

# API Patronum

Feitiços para Iniciantes em Back-End



FERNANDO HENRIQUE

# OS FEITIÇOS DO BACK-END COM API

## O Idioma Secreto dos Sistemas Encantados

Seja bem-vindo(a) ao mundo mágico do desenvolvimento Back-End! Você está prestes a embarcar em uma jornada onde códigos se transformam em pontes, dados ganham vida e sistemas se comunicam como por telepatia. No "API Patronum: Feitiços para Iniciantes em Back-End", desvendaremos os segredos das APIs (Interface de Programação de Aplicações), as verdadeiras varinhas mágicas que impulsionam a maior parte da internet que conhecemos.



FERNANDO HENRIQUE

# 01

## API REST

# A PORTA MÁGICA PARA O MUNDO DO BACK-END



# AS LEIS MÁGICAS QUE REGEM O API REST

## ENDPOINTS: AS PASSAGENS SECRETAS DO CASTELO

APIs REST usam internet (HTTP) para permitir que sistemas troquem dados. São baseadas em endpoints como /usuarios, /produtos, /login.

Exemplo real – Criando um endpoint simples (Node.js + Express):

```
const express = require('express');
const app = express();

app.get('/status', (req, res) => {
  res.json({ mensagem: "API funcionando corretamente!" });
});

app.listen(3000, () => console.log("Servidor rodando..."));
```



# OP

## API DE BANCO DE DADOS O COFRE DE GRINGOTES DOS SEUS DADOS



# COMO OS FEITICEIROS GUARDAM E RECUPERAM INFORMAÇÕES

## Query: OS ENCANTAMENTOS DE BUSCA

Sempre que você salva ou busca algo, está usando uma API de banco. Ela conecta o back-end ao banco de dados e executa consultas (query). Exemplo real — Buscando produtos do banco (MySQL + Node.js)

```
const mysql = require('mysql2');

const db = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
  database: "loja"
});

db.query("SELECT nome, preco FROM produtos", (err, resultado) => {
  console.log(resultado);
});
```



# 03

## API DE AUTENTICAÇÃO O GUARDIÃO DA PORTA DE ENTRADA



# COMO GARANTIR QUE APENAS USUÁRIOS AUTORIZADOS ACESSSEM

## Tokens JWT e Autenticação Moderna

Ela cuida de login, logout e segurança. A mais usada atualmente é baseada em JWT que define uma maneira compacta e independente de transmitir informações de forma segura entre partes, geralmente usando um objeto JSON., é um token seguro enviado ao cliente.

```
const jwt = require("jsonwebtoken");

const usuario = { id: 1, nome: "Fernando" };
const token = jwt.sign(usuario, "chave_super_secreta", { expiresIn: "1h" });

console.log("Token:", token);
```



# O4+

## API DE PAGAMENTOS

### A MAGIA POR TRÁS DAS TRANSAÇÕES ONLINE



# COMO FEITICEIROS LIDAM COM PAGAMENTOS SEGUROS

## PROTEÇÕES CONTRA ARTES DAS TREVAS FINANCEIRAS

Back-ends que lidam com compras usam APIs empresas como Stripe, PayPal e Mercado Pago.

Essas APIs tratam cartões, validações e segurança.

Exemplo real — Criando pagamento com Stripe

```
const stripe = require("stripe")("CHAVE_SECRETA");

async function pagar() {
  const pagamento = await stripe.paymentIntents.create({
    amount: 2500, // R$25,00
    currency: "brl"
  });

  console.log("Pagamento iniciado:", pagamento.client_secret);
}

pagar();
```



# 05

## API DE ENVIO DE E-MAILS AS CORUJAS DIGITAIS DO SEU SISTEMA



# COMO ENVIAR MENSAGENS SEM PRECISAR DE UMA CORUJA DE VERDADE

## COMO AUTOMATIZAR CONFIRMAÇÕES E NOTIFICAÇÕES

As APIs de envio de e-mails são responsáveis por enviar mensagens automáticas para os usuários do seu sistema. Elas são usadas em situações muito comuns no back-end. Essas APIs facilitam o processo porque você não precisa configurar servidores de e-mail manualmente — tudo é feito por ferramentas especializadas que garantem entrega rápida e segura.

