

Back-end (Node.js / Express)

```
const express = require('express');
const cors = require('cors');
const bodyParser = require('body-parser');
const fs = require('fs');
const app = express();
app.use(cors());
app.use(bodyParser.json());

const PORT = 3001;

// Banco de dados simples (arquivo JSON)
const USERS_FILE = './data/users.json';
const GAMES_FILE = './data/games.json';

// Admin login fixo
const ADMIN_USER = 'Gaspar';
const ADMIN_PASS = '829062';

// Função para ler JSON
function readJSON(file) {
  if (!fs.existsSync(file)) return [];
  const data = fs.readFileSync(file);
  return JSON.parse(data);
}

// Função para salvar JSON
function saveJSON(file, data) {
  fs.writeFileSync(file, JSON.stringify(data, null, 2));
}

// Cadastro usuário
app.post('/api/register', (req, res) => {
  const { phone } = req.body;
```

```

    if (!phone) return res.status(400).json({ error: 'Telefone obrigatório' });
    let users = readJSON(USERS_FILE);
    if (users.find(u => u.phone === phone)) {
        return res.status(400).json({ error: 'Telefone já cadastrado' });
    }
    users.push({ phone, balance: 100, history: [] }); // saldo inicial 100 para teste
    saveJSON(USERS_FILE, users);
    res.json({ success: true, balance: 100 });
});

// Login usuário
app.post('/api/login', (req, res) => {
    const { phone } = req.body;
    let users = readJSON(USERS_FILE);
    const user = users.find(u => u.phone === phone);
    if (!user) return res.status(400).json({ error: 'Usuário não encontrado' });
    res.json({ success: true, user });
});

// Admin login
app.post('/api/admin/login', (req, res) => {
    const { username, password } = req.body;
    if (username === ADMIN_USER && password === ADMIN_PASS) {
        res.json({ success: true });
    } else {
        res.status(401).json({ error: 'Credenciais inválidas' });
    }
});

// Jogo Tigre (exemplo simples)
app.post('/api/game/tigre/play', (req, res) => {
    const { phone, bet } = req.body;
    if (!phone || !bet) return res.status(400).json({ error: 'Dados incompletos' });
    let users = readJSON(USERS_FILE);
    let user = users.find(u => u.phone === phone);
    if (!user) return res.status(400).json({ error: 'Usuário não encontrado' });
    if (user.balance < bet) return res.status(400).json({ error: 'Saldo insuficiente' });

```

```

// Controle simples: 70% chance de perder, 30% chance de ganhar x2
const chance = Math.random();
let result = { win: false, amount: 0 };
if (chance < 0.3) {
  const winAmount = bet * 2;
  user.balance += winAmount;
  result = { win: true, amount: winAmount };
} else {
  user.balance -= bet;
  result = { win: false, amount: -bet };
}
// Histórico
user.history.push({ game: 'Tigre', bet, result: result.win ? 'win' : 'lose', amount:
result.amount, date: new Date() });
saveJSON(USERS_FILE, users);

res.json({ success: true, balance: user.balance, result });
});

// Pix info
app.get('/api/pix', (req, res) => {
  res.json({ pixKey: '91985828322' });
});

app.listen(PORT, () => {
  console.log(`Server running on http://localhost:${PORT}`);
});

```

Front-end (React)

```

import React, { useState } from 'react';

const API_URL = 'http://localhost:3001/api';

function App() {
  const [phone, setPhone] = useState('');
  const [balance, setBalance] = useState(null);

```

```
const [loggedIn, setLoggedIn] = useState(false);
const [bet, setBet] = useState(0);
const [message, setMessage] = useState('');

const register = async () => {
  if (!phone) {
    setMessage('Informe o número de telefone');
    return;
  }
  const res = await fetch(`${API_URL}/register`, {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ phone }),
  });
  const data = await res.json();
  if (data.success) {
    setBalance(data.balance);
    setLoggedIn(true);
    setMessage('Cadastro realizado com sucesso!');
  } else {
    setMessage(data.error || 'Erro no cadastro');
  }
};

const login = async () => {
  if (!phone) {
    setMessage('Informe o número de telefone');
    return;
  }
  const res = await fetch(`${API_URL}/login`, {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ phone }),
  });
  const data = await res.json();
  if (data.success) {
    setBalance(data.user.balance);
    setLoggedIn(true);
  }
}
```

```

        setMessage('Login realizado!');
    } else {
        setMessage(data.error || 'Erro no login');
    }
};

const playTigre = async () => {
    if (bet <= 0) {
        setMessage('Informe um valor de aposta válido');
        return;
    }
    const res = await fetch(`${API_URL}/game/tigre/play`, {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({ phone, bet: Number(bet) }),
    });
    const data = await res.json();
    if (data.success) {
        setBalance(data.balance);
        if (data.result.win) {
            setMessage(`Você ganhou! Valor ganho: R${data.result.amount}`);
        } else {
            setMessage(`Você perdeu! Valor perdido: R${-data.result.amount}`);
        }
    } else {
        setMessage(data.error || 'Erro ao jogar');
    }
};

if (!loggedIn) {
    return (
        <div style={{ padding: 20 }}>
            <h1>Gasparbet - Cadastro/Login</h1>
            <input
                placeholder="Número de telefone"
                value={phone}
                onChange={e => setPhone(e.target.value)}
            />

```

```

        <br />
        <button onClick={register}>Cadastrar</button>
        <button onClick={login}>Entrar</button>
        <p>{message}</p>
    </div>
  );
}

return (
  <div style={{ padding: 20 }}>
    <h1>Gasparbet - Jogo Tigre</h1>
    <p>Saldo: R$ {balance.toFixed(2)}</p>
    <input
      type="number"
      placeholder="Valor da aposta"
      value={bet}
      onChange={e => setBet(e.target.value)}
    />
    <button onClick={playTigre}>Jogar</button>
    <p>{message}</p>
    <hr />
    <p>Pix para depósitos: <b>91985828322</b></p>
    <p>Maior de 18 anos, jogo responsável.</p>
  </div>
);
}

export default App;

```