

📸 Guia Visual - Setup do Sistema de Gestão de Metas
Este documento mostra visualmente cada passo do setup.

1 Fluxo Geral de

Setup

SETUP DO PROJETO

1

Criar Conta GitHub
2. Criar Repositório GitHub
3. Configurar Git Localmente
4. Criar Conta Firebase
5. Configurar Firebase (Auth, Firestore, Storage)
6. Criar Conta Vercel
7. Conectar GitHub com Vercel
8. Configurar Banco de Dados (PlanetScale ou MySQL)
9. Configurar Secrets no GitHub
10. Fazer Primeiro Deploy
PRONTO PARA USAR!

2 Arquitetura do

Projeto

SEU COMPUTADOR

```
Terminal / PowerShell
$ git clone ...
$ pnpm install
$ pnpm dev
→ http://localhost:3000
```

git

push

GITHUB Repositório: metas-management-system -

Código fonte | \n | - GitHub Actions (CI/CD) | \n | -
Secrets (DATABASE_URL, JWT_SECRET, etc)

| \n |

↓

webhook \n |

VERCEL | \n | - Build automático | \n | - Deploy em
produção | \n | - URL: seu-projeto.vercel.app

| \n |

↓ \n |

FIREBASE | \n | - Autenticação (Email/Password) | \n | -
Firestore Database | \n | - Cloud Storage

| \n |

↓ \n |

BANCO DE DADOS | \n | PlanetScale MySQL | \n | - Tabelas:
users, sectors, goals, results, approvals

| \n |

— \n \n ## 3 Fluxo de Desenvolvimento \n \n \n Você faz uma
mudança \n ↓ \n \$ git add . \n \$ git commit -m

"Descrição" \n \$ git push origin main \n ↓ \n GitHub

recebe o push \n ↓ \n GitHub Actions executa: \n -

Testes \n - Linting \n - Build \n ↓ \n Se tudo OK → Vercel

faz deploy automático \n ↓ \n  Seu projeto está

atualizado em produção! \n \n — \n \n ## 4 Estrutura de

Pastas \n \n \n metas-management-system / \n | \n | — client / #

Frontend (React) \n | | — src / \n | | | — pages / # Páginas

da aplicação \n | | | — components / # Componentes

```
reutilizáveis\n| | | — App.tsx # Roteamento
principal\n| | | — main.tsx # Entry point\n| | —
public/ # Arquivos estáticos\n|\n| — server/ # Backend
(Express + tRPC)\n| | — routers.ts # APIs (tRPC)\n| | —
db.ts # Funções de banco de dados\n| | — _core/ #
Framework (não editar)\n|\n| — drizzle/ # Banco de
Dados\n| | — schema.ts # Definição de tabelas\n| | —
*.sql # Migrações\n|\n| — .github/\n| | — workflows/ #
GitHub Actions\n| | — deploy.yml # Deploy automático\n|
| — lint.yml # Validação de código\n|\n| — .env.example
# Exemplo de variáveis\n| — vercel.json # Configuração
Vercel\n| — docker-compose.yml # Banco de dados
local\n|\n| — SETUP_PASSO_A_PASSO.md # Este guia!\n| —
DEVELOPMENT.md # Desenvolvimento local\n| —
FIREBASE_SETUP.md # Configuração Firebase\n| —
GITHUB_SETUP.md # Configuração GitHub\n| —
DEPLOYMENT.md # Deploy na Vercel\n| — README_SETUP.md #
Visão geral\n|\n|\n| —\n|\n|### 5 Tabelas do Banco de
Dados\n|\n|\n|
```

BANCO DE DADOS

```
|\n|
|\n| users (Usuários) |\n| | — id (chave primária) |\n|
| — openId (ID do OAuth) |\n| | — name (Nome) |\n| | —
email (Email) |\n| | — role (admin ou user) |\n| | —
createdAt (Data de criação) |\n| |\n| sectors
(Setores/Departamentos) |\n| | — id (chave primária)
```

```

|\n| |— name (Nome do setor) |\n| |— description
(Descrição) |\n| |— driveLink (Link para pasta do
Drive) |\n| |— createdBy (ID do usuário que criou)
|\n| |\n| goals (Metas) |\n| |— id (chave primária)
|\n| |— sectorId (ID do setor) |\n| |— name (Nome da
meta) |\n| |— type (% / inteiro / data) |\n| |—
weight (Peso da meta) |\n| |— targetValue (Valor alvo)
|\n| |\n| results (Resultados Apontados) |\n| |— id
(chave primária) |\n| |— goalId (ID da meta) |\n| |—
userId (ID do usuário) |\n| |— value (Valor apontado)
|\n| |— status (draft/submitted/approved/rejected)
|\n| |— evidenceLinks (Links para evidências) |\n|
|\n| approvals (Aprovações) |\n| |— id (chave
primária) |\n| |— resultId (ID do resultado) |\n| |—
approverId (ID do aprovador) |\n| |— status (approved
/ rejected) |\n| |— comments (Comentários) |\n| |\n|
auditLog (Auditoria) |\n| |— id (chave primária) |\n|
|— userId (ID do usuário) |\n| |— action (Ação
realizada) |\n| |— createdAt (Data/hora) |\n|
|\n|

```

—\n\n## 6 Fluxo de Usuário no

Sistema\n\n\n

FLUXO DE USUÁRIO

```

|\n|

```

ADMIN CADASTRA ESTRUTURA\n |— Cria setores (ex:

Vendas, RH, TI)\n |— Define responsáveis por setor\n

└─ Adiciona link do Google Drive para evidências\n\n2.

