**Sistema de Inventário de Máquinas na Rede**

Sistema completo para coleta automática de informações de hardware, software e configurações de computadores Windows em rede, com armazenamento centralizado em MySQL.

**📋 Funcionalidades**

**Dados Coletados**

* **Identificação**: Nome do computador, domínio, usuário logado
* **Software**: Lista completa de programas instalados com versões
* **Sistema Operacional**: Nome, versão, Service Pack, serial
* **Processador**: Modelo, velocidade, quantidade de cores
* **Memória RAM**: Capacidade total, slots utilizados, velocidade
* **Rede**: Placas de rede, MAC Address, IP, máscara, gateway
* **Discos**: Unidades, tipo, tamanho, espaço livre, sistema de arquivos
* **BIOS**: Versão, fabricante, data de release
* **Controladores**: IDE, USB, Floppy
* **Periféricos**: Mouse, teclado, monitores, impressoras
* **Último Logon**: Usuário e data/hora

**🏗️ Arquitetura**

inventario-rede/

│

├── backend/ # API Flask centralizadora

│ ├── app.py # Servidor principal

│ ├── database.py # Conexão MySQL

│ └── requirements.txt # Dependências Python

│

├── agent/ # Agente que roda em cada máquina

│ ├── agent.py # Script principal

│ ├── collector.py # Coleta de dados WMI

│ └── utils.py # Funções auxiliares

│

├── sql/

│ └── schema.sql # Criação do banco e tabelas

│

└── docs/

└── README.md # Esta documentação

**⚙️ Instalação e Configuração**

**1. Pré-requisitos**

* **Python 3.8+** ([Download](https://www.python.org/downloads/))
* **MySQL Community Server** ([Download](https://dev.mysql.com/downloads/))
* **VS Code** (opcional) ([Download](https://code.visualstudio.com/))
* **Windows** (testado no Windows 10/11)

**2. Instalação das Dependências**

# Instalar bibliotecas Python necessárias

pip install flask flask-cors mysql-connector-python pywin32 wmi psutil requests

Ou usar o arquivo requirements.txt:

pip install -r backend/requirements.txt

**3. Configuração do Banco MySQL**

1. **Criar o banco de dados:**

-- Executar no MySQL Workbench ou linha de comando

CREATE DATABASE inventario;

1. **Executar o script de criação das tabelas:**

mysql -u root -p inventario < sql/schema.sql

1. **Configurar credenciais no código:**
   * Editar backend/database.py e ajustar as configurações:

DB\_CONFIG = {

'host': 'localhost',

'database': 'inventario',

'user': 'root',

'password': 'SUA\_SENHA\_MYSQL' # Inserir sua senha

}

**4. Configuração do Agente**

Criar arquivo agent/config.json:

{

"server\_url": "http://SEU\_SERVIDOR:5000",

"api\_endpoint": "/api/inventory",

"timeout": 30,

"retry\_attempts": 3,

"backup\_enabled": true,

"log\_level": "INFO"

}

**🚀 Execução**

**1. Iniciar o Servidor (Backend)**

cd backend

python app.py

O servidor estará disponível em: http://localhost:5000

**Endpoints disponíveis:**

* GET /health - Status da API
* GET /api/test-connection - Teste de conexão com MySQL
* GET /api/machines - Listar todas as máquinas
* GET /api/machines/<nome> - Detalhes de uma máquina
* POST /api/inventory - Receber dados de inventário (usado pelo agente)

**2. Executar o Agente nas Máquinas Cliente**

cd agent

python agent.py

**Opções disponíveis:**

python agent.py # Execução normal (coleta e envia dados)

python agent.py --test # Modo teste (só coleta, não envia)

python agent.py --help # Ajuda

python agent.py --config # Mostrar configurações

**📊 Fluxo de Funcionamento**

1. **Agente** roda em cada máquina Windows
2. **Coleta** dados usando WMI e APIs do Windows
3. **Valida** e formata dados em JSON
4. **Envia** via HTTP POST para API centralizadora
5. **API** processa e armazena no MySQL
6. **Backup local** é criado em caso de falha

**🔧 Configurações Avançadas**

**Configuração de Rede**

Para usar em rede corporativa, ajustar:

1. **No servidor:**
   * Alterar app.py: app.run(host='0.0.0.0', port=5000)
   * Configurar firewall para porta 5000
2. **Nos clientes:**
   * Editar config.json com IP do servidor
   * Exemplo: "server\_url": "http://192.168.1.100:5000"

**Agendamento Automático**

Para executar periodicamente, usar:

**Windows Task Scheduler:**

1. Abrir "Agendador de Tarefas"
2. Criar nova tarefa
3. Configurar para executar python agent.py
4. Definir periodicidade (diária, semanal, etc.)

**Script Batch (opcional):**

@echo off

cd C:\caminho\para\agent

python agent.py

pause

**Logs e Monitoramento**

* **Logs do servidor**: backend/inventory\_api.log
* **Logs do agente**: agent/inventory\_agent.log
* **Backups**: agent/backups/
* **Relatórios**: agent/reports/ (se habilitado)

**🛠️ Solução de Problemas**

**Problemas Comuns**

**1. Erro de conexão WMI**

Solução: Executar como Administrador

**2. Erro MySQL "Access denied"**

Solução: Verificar usuário/senha em database.py

**3. Timeout na conexão**

Solução: Verificar firewall e conectividade de rede

**4. "Module not found"**

Solução: pip install -r requirements.txt

**Debug e Teste**

# Testar conexão com servidor

curl http://localhost:5000/health

# Testar conexão MySQL

curl http://localhost:5000/api/test-connection

# Executar agente em modo teste

python agent.py --test

**📈 Consultas SQL Úteis**

-- Listar todas as máquinas

SELECT nome\_computador, dominio, usuario\_logado, data\_coleta

FROM maquinas

ORDER BY data\_coleta DESC;

-- Softwares mais comuns

SELECT nome, COUNT(\*) as total\_maquinas

FROM softwares

GROUP BY nome

ORDER BY total\_maquinas DESC

LIMIT 10;

-- Máquinas por sistema operacional

SELECT so.nome, COUNT(\*) as total

FROM sistema\_operacional so

JOIN maquinas m ON so.maquina\_id = m.id

GROUP BY so.nome;

-- Espaço em disco por máquina

SELECT m.nome\_computador, SUM(d.tamanho\_gb) as total\_gb

FROM maquinas m

JOIN discos d ON m.id = d.maquina\_id

GROUP BY m.nome\_computador

ORDER BY total\_gb DESC;

**🔒 Segurança**

* Execute o agente com privilégios de administrador
* Configure firewall adequadamente
* Use HTTPS em ambiente de produção
* Implemente autenticação na API se necessário
* Mantenha backups regulares do banco MySQL

**📞 Suporte**

Para problemas ou dúvidas:

1. Verifique os logs de erro
2. Execute em modo teste primeiro
3. Confirme conectividade de rede
4. Valide configurações do MySQL

**📝 Licença**

Sistema desenvolvido para uso interno/educacional. Adapte conforme necessário para seu ambiente.