

Estrutura de diretórios no Windows/XAMPP

text

CopiarEditar

C:\xampp\htdocs\CSMJ\ ← raiz do seu site no Apache

├─ api\ ← back-end em PHP

| └─ config.php ← conexão PDO + session_start()

| └─ login.php ← autenticação (POST JSON → token)

| └─ news_add.php ← criação de notícias (FormData → upload + DB)

| └─ upload.php ← upload de arquivos de transparência

└─ transparencia\ ← pasta onde ficam os arquivos enviados

└─ ADM\ ← build estático React (após npm run build)

| └─ index.html

| └─ asset-manifest.json

| └─ manifest.json

| └─ static\ ← js, css e media gerados pelo CRA

└─ img\ ← suas imagens públicas

└─ index.html ← páginas estáticas do site público

└─ sobre.html

└─ politica.html

└─ projects.html

└─ donation.html

└─ menu.js

└─ script.js

└─ styles.css

- **api/**: todos os arquivos .php que expõem endpoints (/CSMJ/api/*.php)
- **transparencia/**: conteúdo de uploads do admin
- **ADM/**: resultado do npm run build do painel React
- O restante: seu site público em HTML/JS/CSS

2. Configuração do Banco de Dados (MySQL)

- Banco: **csmj_admin**
- Tabela **users**:

sql

CopiarEditar

```
CREATE TABLE users (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  username VARCHAR(50) UNIQUE NOT NULL,  
  password_hash VARCHAR(255) NOT NULL  
);
```

- Tabela **news** (após ALTER):

sql

CopiarEditar

```
CREATE TABLE news (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  title VARCHAR(150) NOT NULL,  
  subtitle VARCHAR(150) NULL,  
  body TEXT NOT NULL,  
  banner_path VARCHAR(255) NULL,  
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

- Tabela **files** (opcional, para upload de transparência):

sql

CopiarEditar

```
CREATE TABLE files (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  filename VARCHAR(255) NOT NULL,  
  uploaded_at DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

- Para gerar hash de senha: crie um hash.php temporário com `echo password_hash('senha', PASSWORD_DEFAULT);` e copie o valor para `password_hash` no phpMyAdmin.
-

3. Back-end PHP (pasta api/)

config.php

php

CopiarEditar

<?php

// config.php — conexão PDO

\$dbHost = 'localhost';

\$dbName = 'csmj_admin';

\$dbUser = 'root';

\$dbPass = '';

try {

 \$pdo = new PDO("mysql:host=\$dbHost;dbname=\$dbName;charset=utf8mb4", \$dbUser,
 \$dbPass, [

 PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,

 PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,

]);

} catch (PDOException \$e) {

 http_response_code(500);

 echo json_encode(['error'=>'Erro de conexão']);

 exit;

}

session_start();

login.php

php

CopiarEditar

<?php

// CORS + preflight

header('Access-Control-Allow-Origin: http://localhost:3000');

header('Access-Control-Allow-Credentials: true');

header('Access-Control-Allow-Headers: Content-Type, Authorization');

header('Access-Control-Allow-Methods: POST, OPTIONS');

```

if ($_SERVER['REQUEST_METHOD'] === 'OPTIONS') { http_response_code(204); exit; }

header('Content-Type: application/json; charset=utf-8');

require 'config.php';

$input = json_decode(file_get_contents('php://input'), true);

if (!isset($input['username'], $input['password'])) {
    http_response_code(400); echo json_encode(['error' => 'Requisição inválida']); exit;
}

$stmt = $pdo->prepare("SELECT id, password_hash FROM users WHERE username=?");
$stmt->execute([$input['username']]);

$user = $stmt->fetch();

if (!$user || !password_verify($input['password'], $user['password_hash'])) {
    http_response_code(401); echo json_encode(['error' => 'Credenciais inválidas']); exit;
}

// gerar token simples
$token = bin2hex(random_bytes(16));
echo json_encode(['token' => $token]);

