

INSTRUÇÕES COMPLETAS DE INSTALAÇÃO E EXECUÇÃO

Sistema de Gerenciamento de Matrículas Escolares

SUMÁRIO

- [1. Requisitos do Sistema](#)
 - [2. Instalação do MySQL](#)
 - [3. Instalação do Redis](#)
 - [4. Instalação do Backend](#)
 - [5. Instalação do Frontend](#)
 - [6. Execução do Sistema](#)
 - [7. Testes](#)
 - [8. Troubleshooting](#)
-

REQUISITOS DO SISTEMA

Hardware Mínimo

- CPU:** Dual-core 2.0 GHz
- RAM:** 4 GB
- Disco:** 2 GB livres

Software Obrigatório

- Java:** 17 ou superior
- Maven:** 3.8 ou superior
- MySQL:** 8.0 ou superior
- Redis:** 6.0 ou superior
- Git:** Para clonar o repositório
- Navegador:** Chrome, Firefox, Safari ou Edge (versão recente)

Verificar Instalações

Bash

```
# Verificar Java
java -version

# Verificar Maven
mvn -version

# Verificar Git
git --version
```

INSTALAÇÃO DO MYSQL

Windows

1. Baixar MySQL Community Server

- Acesse: <https://dev.mysql.com/downloads/mysql/>
- Selecione versão 8.0 ou superior
- Clique em "Download"

2. Executar Instalador

- Execute o arquivo `.msi` baixado
- Clique em "Next" até "MySQL Server Configuration"
- Selecione "Development Machine"
- Clique em "Next" e "Finish"

3. Configurar MySQL

- Na tela "MySQL Server Configuration", deixe as opções padrão
- Clique em "Next"
- Na tela de autenticação, defina senha para root
- Clique em "Next" e "Finish"

4. Verificar Instalação

macOS (com Homebrew)

Bash

```
# Instalar Homebrew (se não tiver)
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh )"

# Instalar MySQL
brew install mysql

# Iniciar MySQL
brew services start mysql

# Verificar instalação
mysql -u root
```

Linux (Ubuntu/Debian)

Bash

```
# Atualizar pacotes
sudo apt-get update

# Instalar MySQL
sudo apt-get install mysql-server

# Iniciar MySQL
sudo systemctl start mysql

# Verificar instalação
sudo mysql -u root
```

INSTALAÇÃO DO REDIS

Windows

1. Usar WSL (Windows Subsystem for Linux)
2. Instalar Redis no WSL
3. Iniciar Redis

macOS (com Homebrew)

Bash

```
# Instalar Redis
brew install redis

# Iniciar Redis
brew services start redis

# Verificar instalação
redis-cli ping

# Deve retornar: PONG
```

Linux (Ubuntu/Debian)

Bash

```
# Instalar Redis
sudo apt-get install redis-server

# Iniciar Redis
sudo systemctl start redis-server

# Verificar instalação
redis-cli ping

# Deve retornar: PONG
```

INSTALAÇÃO DO BACKEND

1. Clonar o Repositório

Bash

```
# Navegar para pasta de projetos
cd ~/projetos

# Clonar repositório
git clone <url-do-repositorio>
cd matricula-escolar-backend
```

2. Criar Banco de Dados

Bash

```
# Conectar ao MySQL como root
mysql -u root -p

# Executar scripts SQL
source scripts/01-schema.sql;
source scripts/02-create-user.sql;

# Sair do MySQL
exit
```

3. Configurar Aplicação

Editar `src/main/resources/application.properties` :

Plain Text

```
# MySQL
spring.datasource.url=jdbc:mysql://localhost:3306/matricula_escolar?
useSSL=false&serverTimezone=UTC
spring.datasource.username=matricula_user
spring.datasource.password=matricula_password

# Redis
spring.redis.host=localhost
spring.redis.port=6379

# JWT (mude para uma chave segura em produção)
jwt.secret=sua_chave_secreta_super_segura_aqui_com_mais_de_256_bits_para_hs256
jwt.expiration=86400000
```

4. Compilar Projeto

Bash

```
# Compilar
mvn clean compile

# Empacotar
mvn package
```

5. Executar Backend

Bash

```
# Opção 1: Via Maven
mvn spring-boot:run

# Opção 2: Via JAR
java -jar target/matricula-escolar-1.0.0.jar
```

Saída esperada:

Plain Text

```
[INFO]: # "Application 'matricula-escolar' started successfully"
[INFO]: http://localhost:8080/api "Server running on"
```

INSTALAÇÃO DO FRONTEND

1. Navegar para Pasta do Frontend

Bash

```
cd ../matricula-escolar-frontend
```

2. Configurar URL da API

Editar js/api.js :

JavaScript

```
const API_BASE_URL = 'http://localhost:8080/api';
```

3. Servir Frontend

Opção 1: Python (Recomendado)

Bash

```
# Python 3
python -m http.server 3000

# Python 2
python -m SimpleHTTPServer 3000
```

Opção 2: Node.js

Bash

```
# Instalar http-server (primeira vez )  
npm install -g http-server  
  
# Executar  
http-server -p 3000
```

Opção 3: VS Code (Live Server)

1. Instale extensão "Live Server" no VS Code
2. Clique com botão direito em `index.html`
3. Selecione "Open with Live Server"

