em coluna */
        width: 100%;
        gap: 0;
    }

    header nav ul li {
        text-align: center;
        width: 100%;
        border-bottom: 1px solid rgba(255, 255, 255, 0.1);
    }

    header nav ul li a {

```

```

padding: 15px;
display: block;
width: 100%;
border-radius: 0;
}
}

```

## PASTA auth

### login.php

```

<?php
require_once '../config.php';

$error_message = '';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = $_POST['email'];
    $senha = $_POST['senha'];

    $sql = "SELECT id_usuario, nome, senha, perfil FROM usuarios WHERE email = :email AND status = 1";
    $stmt = $pdo->prepare($sql);
    $stmt->execute(['email' => $email]);
    $user = $stmt->fetch();

    if ($user && password_verify($senha, $user['senha'])) {
        $_SESSION['id_usuario'] = $user['id_usuario'];
        $_SESSION['nome'] = $user['nome'];
        $_SESSION['perfil'] = $user['perfil'];
        header("Location: /sistema_estoque/index.php");
        exit();
    } else {
        $error_message = "E-mail ou senha inválidos.";
    }
}
?>
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login - Sistema de Estoque</title>
    <link rel="stylesheet" href="../assets/css/style.css">
</head>

<body class="login-body">
    <div class="login-container">
        <h2>Sistema de Controle de Estoque</h2>
        <p>Escola Técnica de Enfermagem</p>
    </div>
</body>
</html>

```

```

<form action="login.php" method="post">
  <div class="form-group">
    <label for="email">E-mail</label>
    <input type="email" id="email" name="email" required>
  </div>
  <div class="form-group">
    <label for="senha">Senha</label>
    <input type="password" id="senha" name="senha" required>
  </div>
  <?php if ($error_message): ?>
    <p class="error"><?php echo $error_message; ?></p>
  <?php endif; ?>
  <button type="submit">Entrar</button>
</form>
</div>
</body>

</html>

```

### logout.php

```

<?php
session_start();
session_unset();
session_destroy();
header("Location: login.php");
exit();
?>

```

### PASTA requisitante:

#### criar\_solicitacao.php

```

<?php
require_once '../templates/header.php';
check_login('requisitante');

$success_message = "";
$error_message = "";

// Lógica de processamento do formulário quando enviado
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  // 1. Sanitização e validação dos dados de entrada
  $data_prevista = trim(filter_input(INPUT_POST, 'data_prevista'));
  $hora_prevista = trim(filter_input(INPUT_POST, 'hora_prevista'));
  $local_previsto = trim(filter_input(INPUT_POST, 'local_previsto', FILTER_SANITIZE_STRING));
  $observacoes = trim(filter_input(INPUT_POST, 'observacoes', FILTER_SANITIZE_STRING));
  $materiais_post = $_POST['materiais'] ?? [];
  $quantidades_post = $_POST['quantidades'] ?? [];

  // 2. LÓGICA DE AGREGAÇÃO PARA EVITAR ITENS DUPLICADOS

```

```

$itens_agregados = [];
if (is_array($materiais_post)) {
    foreach ($materiais_post as $index => $id_material) {
        $id_material = filter_var($id_material, FILTER_VALIDATE_INT);
        $quantidade = filter_var($quantidades_post[$index], FILTER_VALIDATE_INT);

        if ($id_material && $quantidade > 0) {
            if (isset($itens_agregados[$id_material])) {
                // Se o material já foi adicionado, apenas soma a nova quantidade
                $itens_agregados[$id_material] += $quantidade;
            } else {
                // Adiciona o material pela primeira vez
                $itens_agregados[$id_material] = $quantidade;
            }
        }
    }
}

// 3. Verifica se há itens válidos antes de prosseguir
if (empty($itens_agregados)) {
    $error_message = "Você deve adicionar pelo menos um item válido à solicitação.";
} else {
    try {
        // Inicia uma transação para garantir que tudo seja salvo corretamente
        $pdo->beginTransaction();

        // Insere a solicitação principal
        $sql_solicitacao = "INSERT INTO solicitacoes (id_usuario, data_prevista, hora_prevista,
local_previsto, observacoes) VALUES (?, ?, ?, ?, ?)";
        $stmt_solicitacao = $pdo->prepare($sql_solicitacao);
        $stmt_solicitacao->execute([$SESSION['id_usuario'], $data_prevista, $hora_prevista,
$local_previsto, $observacoes]);
        $id_solicitacao = $pdo->lastInsertId();

