# Manual Técnico do Projeto MAGIC – Versão 8 (COMPLETO)

## 1. Introdução ao Projeto MAGIC

Objetivo: Apresentar o propósito e escopo do sistema de gerenciamento do hostal MAGIC.

Como utilizar os arquivos: Todos os arquivos do projeto participam do escopo geral.

Passo a passo:  
1) Entender o fluxo geral: reserva, serviços, chatbot, análise. 2) Ler a arquitetura proposta. 3) Estudar os módulos individualmente antes de integrar.

## 2. Arquitetura Geral

Objetivo: Descrever a arquitetura baseada em AWS e integrações web do projeto MAGIC.

Como utilizar os arquivos: Diagramas e estrutura de pastas do repositório.

Passo a passo:  
1) Identificar os componentes: frontend, backend, APIs, banco, Lambda, QuickSight. 2) Seguir as seções para implementar cada um separadamente. 3) Integrar como descrito na seção 22.

## 3. Backend (app.py)

Objetivo: Prover endpoints REST para serviços, reservas e integrações.

Como utilizar os arquivos: app.py

Passo a passo:  
1) Instalar dependências (Flask ou FastAPI). 2) Executar app.py localmente ou via Lambda. 3) Validar rotas /servicos, /reserva etc.

## 4. Banco de Dados (RDS / DynamoDB)

Objetivo: Criar e configurar a base de dados do hostal.

Como utilizar os arquivos: terraform.yml, terraform\_glue\_hostal.tf

Passo a passo:  
1) Acesse o AWS RDS Console. 2) Crie instância PostgreSQL. 3) Crie tabelas DynamoDB via console ou API. 4) Configure segurança e conecte ao app.py.

## 5. Serviços REST com Lambda + API Gateway

Objetivo: Expor APIs REST com baixo custo via AWS Lambda.

Como utilizar os arquivos: lambda\_api\_payload.zip

Passo a passo:  
1) Crie uma função Lambda no AWS Console. 2) Faça upload do ZIP com lambda\_function.py. 3) Vincule à API Gateway com método GET/POST.

## 6. Infraestrutura com Terraform

Objetivo: Automatizar a criação da infraestrutura da AWS.

Como utilizar os arquivos: terraform.yml, terraform\_glue\_hostal.tf, terraform\_step\_function\_hostal.tf

Passo a passo:  
1) Instale Terraform. 2) Execute `terraform init`, `terraform plan`, `terraform apply`. 3) Recursos como Glue, Step Functions e banco serão criados.

## 7. Infraestrutura com CloudFormation

Objetivo: Prover alternativa declarativa de infraestrutura.

Como utilizar os arquivos: cloudformation\_glue\_hostal.yaml

Passo a passo:  
1) Acesse AWS CloudFormation Console. 2) Crie novo stack. 3) Faça upload do YAML. 4) Verifique recursos criados.

## 8. Infraestrutura com AWS CDK

Objetivo: Utilizar código Python para definir infraestrutura AWS.

Como utilizar os arquivos: hostal\_stack.py

Passo a passo:  
1) Instale AWS CDK. 2) Execute `cdk init`, `cdk deploy`. 3) Confirme criação de Glue, Step Functions e SageMaker.

## 9. Step Functions (orquestração)

Objetivo: Controlar a sequência entre Glue, SageMaker e outros jobs.

Como utilizar os arquivos: step\_function\_pipeline\_hostal.json

Passo a passo:  
1) Acesse AWS Step Functions. 2) Importe o JSON como nova máquina de estados. 3) Conecte a Glue Job e endpoints.

## 10. AWS Glue (crawler e job)

Objetivo: ETL de dados do S3 para consulta analítica.

Como utilizar os arquivos: terraform\_glue\_hostal.tf, cloudformation\_glue\_hostal.yaml

Passo a passo:  
1) Crie crawler no AWS Glue. 2) Defina a origem (S3) e o destino (Athena ou Redshift). 3) Agende ou execute manualmente.

## 11. Amazon SageMaker (modelo)

Objetivo: Oferecer modelos preditivos para otimizar reservas.

