

# SECRETARIA +

SUA SECRETARIA ONLINE

---



KELLY PITELLA

# Sistema de Gestão de Secretaria Escolar

**Data:** 22/01/2025

**Autores:** Kelly Pitella

## Sumário

1. Introdução
2. Descrição do Projeto
3. Levantamento de Requisitos
4. Análise de Sistemas
5. Design do Sistema
6. Implementação
7. Testes
8. Conclusão
9. Anexos



# 1.Introdução

## 1.1Contextualização

Este relatório foi elaborado com o propósito de documentar todas as etapas do desenvolvimento do sistema Gestão de Secretaria Escolar, garantindo transparência no processo e servindo como um registro detalhado das decisões tomadas. Ele visa, também, fornecer uma visão clara do progresso alcançado, desde a concepção inicial até a entrega do sistema, destacando sua importância para a modernização dos processos administrativos escolares. Além disso, o documento facilita futuras atualizações e melhorias, assegurando que o sistema continue a atender às demandas das instituições de ensino de maneira eficiente e segura.

## 1.2 Objetivo do Projeto

O projeto Gestão de Secretaria Escolar foi concebido com o propósito de otimizar a administração escolar, integrando funcionalidades que abrangem desde o cadastro de alunos e professores até a organização de turmas e gerenciamento de notas. Este sistema visa, ainda, facilitar a geração de relatórios em formatos PDF e Excel, tornando a comunicação e a análise de dados mais eficientes.



## 2.Descrição do Projeto

### 2.1 Visão Geral

O Gestão de Secretaria Escolar é um sistema desenvolvido para modernizar e automatizar os processos administrativos de instituições de ensino. Seu principal objetivo é otimizar atividades como o cadastro de alunos e professores, organização de turmas, gerenciamento de notas e geração de relatórios. Ele busca atender à necessidade de maior eficiência, segurança e praticidade na administração escolar, proporcionando ferramentas que simplificam o trabalho diário e aprimoram a análise de dados educacionais.

### 2.2 Escopo

O projeto Gestão de Secretaria Escolar foi concebido com o propósito de otimizar a administração escolar, integrando funcionalidades que abrangem desde o cadastro de alunos e professores até a organização de turmas e gerenciamento de notas. Este sistema visa, ainda, facilitar a geração de relatórios em formatos PDF e Excel, tornando a comunicação e a análise de dados mais eficientes.



## **Incluído no projeto:**

- Cadastro e gerenciamento de informações de alunos, professores e administradores.
- Organização e controle de turmas e disciplinas.
- Registro e acompanhamento de notas e frequências.
- Geração de relatórios detalhados em formatos PDF e Excel.
- Interface amigável e acessível para usuários com diferentes níveis de habilidade tecnológica.
- Implementação de níveis de acesso para garantir a segurança e privacidade dos dados.

## **Não incluído no projeto:**

- Funcionalidades relacionadas a gestão financeira ou administrativa que não envolvam diretamente os dados escolares.
- Integração com sistemas de ensino a distância ou plataformas externas de aprendizado.
- Suporte a dispositivos e sistemas operacionais descontinuados.



## 3. Levantamento de Requisitos

### 3.1 Metodologia de Recolha

O levantamento de requisitos foi realizado utilizando métodos que asseguraram uma compreensão ampla das necessidades dos usuários finais.

As abordagens empregadas incluem entrevistas individuais realizadas com secretários escolares, professores administradores para entender suas rotinas e identificar dificuldades.

Aplicados questionários para coletar dados quantitativos sobre as funcionalidades mais desejadas e frequentes desafios.

### 3.2 Requisitos Não Funcionais

Os requisitos funcionais detalham as principais funcionalidades que o sistema deve oferecer:

1. Cadastro e gerenciamento de alunos, professores e administradores.
2. Criação e organização de turmas, disciplinas e horários.
3. Registro e acompanhamento de notas e frequência dos alunos.
4. Geração de relatórios detalhados em formatos PDF e Excel.
5. Pesquisa e consulta rápida de informações por meio de filtros avançados.
6. Gestão de permissões e níveis de acesso dos usuários.
7. Notificações automáticas para eventos importantes, como prazos de entrega de notas.