ADMIN CADASTRA METAS\n └─ Define metas para cada
setor\n └─ Define tipo (% , inteiro ou data)\n └─
Define peso (soma não pode ultrapassar 100%)\n └─
Define período (ex: Janeiro 2024)\n\n3. COLABORADOR
APONTA RESULTADO\n └─ Acessa sua meta\n └─ Insere
valor alcançado\n └─ Anexa evidências (fotos,
planilhas, etc)\n └─ Clica em \ "Enviar para
Aprovação\ "\n\n4. ADMIN REVISA E APROVA\n └─ Vê todos
os resultados enviados\n └─ Analisa evidências\n └─
Aprova ou rejeita\n └─ Deixa comentários se
necessário\n\n5. DASHBOARD CONSOLIDA\n └─ Mostra
status por setor\n └─ Gráficos de progresso\n └─
Relatórios por período\n └─ Histórico de
aprovações\n\n\n─\n\n## 7 Variáveis de
Ambiente\n\n\n.env.local (seu computador)\n└─
DATABASE_URL = mysql://user:pass@host/db\n└─
JWT_SECRET = chave-secreta-longa\n└─ VITE_APP_ID =
seu-app-id\n└─ VITE_OAUTH_PORTAL_URL =
https://oauth.manus.im\n\nGitHub Secrets
(repositório)\n└─ DATABASE_URL\n└─ JWT_SECRET\n└─
VERCEL_TOKEN\n└─ VERCEL_ORG_ID\n└─
VERCEL_PROJECT_ID\n\n\nVercel Environment Variables
(produção)\n└─ DATABASE_URL\n└─ JWT_SECRET\n└─
VITE_APP_ID\n└─ VITE_OAUTH_PORTAL_URL\n\n\n─\n\n\n## 8
Comandos Principais\n\n\nbash\n# Desenvolvimento\nnpm

dev # Inicia servidor local\npnpm build # Build para produção\npnpm start # Executa build em produção\n\n# Banco de Dados\npnpm db:push # Gera e aplica migrações\npnpm db:studio # Abre interface web do banco\n\n# Testes e Validação\npnpm test # Executa testes\npnpm check # Verifica tipos TypeScript\npnpm format # Formata código\npnpm format --check # Verifica formatação\n\n# Git\ngit clone <url> # Clona repositório\ngit add . # Adiciona arquivos\ngit commit -m \"msg\" # Faz commit\ngit push origin main # Envia para GitHub\n\n—\n\n## 9 Checklist de Setup\n\n☐ Node.js instalado (v18+)\n☐ Git instalado\n☐ Conta GitHub criada\n☐ Repositório GitHub criado\n☐ Git configurado localmente\n☐ SSH key adicionada no GitHub\n☐ Conta Firebase criada\n☐ Firebase configurado (Auth, Firestore, Storage)\n☐ Conta Vercel criada\n☐ Vercel conectado com GitHub\n☐ Banco de dados criado (PlanetScale ou MySQL)\n☐ Secrets configurados no GitHub\n☐ Repositório clonado localmente\n☐ Dependências instaladas (pnpm install)\n☐ Migrações executadas (pnpm db:push)\n☐ Servidor rodando localmente (pnpm dev)\n☐ Primeiro commit feito (git push)\n☐ Deploy automático na Vercel ☒

PRONTO!\n\n—\n\n## 10 Diagrama de Fluxo de Deploy\n\nVocê faz mudança no código\n↓\n\$ git push origin main\n↓\n_____

GitHub recebe o push | \n | Dispara GitHub Actions

| \n | _____ | \n

↓ \n | _____ | \n | GitHub

Actions: | \n | 1. npm install | \n | 2. npm run test | \n |

3. npm run check | \n | 4. npm run format --check | \n | 5.

npm run build | \n | _____ | \n

↓ \n Tudo OK? \n / \ \ \n SIM NÃO \n ↓ ↓ \n   Falha \n | |

(Email enviado) \n | |  Verifique logs \n

↓ \n | _____ | \n | Vercel recebe

sinal | \n | Inicia novo deployment

| \n | _____ | \n

↓ \n | _____ | \n | Vercel: | \n |

1. Build | \n | 2. Testa | \n | 3. Deploy

| \n | _____ | \n ↓ \n  Seu

projeto está em produção! \n URL: seu-

projeto.vercel.app \n \n \n — \n \n ##  Acessando o

Projeto \n \n \n Desenvolvimento Local: \n →

http://localhost:3000 \n \n Produção (Vercel): \n →

https://seu-projeto.vercel.app \n \n Banco de Dados

(Drizzle Studio): \n → http://localhost:5555 \n \n Painel de

Controle: \n → GitHub: https://github.com/seu-

usuario/metasp-management-system \n → Vercel:

https://vercel.com/seu-usuario/metasp-management-

system \n → Firebase:

https://console.firebase.google.com \n → PlanetScale:

https://app.planetscale.com \n \n \n — \n \n Próximo passo:

Leia o arquivo `SETUP_PASSO_A_PASSO.md` para instruções detalhadas!\n