As mais usadas são:

Nodemailer → Ideal para projetos Node.js e muito simples para começar

SendGrid → Focado em alta entrega e monitoramento profissional

Mailgun → Usado em sistemas maiores, com ótima reputação de envio

Essas ferramentas funcionam como “corujas digitais”: você manda a mensagem, e elas entregam no destino certo.

```
const nodemailer = require("nodemailer");

const mail = nodemailer.createTransport({
  service: "gmail",
  auth: { user: "exemplo@gmail.com", pass: "senha123" },
});

mail.sendMail({
  from: "exemplo@gmail.com",
  to: "cliente@gmail.com",
  subject: "Bem-vindo!",
  text: "Seu cadastro foi concluído!",
});
```



# OS

## API GRAPHQL

### O MAPA DO MAROTO DA BUSCA DE DADOS



FERNANDO HENRIQUE

VEJA APENAS O QUE PRECISA:  
"EU JURO SOLENEMENTE QUE NÃO VOU  
FAZER NADA DE BOM"

## REST VS GRAPHQL: DUELO DE FEITIÇOS

Ao contrário do REST, onde você recebe tudo, o GraphQL entrega só o que você pedir.

Perfeito para apps grandes com telas variadas.

Exemplo real – Query GraphQL simples

```
query {
  usuario(id: 10) {
    nome
    email
  }
}
```



# OZ

## API WEBSOCKET

### O FEITICO DE COMUNICAÇÃO INSTANTÂNEA



FERNANDO HENRIQUE

# QUANDO INVOCAR COMUNICAÇÃO EM TEMPO REAL

CONEXÕES CONTÍNUAS: COMO SE FOSSE  
LEGILIMÊNCIA DIGITAL

Ideal para chats, notificações instantâneas, jogos e dashboards.

Diferente do HTTP, ela mantém conexão permanente com o cliente.

Exemplo real – Servidor WebSocket básico

```
const WebSocket = require("ws");
const server = new WebSocket.Server({ port: 8080 });

server.on("connection", socket => {
  socket.send("Conexão estabelecida!");

  socket.on("message", msg => {
    socket.send("Você enviou: " + msg);
  });
});
```



# AGRADECIMENTOS



# PARABÉNS JOVEM BRUXO!

Se você chegou até aqui, significa que percorreu cada capítulo, lançou cada feitiço de código e abriu cada porta mágica que as APIs podem oferecer. Sua jornada no mundo do desenvolvimento back-end está apenas começando, mas você já demonstrou algo que nem todos possuem: vontade de aprender e coragem para explorar o desconhecido.

Que esse pequeno ebook tenha iluminado um pouco seu caminho e mostrado que, assim como na magia, a tecnologia também exige prática, estudo e curiosidade.

Cada requisição enviada, cada resposta recebida, cada integração feita — tudo isso é parte do seu crescimento como desenvolvedor.

Desejo que você continue avançando, aprimorando suas habilidades e buscando novos conhecimentos pelo vasto universo das APIs. Que sua varinha (ou teclado) nunca pare de criar, testar e conjurar novas soluções.

E lembre-se sempre:

No mundo da programação, o conhecimento é a magia mais poderosa que existe.

Que seus próximos feitiços de código sejam ainda mais brilhantes.

Até a próxima aventura, jovem bruxo!





Esse Ebook foi gerado por IA, diagramado por humano.

O passo a passo se encontra no meu Github

Esse conteúdo foi gerado com fins didáticos de construção, não foi  
realizado uma validação cuidadosa humana no conteúdo e pode  
conter erros gerados por uma IA.



<https://github.com/ferfau>