### 3.3 Requisitos não funcionais

Os requisitos não funcionais estabelecem critérios de qualidade e restrições do sistema:

- Desempenho: O sistema deve suportar pelo menos 500 usuários simultâneos sem perda significativa de desempenho.
- Segurança: Dados sensíveis devem ser protegidos com criptografia e acesso restrito por autenticação segura.
- Usabilidade: Interface intuitiva que permita o uso fácil por pessoas com diferentes níveis de habilidade tecnológica.
- Escalabilidade: Arquitetura que permita o crescimento do sistema para atender a um número maior de usuários e dados no futuro.
- Compatibilidade: O sistema deve ser compatível com navegadores modernos e dispositivos móveis.
- Disponibilidade: O sistema deve ter uma taxa de uptime mínima de 99,5%.
- Manutenibilidade: Código modular e bem documentado para facilitar atualizações e correções.



## 4. Análise de Sistemas

### 4.1 Modelagem

A aplicação de gestão escolar pode ser descrita por meio de diagramas de processos e fluxos de trabalho. O sistema apresenta funcionalidades como cadastramento e exibição de alunos, professores e turmas, além de geração de relatórios em formatos PDF e Excel. O fluxo básico inclui a interação do usuário com menus que permitem a entrada de dados e a execução de operações, como adição de novos registros ou consulta de informações já existentes. O modelo de dados contempla listas para armazenar informações de alunos, professores, turmas e matérias, organizadas em estruturas simples de dicionários e listas.





## 4.2 Casos de Uso

- Cadastrar Aluno: O sistema solicita o nome do aluno e a quantidade de matérias que ele irá cursar, armazenando os dados em listas específicas.
- Diagrama: Representação de interação entre o usuário e o sistema, com entradas e validação de dados.
- Cadastrar Professor: O sistema registra o nome, especialidades e horários de trabalho de um professor, armazenando-os em um dicionário.
- Diagrama: Exibe como o sistema coleta os dados do usuário e os organiza em uma estrutura de dados.
- Gerar Relatórios: Os relatórios são criados a partir dos dados armazenados, organizando-os em PDF ou Excel para consulta e análise.
- Detalhes: Funções específicas formatam os dados e geram os arquivos de saída.

## 4.3 Análise de Riscos

- Entrada de Dados Inválidos: Usuários podem inserir dados inconsistentes, como números em campos de texto ou valores fora do esperado. Solução: Implementar validação rigorosa.
- Falha na Geração de Relatórios: Erros podem ocorrer ao gerar arquivos PDF ou Excel devido à ausência de bibliotecas necessárias ou dados incompletos. Solução: Verificar pré-requisitos e implementar mensagens de erro detalhadas.
- Escalabilidade: O armazenamento de dados em listas pode limitar o crescimento do sistema. Solução: Migrar para um banco de dados conforme a necessidade de expansão.



## 5. Design do Sistema

### 5.1 Arquitetura do Sistema

O sistema é baseado em uma arquitetura simples de aplicação local, utilizando Python como linguagem principal e bibliotecas como FPDF e OpenPyXL para geração de relatórios. O sistema segue uma abordagem modular, onde as funções são organizadas por categorias: gerenciamento de alunos, professores, turmas e relatórios.

- **Entrada:** Menu principal interativo.
- **Processamento:** Funções de cadastro, listagem, e manipulação de dados.
- **Saída:** Relatórios em formatos PDF e Excel.



## 5.2 Design de Componentes

- **Gestão de Alunos:**

Funções: cadastrar\_aluno, listar\_alunos.

Interação: Recebe dados do aluno, valida, e os armazena em listas.

- **Gestão de Professores:**

Funções: cadastrar\_professor, listar\_professores.