```

news_add.php

php

CopiarEditar

<?php

```

// CORS + preflight (mesmo de login)

header('Access-Control-Allow-Origin: http://localhost:3000');

header('Access-Control-Allow-Credentials: true');

header('Access-Control-Allow-Headers: Content-Type, Authorization');

header('Access-Control-Allow-Methods: POST, OPTIONS');

if ($_SERVER['REQUEST_METHOD'] === 'OPTIONS') { http_response_code(204); exit; }

header('Content-Type: application/json; charset=utf-8');

require 'config.php';

if (!isset($_FILES['banner']) || $_FILES['banner']['error'] !== UPLOAD_ERR_OK) {

```

```
http_response_code(400); echo json_encode(['error'=>'Banner inválido']); exit;
}
```

```
$title = $_POST['title'] ?? '';
$subtitle = $_POST['subtitle'] ?? '';
$body = $_POST['body'] ?? '';
if (!$title || !$body) {
    http_response_code(400); echo json_encode(['error'=>'Título e corpo obrigatórios']); exit;
}
```

```
$uploadDir = __DIR__.'../transparencia/';
$filename = uniqid('news_').'_'.basename($_FILES['banner']['name']);
move_uploaded_file($_FILES['banner']['tmp_name'], $uploadDir.$filename);
```

```
$stmt = $pdo->prepare("
    INSERT INTO news(title,subtitle,body,banner_path,created_at)
    VALUES(?,?,?,?,NOW())
");
$stmt->execute([$title,$subtitle,$body,$filename]);
```

```
echo json_encode(['success'=>true]);
```

upload.php

php

CopiarEditar

<?php

```
// CORS + preflight
```

```
header('Access-Control-Allow-Origin: http://localhost:3000');
```

```
header('Access-Control-Allow-Credentials: true');
```

```
header('Access-Control-Allow-Headers: Content-Type, Authorization');
```

```
header('Access-Control-Allow-Methods: POST, OPTIONS');
```

```
if ($_SERVER['REQUEST_METHOD']==='OPTIONS'){http_response_code(204);exit;}
```

```

header('Content-Type: application/json; charset=utf-8');

require 'config.php';

if (!isset($_FILES['file'])) {
    http_response_code(400); echo json_encode(['error'=>'Nenhum arquivo']); exit;
}

$dir = __DIR__.'../transparencia/';
$name=basename($_FILES['file']['name']);
move_uploaded_file($_FILES['file']['tmp_name'],$dir.$name);

$stmt = $pdo->prepare("INSERT INTO files(filename) VALUES(?)");
$stmt->execute([$name]);

echo json_encode(['success'=>true]);

```

4. Front-end React (projeto csmj-admin/)

Estrutura

pgsql

CopiarEditar

csmj-admin/

└─ public/ ← index.html dev, favicon, manifest

└─ src/

| └─ services/

| | └─ auth.js ← login()

| | └─ news.js ← createNews()

| | └─ upload.js ← uploadFile()

| └─ pages/

| | └─ Login.jsx

| | └─ Dashboard.jsx

| | └─ NewsAdd.jsx

| | └─ Upload.jsx

| | — ProtectedRoute.jsx

| | — App.js

| — package.json

| — ...

Principais arquivos

src/services/auth.js

js

CopiarEditar

```
const API = 'http://localhost/CSMJ/api';

export async function login(u,p){

  const res = await fetch(`${API}/login.php`, { /* POST JSON */ });

  if(!res.ok) throw new Error('Login falhou');

  return res.json();

}
```

src/services/news.js

js

CopiarEditar

```
export async function createNews({title,subtitle,body,banner}){

  const form = new FormData();

  form.append('banner',banner);

  form.append('title',title);

  form.append('subtitle',subtitle);

  form.append('body',body);

  const res = await fetch(`${API}/news_add.php`,{

    method:'POST',

    headers:{ Authorization:`Bearer ${token}` },

    body: form

  });

  if(!res.ok) throw new Error(await res.text());

  return res.json();

}
```