Como utilizar os arquivos: hostal\_stack.py

Passo a passo:  
1) Criar modelo e endpoint no SageMaker. 2) Implantar via CDK. 3) Usar a partir da Step Function ou diretamente via app.py.

## 12. Amazon Athena (consultas)

Objetivo: Realizar consultas SQL em dados armazenados no S3.

Como utilizar os arquivos: Athena\_estrutura\_consulta.sql

Passo a passo:  
1) Acesse Athena. 2) Configure o bucket S3. 3) Execute as queries para ver histórico de reservas e serviços.

## 13. Amazon QuickSight (painel)

Objetivo: Visualizar os dados analíticos de uso e reservas.

Como utilizar os arquivos: quicksight\_analysis\_hostal.json

Passo a passo:  
1) Acesse QuickSight. 2) Configure origem Athena. 3) Importe ou recrie painel baseado no JSON fornecido.

## 14. CI/CD com pipeline\_generator.py

Objetivo: Criar infraestrutura sem usar Terraform ou CloudFormation.

Como utilizar os arquivos: pipeline\_generator.py

Passo a passo:  
1) Configure boto3 com suas credenciais AWS. 2) Execute `python pipeline\_generator.py`. 3) Confirme criação da Step Function.

## 15. Ambientes Staging e Produção

Objetivo: Separar os testes da operação real com infraestrutura duplicada.

Como utilizar os arquivos: terraform.yml, variáveis de ambiente

Passo a passo:  
1) Criar duas configurações de variáveis. 2) Provisione com ambientes distintos. 3) Use tags para diferenciar os recursos.

## 16. Frontend (Next.js ou integração)

Objetivo: Interface visual para hóspedes e administradores.

Como utilizar os arquivos: frontend/ (estrutura a ser definida)

Passo a passo:  
1) Criar frontend em Next.js. 2) Conectar às APIs REST. 3) Deploy via Vercel ou AWS Amplify.

## 17. Integração com Wix (interface do cliente)

Objetivo: Exibir serviços e permitir reservas no site Wix.

Como utilizar os arquivos: chatbot\_handler.js

Passo a passo:  
1) Ativar Velo no Wix. 2) Adicionar componente de input/chat. 3) Conectar ao backend via fetch().

## 18. Chatbot com ChatGPT via API

Objetivo: Atender dúvidas automaticamente e com empatia.

Como utilizar os arquivos: chatbot\_handler.js

Passo a passo:  
1) Obter chave OpenAI. 2) Editar o JS com a chave. 3) Conectar componente de input. 4) Enviar perguntas ao ChatGPT.

## 19. Integração do Chatbot com o Backend

Objetivo: Obter respostas com base em dados reais do sistema.

Como utilizar os arquivos: chatbot\_handler.js + app.py

Passo a passo:  
1) Verificar chamadas à API via fetch(). 2) Expor rotas no backend para /servicos, /reserva. 3) Validar com logs do Wix.

## 20. Resumo de Arquivos do Projeto

Objetivo: Consolidar os principais arquivos e sua finalidade.

Como utilizar os arquivos: Todos os arquivos do projeto.

Passo a passo:  
1) Ver README.txt (caso incluído). 2) Identificar por nome e estrutura o papel de cada arquivo para troubleshooting.

## 21. Instruções de Deploy e Execução

Objetivo: Guia final de como colocar tudo em produção.

Como utilizar os arquivos: Todos os arquivos

Passo a passo:  
1) Provisione infraestrutura. 2) Suba API e frontend. 3) Configure chatbot. 4) Verifique QuickSight. 5) Validar tudo no ambiente Prod.

## 22. Integração Convergente de Todas as Soluções

Objetivo: Unificar todos os módulos do projeto em uma operação funcional.

Como utilizar os arquivos: Todos os módulos mencionados nas seções anteriores.

Passo a passo:  
1) Provisione com Terraform/CDK. 2) Implante o backend (app.py). 3) Configure chatbot no Wix. 4) Verifique integração com banco e SageMaker. 5) Execute queries com Athena. 6) Veja relatórios no QuickSight.