

## Relatório de Projeto 01 - Sistema de Biblioteca

**Disciplina:** Desenvolvimento de Software para Persistência

**Autores:** George Felipe Placido Nogueira e Antonio Avelino da Silva

**Matrículas:** 548348 (George), 552416 (Avelino)

**Professor:** Francisco Victor da Silva Pinheiro

---

### Descrição do Projeto

O projeto consistiu no desenvolvimento de uma **aplicação web com FastAPI**, com foco na construção de uma API REST para o gerenciamento de uma **biblioteca**. O armazenamento dos dados foi realizado em **arquivos CSV**, de modo a simular persistência em arquivos planos. Além das operações CRUD, funcionalidades adicionais como **exportação ZIP**, **cálculo de hash SHA256**, **conversão CSV para XML** e **logging** foram implementadas para enriquecer a solução com elementos de um ambiente real de produção.

---

### Entidades e Funcionalidades Implementadas

#### 1. Livro

- **Atributos:** id, titulo, autor, genero, ano\_publicacao, disponivel
- **Endpoints:**
  - GET /livros/ → Listar todos os livros, com filtros opcionais por autor, gênero e ano de publicação.
  - GET /livros/{id} → Buscar livro por ID.
  - POST /livros/ → Cadastrar novo livro.
  - PUT /livros/{id} → Atualizar um livro.
  - DELETE /livros/{id} → Remover um livro.
  - GET /livros/quantidade → Retorna a quantidade de livros cadastrados.
  - GET /livros/exportar → Exporta o arquivo livros.csv em formato .zip.
  - GET /livros/hash → Gera o hash SHA256 do arquivo ZIP.
  - GET /livros/csv-para-xml/{nome\_csv} → Converte o arquivo CSV em XML.

#### 2. Usuário

- **Atributos:** id, nome, email, telefone, data\_cadastro
- **Endpoints:**
  - GET /usuarios/ → Listar todos os usuários, com filtros opcionais.
  - GET /usuarios/{id} → Buscar usuário por ID.
  - POST /usuarios/ → Cadastrar novo usuário.
  - PUT /usuarios/{id} → Atualizar um usuário.
  - DELETE /usuarios/{id} → Remover um usuário.
  - GET /usuarios/quantidade → Retorna a quantidade de usuários cadastrados.
  - GET /usuarios/exportar → Exporta o arquivo usuarios.csv em formato .zip.
  - GET /usuarios/hash → Gera o hash SHA256 do arquivo ZIP.
  - GET /usuarios/csv-para-xml/{nome\_csv} → Converte o arquivo CSV em XML.

### 3. Empréstimo

- **Atributos:** id, usuario\_id, livro\_id, data\_emprestimo, data\_devolucao
  - **Endpoints:**
    - GET /emprestimos/ → Listar todos os empréstimos.
    - GET /emprestimos/{id} → Buscar empréstimo por ID.
    - POST /emprestimos/ → Registrar novo empréstimo.
    - PUT /emprestimos/{id} → Atualizar dados do empréstimo.
    - DELETE /emprestimos/{id} → Excluir empréstimo.
    - GET /emprestimos/quantidade → Retorna a quantidade de empréstimos cadastrados.
    - GET /emprestimos/exportar → Exporta o arquivo emprestimos.csv em formato .zip.
    - GET /emprestimos/hash → Gera o hash SHA256 do arquivo ZIP.
    - GET /emprestimos/csv-para-xml/{nome\_csv} → Converte o arquivo CSV em XML.
- 

### Funcionalidades Adicionais

- **Exportação de Dados:** Todas as entidades possuem funcionalidade de exportação de seus respectivos arquivos CSV em formato ZIP.
- **Hash de Arquivo:** Implementado hash SHA256 para garantir integridade dos arquivos exportados.

- **Conversão de Arquivos:** Os arquivos CSV podem ser convertidos para XML por meio de um endpoint específico.
  - **Logging:** Todas as operações de criação, atualização e exclusão são registradas em log para auditoria.
  - **Filtros de Busca:** Endpoints de listagem permitem busca refinada por campos específicos.
- 

## Divisão de Tarefas

- **Antonio Avelino da Silva:** Responsável pela implementação completa do CRUD das três entidades: Livro, Usuário e Empréstimo.
- **George Felipe Placido Nogueira:** Desenvolveu os endpoints adicionais, incluindo:
  - Geração de hash SHA256
  - Registro de logs
  - Conversão CSV → XML
  - Exportação ZIP
  - E também foi o responsável pela redação deste relatório técnico.

## Conclusão

O projeto proporcionou aos autores a oportunidade de aplicar na prática conceitos fundamentais do desenvolvimento backend com Python, como criação de APIs RESTful, tratamento de arquivos, logging e segurança de dados. A implementação completa das operações CRUD, juntamente com funcionalidades adicionais de exportação, hash e transformação de dados, simulou um sistema real com boas práticas de engenharia de software.