src/services/upload.js

js

CopiarEditar

```
export async function uploadFile(file){  
  const form = new FormData();  
  form.append('file',file);  
  const res = await fetch(`${API}/upload.php`,{/*...*/});  
  if(!res.ok) throw new Error(await res.text());  
  return res.json();  
}
```

src/ProtectedRoute.jsx

jsx

CopiarEditar

```
export default function ProtectedRoute({children}){  
  return localStorage.getItem('token')  
    ? children  
    : <Navigate to="/login" replace />;  
}
```

src/App.js

jsx

CopiarEditar

```
<HashRouter>  
  <Routes>  
    <Route path="/login" element={<Login/>}/>  
    <Route path="/dashboard" element={<ProtectedRoute><Dashboard/></ProtectedRoute>}/>  
    <Route path="/news/add" element={<ProtectedRoute><NewsAdd/></ProtectedRoute>}/>  
    <Route path="/upload" element={<ProtectedRoute><Upload/></ProtectedRoute>}/>  
    <Route path="*" element={  
      localStorage.getItem('token')  
        ? <Navigate to="/dashboard" replace/>  
        : <Navigate to="/login" replace/>  
    }/>  
  </Routes>  
</HashRouter>
```

```
}>
```

```
</Routes>
```

```
</HashRouter>
```

src/pages/Login.jsx

- Formulário de usuário/senha
- Chama apiLogin(user,pass)
- Salva token em localStorage
- Redireciona p/ /dashboard

src/pages/Dashboard.jsx

- Botão **Logout** (remove token + navigate /login)
- Links para **Nova Notícia** e **Upload Transparência**

src/pages/NewsAdd.jsx

- Form com <input type="file"> para banner
- Inputs para title, subtitle, body
- Chama createNews({...}) e redireciona a /dashboard

src/pages/Upload.jsx

- Form com <input type="file">
- Chama uploadFile(file) e mostra mensagem

5. Fluxo de trabalho

1. Dev

bash

CopiarEditar

cd csmj-admin

npm install

npm start # abre em localhost:3000/#/login

2. Testes

- Login → Dashboard → Nova Notícia → Upload → Logout
- Verificar Network / phpMyAdmin / pasta transparencia/

3. Build & deploy local

bash

CopiarEditar

npm run build

Copiar o conteúdo de **csmj-admin/build/** para

C:\xampp\htdocs\CSMJ\ADM

Acessar no navegador:

bash

CopiarEditar

<http://localhost/CSMJ/ADM/index.html#/login>

Recapitulação do Projeto Site CSMJ - ADM e API 1. Estrutura de Diretórios (XAMPP):

C:\xampp\htdocs\CSMJ\api\ (PHP endpoints: config.php, login.php, news_add.php, upload.php) transparencia\ (arquivos enviados) ADM\ (build React estático) site público (HTML, JS, CSS) 2. Banco de Dados MySQL:- Banco: csmj_admin- Tabela users (id, username, password_hash)- Tabela news (id, title, subtitle, body, banner_path, created_at)- Tabela files (id, filename, uploaded_at) 3. Back-end PHP (api/):- config.php: conexão PDO e session_start()- login.php: CORS + autenticação JSON → token- news_add.php: upload de banner + FormData → grava no DB- upload.php: upload de transparência → grava no DB 4. Front-end React (csmj-admin/):- services/auth.js: login()- services/news.js: createNews() com FormData- services/upload.js: uploadFile() com FormData- pages/Login.jsx: formulário login- pages/Dashboard.jsx: painel ADM e links- pages/NewsAdd.jsx: form de nova notícia- pages/Upload.jsx: form de upload transparência- ProtectedRoute.jsx: rota protegida- App.js: roteamento com HashRouter e ProtectedRoute 5. Fluxo de Trabalho:- Desenvolvimento: npm start em csmj-admin/- Testes: login → dashboard → news add → upload → logout- Build: npm run build → copiar build/ para htdocs/CSMJ/ADM/- Acesso produção local: <http://localhost/CSMJ/ADM/index.html#/login> Guarde este documento para futuras consultas