Acesso: <http://localhost:3000>

EXECUÇÃO DO SISTEMA

Verificação Pré-Execução

Bash

```
# Verificar MySQL  
mysql -u matricula_user -p matricula_escolar -e "SELECT COUNT(*) FROM usuarios;"  
  
# Verificar Redis  
redis-cli ping  
# Deve retornar: PONG  
  
# Verificar Backend  
curl http://localhost:8080/api/auth/login  
# Deve retornar erro 405 (método não permitido )  
  
# Verificar Frontend  
curl http://localhost:3000  
# Deve retornar HTML da página
```

Iniciar Todos os Serviços

Terminal 1 - MySQL:

Bash

```
# Já deve estar rodando como serviço  
# Se não, execute:  
mysql -u root -p
```

Terminal 2 - Redis:

Bash

```
redis-server
```

Terminal 3 - Backend:

Bash

```
cd matricula-escolar-backend  
mvn spring-boot:run
```

Terminal 4 - Frontend:

Bash

```
cd matricula-escolar-frontend  
python -m http.server 3000
```

Acessar Sistema

1. Abra navegador
2. Acesse: <http://localhost:3000>
3. Faça login com credenciais de teste:
 - Email: `admin@matricula.com`
 - Senha: `admin123456`

TESTES

Teste de Login

1. Na página de login, insira:
 - Email: `admin@matricula.com`

- Senha: admin123456
2. Clique em "Entrar"
 3. Deve ser redirecionado para o dashboard

Teste de Criação de Aluno

1. Clique em "Alunos" na navbar
2. Clique em "Novo Aluno"
3. Preencha o formulário:
 - Nome: João Silva
 - CPF: 123.456.789-00
 - Email: joao@teste.com
 - Telefone: (11) 99999-9999
 - Endereço: Rua Teste, 100
 - Idade: 12
 - Data de Nascimento: 2011-05-15
 - Turma: 6º A
 - Turno: MATUTINO
4. Clique em "Salvar"
5. Deve aparecer mensagem de sucesso

Teste de Criação de Matrícula

1. Clique em "Matrículas" na navbar
2. Clique em "Nova Matrícula"
3. Selecione:
 - Aluno: João Silva
 - Turma: 6º A
 - Turno: MATUTINO
4. Clique em "Registrar Matrícula"
5. Deve aparecer mensagem de sucesso

Teste de Fila Redis

1. No dashboard, verifique "Fila de Requisições"

2. Registre uma matrícula
 3. A fila deve aumentar em 1
-

TROUBLESHOOTING

Erro: "Connection refused" ao MySQL

Solução:

Bash

```
# Verificar se MySQL está rodando
sudo systemctl status mysql

# Se não estiver, iniciar
sudo systemctl start mysql

# Windows: Verificar Serviços
# Services → MySQL80 → Iniciar
```

Erro: "Could not connect to Redis"

Solução:

Bash

```
# Verificar se Redis está rodando
redis-cli ping

# Se não retornar PONG, iniciar Redis
redis-server

# Windows (WSL):
wsl redis-server
```

Erro: "Port 8080 already in use"

Solução:

Bash

```
# Linux/Mac: Encontrar processo usando porta 8080
lsof -i :8080
```

```
# Matar processo
kill -9 <PID>

# Windows:
netstat -ano | findstr :8080
taskkill /PID <PID> /F
```

Erro: "Port 3000 already in use"

Solução:

Bash

```
# Usar porta diferente
python -m http.server 3001

# Atualizar URL no frontend
# Editar js/api.js e mudar para http://localhost:3001
```

Erro: "CORS error" no Frontend

Solução:

1. Verificar se backend está rodando em `http://localhost:8080`
2. Verificar se `application.properties` tem CORS habilitado
3. Limpar cache do navegador (Ctrl+Shift+Delete)

Erro: "Unauthorized" após login

Solução:

1. Verificar se token JWT está sendo armazenado em localStorage
2. Abrir DevTools (F12) → Application → LocalStorage
3. Verificar se `authToken` está presente
4. Se não, fazer login novamente

Erro: "Table 'matricula_escolar.usuarios' doesn't exist"

Solução:

Bash

```
# Verificar se scripts SQL foram executados
mysql -u matricula_user -p matricula_escolar
```

```
# Se tabelas não existem, executar scripts
source scripts/01-schema.sql;
source scripts/02-create-user.sql;
```

Erro: "Access denied for user 'matricula_user'"

Solução:

Bash

```
# Verificar se usuário foi criado
mysql -u root -p

# Executar script de criação de usuário
source scripts/02-create-user.sql;

# Verificar permissões
SHOW GRANTS FOR 'matricula_user'@'localhost';
```

PRÓXIMOS PASSOS

Desenvolvimento

1. Adicionar mais validações
2. Implementar paginação
3. Adicionar filtros avançados
4. Melhorar performance com caching

Produção

1. Usar HTTPS em vez de HTTP
2. Configurar firewall
3. Fazer backup regular do banco de dados
4. Monitorar performance
5. Implementar logging centralizado

Segurança

1. Alterar chave JWT para valor seguro

2. Configurar rate limiting
 3. Implementar 2FA
 4. Usar variáveis de ambiente para credenciais
 5. Fazer auditoria de segurança
-

SUORTE

Para dúvidas ou problemas:

1. Consulte o `DOCUMENTO_TECNICO.md` para detalhes arquiteturais
 2. Consulte o `README.md` do backend ou frontend
 3. Verifique os logs da aplicação
 4. Abra uma issue no repositório
-

Versão: 1.0.0

Última Atualização: Novembro de 2025

Status: Pronto para Produção