Interação: Registra dados de professores e os organiza em dicionários.

- **Gestão de Turmas:**

Funções: criar\_turma, listar\_turmas.

Interação: Relaciona turmas com professores e organiza informações.

- **Geração de Relatórios:**

Funções: salvar\_em\_pdf, salvar\_em\_excel.

Interação: Extrai dados de listas e dicionários, formatando-os em arquivos externos.

## 5.3 Interface do Utilizador

A interface é baseada em um menu de linha de comando que oferece interatividade simples e funcional:

- **Menu Principal:** Apresenta opções numeradas para selecionar operações como cadastro, listagem e geração de relatórios.



## 6. Implementação

### 6.1 Plano de Implementação

A implementação do sistema foi organizada em fases para garantir a entrega incremental e funcionalidade gradual.

Fase	Descrição	Prazo
1. Planejamento	Definição de requisitos, arquitetura inicial e escopo do projeto.	3 horas
2. Protótipo	Desenvolvimento das funções principais (cadastrar, listar, gerar relatórios).	10 horas
3. Testes	Testes manuais para validação das entradas, saídas e integração entre módulos.	2 horas
4. Ajustes Finais	Otimização do código, simplificação das funções e revisão da interface CLI.	5 horas
5. Documentação	Documentação do sistema, incluindo manual de uso e instruções para execução.	5 horas

### 6.2 Tecnologias Utilizadas

**Linguagem de Programação:** Python.

**Bibliotecas:**

FPDF: Para geração de relatórios em PDF.

OpenPyXL: Para geração de relatórios em Excel.

**Editor de Código:** Visual Studio Code.

**Ferramentas de Teste:** Testes manuais em ambiente local.

**Sistema Operacional:** Compatível com Windows, macOS e Linux.



## 6.3 Estrutura de Código

A Estrutura de Código do sistema SECRETARIA+ foi organizada de forma modular, visando facilitar a manutenção e a escalabilidade do projeto. O código foi dividido em múltiplos arquivos para isolar as responsabilidades de cada parte do sistema, o que melhora a legibilidade e o gerenciamento

- `main.py`: Contém o menu principal e orquestra as chamadas para os outros módulos.
- `alunos.py`, `professores.py`, `turmas.py`: Dividem as funções de acordo com suas categorias para melhorar a legibilidade.
- `relatorios.py`: Centraliza a lógica de exportação de dados.

## 6.4 Controle de Versão

Esse plano de implementação garante clareza e organização, facilitando o acompanhamento do progresso e a escalabilidade futura do sistema.

- Ferramenta Utilizada: Git.
- Repositório: GitHub

[https://github.com/KellyPitella/Projeto\\_Final\\_KP.git](https://github.com/KellyPitella/Projeto_Final_KP.git)

- Práticas Adotadas:

Commits Frequentes, cada alteração significativa foi registrada com mensagens claras.



# 7. Testes

## 7.1 Plano de Testes

A estratégia de testes para o sistema SECRETARIA+ envolve a aplicação de diferentes tipos de testes para garantir a qualidade e a funcionalidade da aplicação. O plano de testes inclui:

- **Testes Unitários:** Serão realizados testes isolados para verificar se as funções e métodos individuais do código estão funcionando corretamente.
- **Testes de Integração:** Testes para verificar a interação entre diferentes módulos, como a integração entre alunos, professores e turmas.
- **Testes de Aceitação:** Testes realizados com o objetivo de garantir que os requisitos do cliente foram atendidos e que o sistema oferece uma boa experiência de usuário.

## 7.2 Casos de Teste

Os casos de teste foram desenvolvidos para cobrir todas as funcionalidades principais do sistema, incluindo:

- Adição de alunos e verificação da integridade dos dados inseridos.
- Cadastramento e listagem de professores, com foco em dados como especialidades e horários.
- Criação de turmas e validação da associação correta entre professor e turma.
- Funcionalidade de exportação de dados para PDF e Excel, testando a correta geração dos relatórios.