        // Insere os itens da solicitação (usando a lista agregada)
        $sql_item = "INSERT INTO solicitacao_itens (id_solicitacao, id_material, quantidade_solic)
VALUES (?, ?, ?)";
        $stmt_item = $pdo->prepare($sql_item);

        foreach ($itens_agregados as $id_material => $quantidade_total) {
            $stmt_item->execute([$id_solicitacao, $id_material, $quantidade_total]);
        }

        // Registra a ação no log de auditoria
        registrar_log($pdo, $SESSION['id_usuario'], 'Criação de Solicitação', "ID da Solicitação:
{$id_solicitacao}");

        // Confirma as alterações no banco de dados
        $pdo->commit();
        $success_message = "Solicitação nº {$id_solicitacao} criada com sucesso! <a
href='../ver_solicitacao.php?id={$id_solicitacao}'>Ver detalhes</a>";
    }
}

```

```

    } catch (Exception $e) {
        // Em caso de erro, desfaz todas as alterações
        $pdo->rollBack();
        $error_message = "Erro ao criar solicitação: " . $e->getMessage();
    }
}
}

```

```

// Busca os materiais disponíveis para preencher o formulário
$materiais_db = $pdo->query("SELECT id_material, nome, quantidade FROM materiais WHERE
status = 1 ORDER BY nome")->fetchAll(PDO::FETCH_ASSOC);
?>

```

<h2>Criar Nova Solicitação de Material</h2>

```

<?php if ($success_message): ?><p class="success"><?php echo $success_message; // Mensagem já
contém link, não precisa de htmlspecialchars ?></p><?php endif; ?>
<?php if ($error_message): ?><p class="error"><?php echo htmlspecialchars($error_message); ?>
</p><?php endif; ?>

```

```

<form action="criar_solicitacao.php" method="post" id="form-solicitacao">
    <div class="form-group">
        <label for="data_prevista">Data Prevista de Uso</label>
        <input type="date" id="data_prevista" name="data_prevista" required min="<?php echo
date('Y-m-d'); ?>">
    </div>
    <div class="form-group">
        <label for="hora_prevista">Hora Prevista de Uso</label>
        <input type="time" id="hora_prevista" name="hora_prevista" required>
    </div>
    <div class="form-group">
        <label for="local_previsto">Local de Uso</label>
        <input type="text" id="local_previsto" name="local_previsto" placeholder="Ex: Laboratório 1,
Sala 3" required>
    </div>
    <div class="form-group">
        <label for="observacoes">Observações</label>
        <textarea id="observacoes" name="observacoes" rows="3" placeholder="Qualquer informação
adicional relevante"></textarea>
    </div>

```

```

<h3>Itens da Solicitação</h3>
<div id="itens-container">
    <div class="item-row">
        <select name="materiais[]" class="material-select" required>
            <option value="">Selecione um material</option>
            <?php foreach ($materiais_db as $material): ?>
                <option value="<?php echo $material['id_material']; ?>"><?php echo
htmlspecialchars($material['nome']); ?></option>
            <?php endforeach; ?>

```

```

        </select>
        <input type="number" name="quantidades[]" class="quantidade-input" placeholder="Qtd."
required min="1" style="width: 100px;">
        <span class="stock-info"></span>
        <button type="button" class="btn btn-danger remove-item">Remover</button>
    </div>
</div>

<div style="display:flex; gap: 10px; margin-top: 10px;">
    <button type="button" id="add-item" class="btn btn-secondary">Adicionar Outro
Item</button>
</div>
<hr>
<div style="display:flex; gap: 10px;">
    <button type="submit" id="submit-btn">Enviar Solicitação</button>
    <a href="/sistema_estoque/requisitante/" class="btn btn-secondary">Cancelar</a>
</div>
</form>

```

```

<style>
.item-row { display: flex; gap: 10px; align-items: center; margin-bottom: 10px; flex-wrap: wrap; }
.item-row select { flex: 1; min-width: 200px; }
.stock-info { font-size: 0.9em; min-width: 150px; color: #6c757d; }
.item-row .btn-danger { padding: 8px 12px; }
</style>

```

```

<script>
document.addEventListener('DOMContentLoaded', function() {
    const itensContainer = document.getElementById('itens-container');
    const addItemBtn = document.getElementById('add-item');
    const submitBtn = document.getElementById('submit-btn');

    async function checkStock(materialSelect, quantidadeInput, stockInfoSpan) {
        const materialId = materialSelect.value;
        const quantidadeDesejada = parseInt(quantidadeInput.value, 10);

        if (!materialId || isNaN(quantidadeDesejada) || quantidadeDesejada <= 0) {
            stockInfoSpan.textContent = '';
            validateAllFields();
            return;
        }

        try {
            const response = await fetch(`/sistema_estoque/api/get_stock.php?id=${materialId}`);
            const data = await response.json();

            if (data.error) {
                stockInfoSpan.textContent = 'Erro ao verificar.'; stockInfoSpan.style.color = 'red';
            } else {
                const estoqueDisponivel = data.quantidade;
                stockInfoSpan.textContent = `Disponível: ${estoqueDisponivel}`;
            }
        }
    }

```

