# 🏰 Documento de Construção do Jogo: A Jornada Matemática

## 🎮 Descrição Geral

"A Jornada Matemática" é um jogo educacional em HTML, CSS e JavaScript que leva os(as) jogadores(as) por uma aventura lúdica onde devem resolver desafios matemáticos baseados em progressões aritméticas (PA) e geométricas (PG). A estética medieval e os personagens interativos reforçam o aprendizado de forma envolvente.

## 🧱 Estrutura do Projeto

/jornada-matematica  
├── index.html # Página principal do jogo  
├── MedievalSharp-Regular.ttf # Fonte personalizada  
├── floresta-background.png # Plano de fundo dos níveis  
├── cidade-background.png  
├── guardiao-background.jpeg  
├── bobo-background.jpeg  
├── medieval-background.png  
├── reino-background.jpeg  
├── server.js # Servidor Node.js com Express  
└── package.json # Dependências do Node.js

## 🌐 Servidor com Node.js e Express

Arquivo: server.js

const express = require('express');  
const path = require('path');  
const app = express();  
  
app.use(express.static(\_\_dirname));  
  
app.get('/', (req, res) => {  
 res.sendFile(path.join(\_\_dirname, 'index.html'));  
});  
  
const PORT = process.env.PORT || 3000;  
app.listen(PORT, () => {  
 console.log(`Servidor rodando em http://localhost:${PORT}`);  
});

Como rodar localmente:

1. Instale o Node.js (https://nodejs.org).  
2. No terminal:  
 npm init -y  
 npm install express  
 node server.js  
3. Acesse http://localhost:3000 no navegador.

## 🚀 Deploy na AWS via Git

Pré-requisitos:

- Uma instância EC2 rodando Amazon Linux ou Ubuntu.  
- Git e Node.js instalados na máquina.  
- Porta 3000 liberada no grupo de segurança da instância.

Passo a passo do deploy:

1. Acesse sua instância EC2 via SSH:  
 ssh -i "sua-chave.pem" ec2-user@seu-endereco-publico

2. Clone o repositório do projeto:  
 git clone https://github.com/seu-usuario/jornada-matematica.git  
 cd jornada-matematica

3. Instale o Node.js e as dependências:  
 sudo yum update -y  
 sudo yum install nodejs git -y # ou `sudo apt install` no Ubuntu  
 npm install

4. Inicie o servidor:  
 node server.js

5. (Opcional) Execute com PM2 para manter rodando em segundo plano:  
 npm install -g pm2  
 pm2 start server.js  
 pm2 save  
 pm2 startup

6. Acesse pelo navegador usando o IP público da instância:  
 http://<SEU\_IP\_PUBLICO>:3000

## 🧩 Recursos Utilizados

- HTML5 + CSS3: estrutura da interface e ambientação visual.  
- JavaScript puro: lógica de jogo, interações e navegação por níveis.  
- Express (Node.js): servidor leve para servir os arquivos do jogo.  
- Fonte Medieval: MedievalSharp (TrueType Font).  
- Imagens e cenários temáticos: arquivos .png e .jpeg ambientados.

## 📌 Observações Finais

- Todos os assets devem estar no mesmo diretório do server.js para garantir que o Express sirva corretamente os arquivos.  
- O jogo pode ser facilmente convertido em um Progressive Web App (PWA) ou adaptado para dispositivos móveis no futuro.  
- Para ambientes de produção, considere usar um proxy reverso como Nginx e habilitar HTTPS.