

# Relatório de Projeto: Desenvolvimento de Bot para Automação de Mineração

**Autor: Felipe Coutinho**

Período de Desenvolvimento: Desde 01/04/2025 (em andamento)

Projeto: Bot de Mineração Automatizada

Status: Versão 2.0 - Em desenvolvimento

## Resumo Executivo

Este relatório apresenta o desenvolvimento de um software para automação da coleta de recursos em um ambiente de jogo, utilizando Python e as bibliotecas pyautogui para simulação de ações do usuário e tqdm para feedback visual de progresso. O bot executa um ciclo completo de mineração composto por 6 fases, incluindo troca de ferramentas e geração opcional de relatório final. O projeto inclui documentação técnica detalhada e planeja sua evolução futura para uma solução baseada em visão computacional, aumentando robustez e adaptabilidade.

## 1. Introdução 1.1 Contexto

A coleta manual de recursos em jogos digitais é uma tarefa repetitiva que consome tempo valioso dos usuários, especialmente daqueles com disponibilidade limitada. Automatizar esse processo pode melhorar a eficiência e liberar o usuário para atividades estratégicas, mantendo o progresso do jogo.

## 1.2 Objetivos

**Primário:** Desenvolver um script funcional para automatizar um ciclo de mineração em 6 fases, com navegação entre pontos de coleta, realizando 9 cliques por ciclo para extrair os recursos disponíveis.

**Secundários:**

- Implementar troca automatizada de ferramentas e utilização de reparadores para maximizar a durabilidade dos equipamentos.
- Fornecer feedback visual em tempo real sobre o andamento da automação.
- Garantir configurabilidade para diferentes cenários, mantendo possível intervenção manual.
- Criar documentação técnica clara e detalhada para facilitar uso e manutenção.

# Relatório de Projeto: Desenvolvimento de Bot para Automação de Mineração

## 2. Metodologia e Ferramentas

Optou-se pela automação via interface gráfica (GUI Automation), simulando interações humanas para garantir compatibilidade com o ambiente de jogo.

Linguagem: Python 3, pela sua simplicidade e ecossistema rico para prototipagem.

Automação GUI: pyautogui, para controle programático do mouse e teclado, incluindo detecção de imagens na tela.

Feedback Visual: tqdm, para exibição de barras de progresso e status no terminal.

## 3. Desenvolvimento e Implementação 3.1 Estrutura do Código

O projeto está modularizado em funções principais:

- mouse\_position(): Navegação entre pontos de mineração.
- item(): Loop principal de mineração e uso de reparadores.
- troca\_ferramenta(): Troca automática de ferramentas no inventário para gerenciar desgaste.
- tempo(): Cálculo e exibição de relatório final com métricas de desempenho, incluindo opção para salvar em planilha.

## 3.2 Desafios e Soluções

Dependência de Coordenadas Fixas:

Desafio: Sensibilidade a variações de resolução e posicionamento da janela do jogo.

Solução: Padronização da resolução em 1366x768 e configuração rigorosa da câmera, com script auxiliar para calibrar coordenadas localmente.

Sincronização com o Jogo:

Desafio: Garantir que o bot respeite tempos de resposta e animações.

Solução: Inserção de delays estratégicos (time.sleep()), ajustados empiricamente para evitar erros por ações prematuras.

## 4. Resultados e Entregáveis

## Relatório de Projeto: Desenvolvimento de Bot para Automação de Mineração

- Software Funcional: Script Python capaz de executar o ciclo completo de mineração autonomamente.
- Documentação Técnica: Guia detalhado publicado no GitHub, explicando instalação, uso, configuração e roadmap futuro.
- Declaração Ética: Aviso legal reforçando o uso educacional e desencorajando uso indevido para vantagens desleais.
- Relatório de Uso de IA: Transparência no uso de ferramentas de IA como suporte ao desenvolvimento.

### 5. Conclusão e Próximos Passos

O bot apresenta uma prova de conceito funcional para automação via GUI, atingindo seus objetivos iniciais. Contudo, a abordagem atual, baseada em coordenadas fixas, é limitada e frágil em ambientes variáveis.

Para ampliar robustez e adaptabilidade, planeja-se evoluir para uma solução baseada em visão computacional, utilizando bibliotecas como OpenCV e modelos de detecção como YOLO para:

- Detectar dinamicamente nós de minério independentemente da posição na tela.
- Adaptar-se a diferentes resoluções e layouts.
- Tomar decisões mais complexas com base na análise visual do ambiente.

Essa evolução permitirá transformar o bot em um agente inteligente, capaz de operar com maior autonomia e confiabilidade.