

Couple Sync — Apresentação

Data: 06/11/2025 Autor: Levy Tavares

Resumo

Couple Sync é uma aplicação full-stack para guardar memórias em fotos, feita com React (Vite) no frontend e Express.js no backend. As imagens são armazenadas no Cloudinary e os metadados em PostgreSQL (Neon). O foco é uma UX caprichada, segura e simples para casais.

Objetivos

- Upload de fotos com descrição e data
- Autenticação (JWT)
- Edição e exclusão com confirmação
- Favoritos e filtro rápido
- Visualização em tela cheia (Lightbox)
- Tutorial embutido para onboarding

Arquitetura

```
Cliente (React + Vite)
  ↓ HTTP/JSON (Bearer Token)
API (Express.js)
  ↔ PostgreSQL (Neon)      → metadados das fotos
  ↔ Cloudinary              → armazenamento de imagens/CDN
```

Fluxo de Upload (alto nível)

1. Usuário escolhe a imagem e preenche descrição/data.
2. Front envia **multipart/form-data** (imageFile, description, photoDate) com **Authorization: Bearer <token>**.
3. API valida JWT, faz upload para o Cloudinary e grava metadados (image_url, description, photo_date) no Neon.
4. API responde com a foto criada (image_url pronto para renderização via CDN).

Stack Principal

- Frontend: React, React Router, React Toastify, SASS (SCSS), React Icons, Vite
- Backend: Node.js, Express.js, Multer (memoryStorage), Cloudinary SDK, pg, bcrypt, JWT, dotenv, CORS
- Banco: PostgreSQL (Neon)
- Armazenamento de Imagens: Cloudinary

Funcionalidades de Destaque

- Tema escuro com variáveis globais e componentes acessíveis

- FAB com gradiente, animação e tooltip
- Lightbox com ESC/Setas e botão de download
- Favoritos persistidos em `localStorage` (hook `useFavorites`)
- Confete discreto no primeiro upload do dia (import dinâmico)
- Página de Tutorial com passos ilustrados

Endpoints Principais (prefixo /api)

Método	Rota	Auth	Descrição
POST	/register	-	Cria usuário
POST	/login	-	Retorna JWT + usuário
GET	/fotos	✓	Lista fotos do usuário
POST	/upload	✓	Upload multipart + Cloudinary
PUT	/fotos/:id	✓	Atualiza descrição/data
DELETE	/fotos/:id	✓	Remove registro e imagem

Variáveis de Ambiente (exemplos)

Backend (`api/.env`):

```
PORT=4000
DATABASE_URL=postgres://user:password@host:port/dbname
JWT_SECRET=defina-uma-chave-segura
CLOUDINARY_CLOUD_NAME=xxxxxxxxx
CLOUDINARY_API_KEY=xxxxxxxxx
CLOUDINARY_API_SECRET=xxxxxxxxx
```

Frontend (`client/.env`):

```
VITE_API_BASE_URL=http://localhost:4000/api
```

Como Rodar (Dev)

```
# Backend
cd api
npm install
npm run dev

# Frontend
cd client
npm install
npm run dev
```

Acesse: <http://localhost:5173>

Prints (opcional)

- Login e Registro com inputs customizados
- Galeria com edição, exclusão e favoritos
- Lightbox e FAB de upload

Observações de Segurança

- MVP acadêmico com JWT simples; produção recomenda refresh tokens e cookies httpOnly
- Upload validado no backend; imagens nunca são salvas no banco, apenas URLs
- SSL habilitado para conexão com Neon (ajustes de `rejectUnauthorized` em dev)

Próximos Passos (sugestões)

- Testes automatizados (unitários e de integração)
- CI/CD simples (GitHub Actions)
- Páginas de álbuns/coleções
- Compartilhamento seguro de links (assinados)