```

        if (quantidadeDesejada > estoqueDisponivel) {
            stockInfoSpan.textContent += ' - Insuficiente!';
            stockInfoSpan.style.color = 'red';
        } else {
            stockInfoSpan.style.color = 'green';
        }
    }
    validateAllFields();
} catch (error) {
    stockInfoSpan.textContent = 'Erro de conexão.'; stockInfoSpan.style.color = 'red';
    validateAllFields();
}
}

```

```

function validateAllFields() {
    let allValid = true;
    const selectedMaterials = new Set();

    document.querySelectorAll('.item-row').forEach(row => {
        const select = row.querySelector('.material-select');
        const stockInfoSpan = row.querySelector('.stock-info');

        // Valida estoque insuficiente
        if (stockInfoSpan.style.color === 'red') {
            allValid = false;
        }
        // Valida itens duplicados
        if (select.value) {
            if (selectedMaterials.has(select.value)) {
                allValid = false;
                select.style.borderColor = 'red';
            } else {
                selectedMaterials.add(select.value);
                select.style.borderColor = '#ccc';
            }
        }
    });
    submitBtn.disabled = !allValid;
}

```

```

itemsContainer.addEventListener('input', function(e) {
    const target = e.target;
    if (target.classList.contains('quantidade-input') || target.classList.contains('material-select')) {
        const itemRow = target.closest('.item-row');
        const materialSelect = itemRow.querySelector('.material-select');
        const quantidadeInput = itemRow.querySelector('.quantidade-input');
        const stockInfoSpan = itemRow.querySelector('.stock-info');
        checkStock(materialSelect, quantidadeInput, stockInfoSpan);
        if (target.classList.contains('material-select')) {
            validateAllFields(); // Valida duplicidade ao mudar a seleção
        }
    }
}

```

```

    }
  });

  itensContainer.addEventListener('click', function(e) {
    if (e.target.classList.contains('remove-item')) {
      if (itensContainer.children.length > 1) {
        e.target.closest('.item-row').remove();
      } else {
        const firstRow = itensContainer.querySelector('.item-row');
        firstRow.querySelector('select').selectedIndex = 0;
        firstRow.querySelector('input').value = "";
        firstRow.querySelector('.stock-info').textContent = "";
        firstRow.querySelector('.stock-info').style.color = "";
      }
      validateAllFields();
    }
  });

  addItemBtn.addEventListener('click', function() {
    const firstItemRow = itensContainer.firstElementChild;
    const newItemRow = firstItemRow.cloneNode(true);
    newItemRow.querySelector('select').selectedIndex = 0;
    newItemRow.querySelector('input').value = "";
    newItemRow.querySelector('.stock-info').textContent = "";
    newItemRow.querySelector('.stock-info').style.color = "";
    newItemRow.querySelector('.material-select').style.borderColor = '#ccc';
    itensContainer.appendChild(newItemRow);
    validateAllFields();
  });

  validateAllFields(); // Executa uma validação inicial
});
</script>

```

```

<?php
require_once '../templates/footer.php';
?>

```

### historico.php

```

<?php
require_once '../templates/header.php';
check_login('requisitante');

// Buscar o histórico de solicitações do usuário logado
$id_usuario = $_SESSION['id_usuario'];
$stmt = $pdo->prepare("
  SELECT id_solicitacao, data_prevista, status, criado_em
  FROM solicitacoes
  WHERE id_usuario = ?
  ORDER BY criado_em DESC

```



```

");
$stmt->execute([$id_usuario]);
$solicitacoes = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>

<h2>Minhas Solicitações</h2>

<?php if (empty($solicitacoes)): ?>
    <p>Você ainda não fez nenhuma solicitação.</p>
    <a href="criar_solicitacao.php" class="btn">Criar minha primeira solicitação</a>
<?php else: ?>
    <div class="table-wrapper">
        <table class="responsive-table">
            <thead>
                <tr>
                    <th>Nº da Solicitação</th>
                    <th>Data de Criação</th>
                    <th>Data Prevista de Uso</th>
                    <th>Status</th>
                    <th>Ações</th> <!-- NOVA COLUNA -->
                </tr>
            </thead>
            <tbody>
                <?php foreach ($solicitacoes as $s): ?>
                    <tr>
                        <td><?php echo $s['id_solicitacao']; ?></td>
                        <td><?php echo date('d/m/Y H:i', strtotime($s['criado_em'])); ?></td>
                        <td><?php echo date('d/m/Y', strtotime($s['data_prevista'])); ?></td>
                        <td><strong class="status-badge status-<?php echo htmlspecialchars($s['status']); ?>">
<?php echo htmlspecialchars(ucfirst($s['status'])); ?></strong></td>
                        <td>
                            <!-- **NOVO LINK AQUI** -->
                            <a href=" ../ver_solicitacao.php?id=<?php echo $s['id_solicitacao']; ?>" class="btn">Ver
Detalhes</a>
                        </td>
                    </tr>
                <?php endforeach; ?>
            </tbody>
        </table>
    </div>
<?php endif; ?>