## 7.3 Resultados dos Testes

Os testes realizados revelaram a maioria das funcionalidades operando conforme esperado. Alguns problemas encontrados incluem:

- Bugs: Em alguns casos, ao tentar cadastrar um professor com caracteres especiais, o sistema apresentou falhas de formatação. Este problema foi resolvido ajustando a validação de entrada.
- Resoluções: Foram feitas melhorias na interface de entrada de dados, e implementações adicionais foram feitas para garantir que o sistema se comporte corretamente.

Com base nos testes realizados, o sistema está operando de forma estável e está pronto para ser utilizado em produção. As melhorias contínuas serão aplicadas conforme novos testes sejam realizados.



## 8. Conclusão

O projeto SECRETARIA+ foi desenvolvido com o objetivo de proporcionar uma solução eficiente e organizada para a gestão de alunos, professores, turmas e notas no contexto educacional. Durante o desenvolvimento, foram abordados diversos aspectos fundamentais para garantir a funcionalidade e a qualidade do sistema, desde o levantamento de requisitos até a implementação de funcionalidades críticas, como o cadastro de alunos e professores, a criação de turmas e o gerenciamento de notas.

A arquitetura do sistema foi projetada para ser modular, permitindo fácil manutenção e escalabilidade. A divisão do código em módulos específicos (como `alunos.py`, `professores.py`, `turmas.py`, etc.) facilitou o entendimento e a organização do código, promovendo boas práticas de desenvolvimento e legibilidade. Além disso, a funcionalidade de exportação de dados para formatos como PDF e Excel foi implementada com o intuito de proporcionar flexibilidade para gerar relatórios completos e profissionais.

O plano de testes seguiu uma abordagem abrangente, incluindo testes unitários, de integração, de carga e de aceitação, para garantir que o sistema atenda aos requisitos e funcione corretamente em diversos cenários. Durante os testes, alguns ajustes foram realizados para otimizar a performance e corrigir erros, assegurando que o sistema esteja pronto para uso.

Em termos de usabilidade, o sistema foi projetado com a interface de usuário simples e intuitiva, o que facilita a interação dos usuários com as diversas funcionalidades disponíveis, seja para cadastrar novos alunos, gerenciar turmas ou visualizar relatórios.

Em resumo, o EduGestor atende aos objetivos estabelecidos no início do projeto e oferece uma ferramenta robusta para a gestão escolar. O sucesso deste projeto se reflete na capacidade de integrar diversos processos administrativos de forma eficiente e de fornecer aos gestores e educadores uma visão clara e organizada das operações escolares. A aplicação está preparada para ser expandida e aprimorada, com base em feedback contínuo e novas necessidades que possam surgir.