<?php require_once '../templates/footer.php'; ?>

```

## index.php

```

<?php
require_once '../templates/header.php';
check_login('requisitante');
?>

```

```
<h2>Painel do Requisitante</h2>
<p>Bem-vindo(a), <?php echo htmlspecialchars($_SESSION['nome']); ?>!</p>
<p>Use os links abaixo para gerenciar suas solicitações de materiais.</p>
```

```
<div class="dashboard-actions">
  <a href="criar_solicitacao.php" class="action-card">
    <h3>Criar Nova Solicitação</h3>
    <p>Peça os materiais necessários para suas atividades.</p>
  </a>
  <a href="historico.php" class="action-card">
    <h3>Minhas Solicitações</h3>
    <p>Consulte o status e o histórico de todos os seus pedidos.</p>
  </a>
</div>
```

```
<style>
.dashboard-actions {
  margin-top: 30px;
  display: flex;
  gap: 20px;
}
.action-card {
  background-color: #e9f7ff;
  border: 1px solid #b3e0ff;
  border-radius: 8px;
  padding: 20px;
  text-decoration: none;
  color: #333;
  flex: 1;
  transition: transform 0.2s, box-shadow 0.2s;
}
.action-card:hover {
  transform: translateY(-5px);
  box-shadow: 0 4px 12px rgba(0,0,0,0.1);
}
.action-card h3 {
  margin-top: 0;
  color: #0056b3;
}
</style>
```

```
<?php require_once '../templates/footer.php'; ?>
```

### **PASTA templates:**

#### **footer.php**

```
</div>
</main>
<footer>
<p>© <?php echo date("Y"); ?> - Escola Técnica de Enfermagem. Todos os direitos reservados.</p>
```

```

</footer>

<!-- SCRIPT PARA O MENU RESPONSIVO -->
<script>
    document.addEventListener('DOMContentLoaded', function() {
        const menuToggle = document.querySelector('.menu-toggle');
        const mainNav = document.querySelector('#main-nav');

        if (menuToggle && mainNav) {
            menuToggle.addEventListener('click', function() {
                mainNav.classList.toggle('active');
            });
        }
    });
</script>
</body>
</html>

```

### header.php

```

<?php require_once __DIR__ . '/../config.php'; ?>
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sistema de Estoque</title>
    <link rel="stylesheet" href="/sistema_estoque/assets/css/style.css">
</head>
<body>
    <header>
        <div class="header-container">
            <h1><a href="/sistema_estoque/">Sistema de Estoque</a></h1>

            <!-- Botão Hamburger para Mobile -->
            <button class="menu-toggle" aria-label="Abrir menu">≡</button>

            <!-- Adicionado ID "main-nav" para o JS -->
            <nav id="main-nav">
                <ul>
                    <?php if (isset($_SESSION['perfil'])): ?>
                        <?php if ($_SESSION['perfil'] === 'admin'): ?>
                            <li><a href="/sistema_estoque/admin/">Painel</a></li>
                            <li><a href="/sistema_estoque/admin/gerenciar_solicitacoes.php">Solicitações</a>
                        </li>

                            <li><a href="/sistema_estoque/admin/gerenciar_materiais.php">Materiais</a></li>
                            <li><a href="/sistema_estoque/admin/gerenciar_usuarios.php">Usuários</a></li>
                            <li><a href="/sistema_estoque/admin/relatorios.php">Relatórios</a></li>
                        <?php else: ?>
                            <li><a href="/sistema_estoque/requisitante/">Painel</a></li>
                            <li><a href="/sistema_estoque/requisitante/criar_solicitacao.php">Nova

```

```
Solicitação</a></li>
    <li><a href="/sistema_estoque/requisitante/historico.php">Minhas Solicitações</a>
</li>
    <?php endif; ?>
    <li><a href="/sistema_estoque/auth/logout.php">Sair</a></li>
    <?php endif; ?>
</ul>
</nav>
</div>
</header>
<main>
    <div class="container">
```