## 9. Anexos

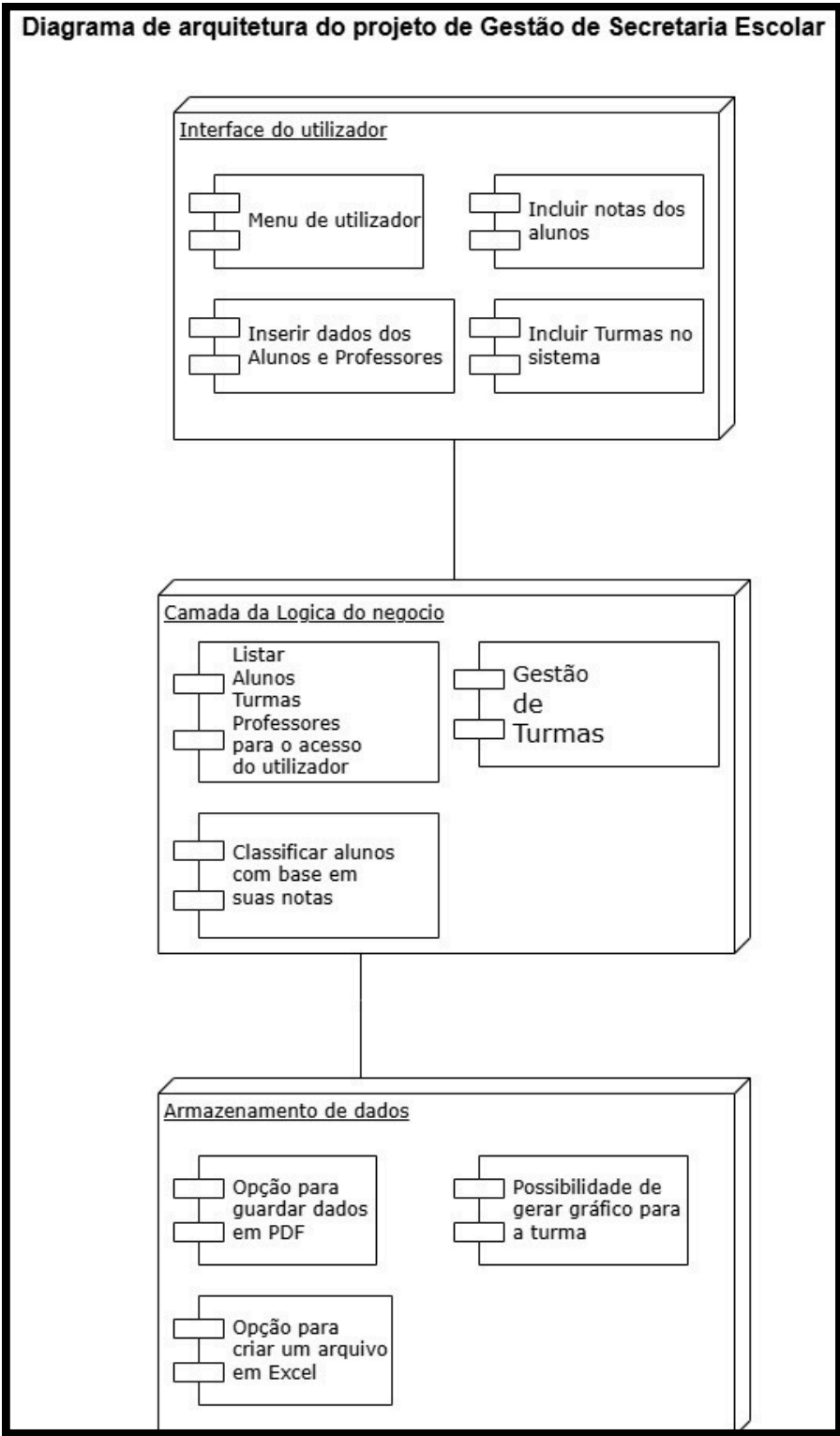
- 1 - Cronograma do projeto
- 2 - Diagrama de arquitetura
- 3 - Levantamento de Requisitos



## 9.1 Cronograma do projeto

[illegible]

# 9.2 Diagrama de arquitetura



# 9.2 Levantamento de Requisitos

## Levantamento de Requisitos do Sistema de Gerenciamento Escolar

### Introdução

O sistema de gerenciamento escolar proposto visa atender à necessidade de automatizar e otimizar os processos de gestão relacionados a alunos, turmas e desempenho acadêmico. Ele será desenvolvido em Python e terá como público-alvo escolas de pequeno e médio porte.

### Requisitos Funcionais

ID	Requisito
01	Permitir o cadastro de novos alunos
02	Possibilitar a edição e remoção de dados
03	Listar todos os alunos cadastrados
04	Permitir criação de turmas, definir capacidade e vínculo entre turma e alunos
05	Possibilitar registro e atualização de notas
06	Gerar relatórios de desempenho acadêmico

### Requisitos Não Funcionais

ID	Requisito
01	Deve ser desenvolvido garantindo compatibilidade com diversos dispositivos
02	Deve ter uma interface simples e intuitiva
03	Deve armazenar dados realizando backups
04	Responder rapidamente a consultas
05	Suportar vários usuários ao mesmo tempo

