

MANUAL DO PRODUTO

Academix

Aplicação desktop de gestão escolar em Java Swing,
modelada com Programação Orientada a Objetos.

VERSÃO

1.0

DATA

Agosto de 2026

BASE

Java Swing · POO · persistência serializada

github.com/eduardofabrii/academix

Telas reproduzidas a partir dos arquivos `.form` do repositório, renderizados com o mecanismo de layout real do GUI Designer (forms_rt). Nenhum valor foi inventado.

O que há neste manual

01	Visão geral do produto	3
02	Modelo de domínio e herança	4
03	Como executar o projeto	5
04	Menu principal	6
05	Painel de Cadastro	7
06	Cadastro de Aluno	8
07	Cadastro de Professor	9
08	Cadastro de Funcionários	10
09	Cadastro de Disciplina	11
10	Cadastro de Turma	12
11	Cadastro de Sala e de Livro	13
12	Entrar Como — seleção de papel	14
13	Painel do Professor	15
14	Painel do Aluno	16
15	Painel do Diretor	17
16	Regras de negócio e decisões de UX	18
17	Estado do projeto e limitações	19

As páginas de tela seguem a mesma estrutura: título e descrição, captura anotada com legendas numeradas, cartões de funcionalidades, tabela de campos e uma observação de comportamento.

Visão geral do produto

O **Academix** simula a administração de uma escola: cadastro de **alunos, professores, funcionários, disciplinas, turmas, salas e livros**, e operação por papel (professor lança nota, aluno vê boletim, diretor emite relatório). É um aplicativo **desktop** em Java Swing, sem servidor e sem banco de dados relacional — o estado é gravado em arquivos serializados (o “minibanco”). O README declara: *“Não foi finalizado”*.



Domínio

Escola, Turma, Disciplina, Boletim, Nota, Biblioteca, Livro, Sala, Cardápio.



Pessoas

Aluno (e Monitor), Responsável e oito papéis de Funcionário.



Dois fluxos

Cadastro (dados mestres) e “Entrar Como” (operação por perfil).



Interface

Janelas Swing de tamanho fixo, construídas no GUI Designer da IDE (arquivos .form).

EM UMA FRASE POR CAMADA

CAMADA	TECNOLOGIA	PAPEL
Interface	Java Swing + GUI Designer	~28 telas <code>.form</code> ligadas a classes <code>*UI</code> .
Domínio	POO (herança, classes abstratas)	Pessoas e recursos da escola; regras via exceções próprias.
Persistência	Serialização Java (<code>.ser</code>)	“Minibanco”: <code>dadosPessoa.ser</code> , <code>dadosboletim.ser</code> , etc.
Execução	JVM desktop	<code>EscolaPrincipalUI.main()</code> abre a janela inicial.



OBSERVAÇÃO

Como o projeto é inacabado, algumas telas têm arestas conhecidas (rótulo branco sobre fundo branco, ausência de validação de formato, telas de submenu sem lista dos registros já criados). Estão sinalizadas nas páginas correspondentes e na página **Estado do projeto**.

Modelo de domínio e herança

O coração do Academix é a hierarquia de classes. Tudo que é “gente” deriva de `Pessoa`; funcionários acrescentam salário e ponto; alunos acrescentam boletim e turma.



REGRAS EXPRESSAS COMO EXCEÇÕES

EXCEÇÃO	QUANDO É LANÇADA
<code>DisciplinaInvalidaException</code>	Relatório de desempenho para uma disciplina que o professor não leciona.
<code>TurmaSemAlunosException</code>	Operação de turma sem nenhum aluno vinculado.
<code>BoletimNaoRegistradoException</code>	Consulta de notas/boletim de um aluno sem boletim criado.



OBSERVAÇÃO

A persistência é feita por serialização: as classes de domínio implementam `Serializable` e o pacote `minibanco.lê/grava` arquivos `.ser` (um por área: pessoas, boletim, administração, salas).

Como executar o projeto

O projeto é um módulo IntelliJ IDEA (sem Maven/Gradle). As telas são arquivos `.form` do GUI Designer, que a IDE tece no bytecode ao compilar. O caminho natural é abrir na IDE e rodar a classe principal.

1 · ABRIR E EXECUTAR

1. Abrir a pasta `academix` como projeto no IntelliJ IDEA.
2. Garantir um JDK configurado (o código usa `java.time` e genéricos — JDK 8+).
3. Executar `escola.principal.EscolaApp` para o teste de domínio no console, ou `escola.UI.EscolaPrincipalUI` para abrir a interface.

2 · TESTE DE DOMÍNIO (CONSOLE)

Saída real de `EscolaApp.main()` nesta execução:

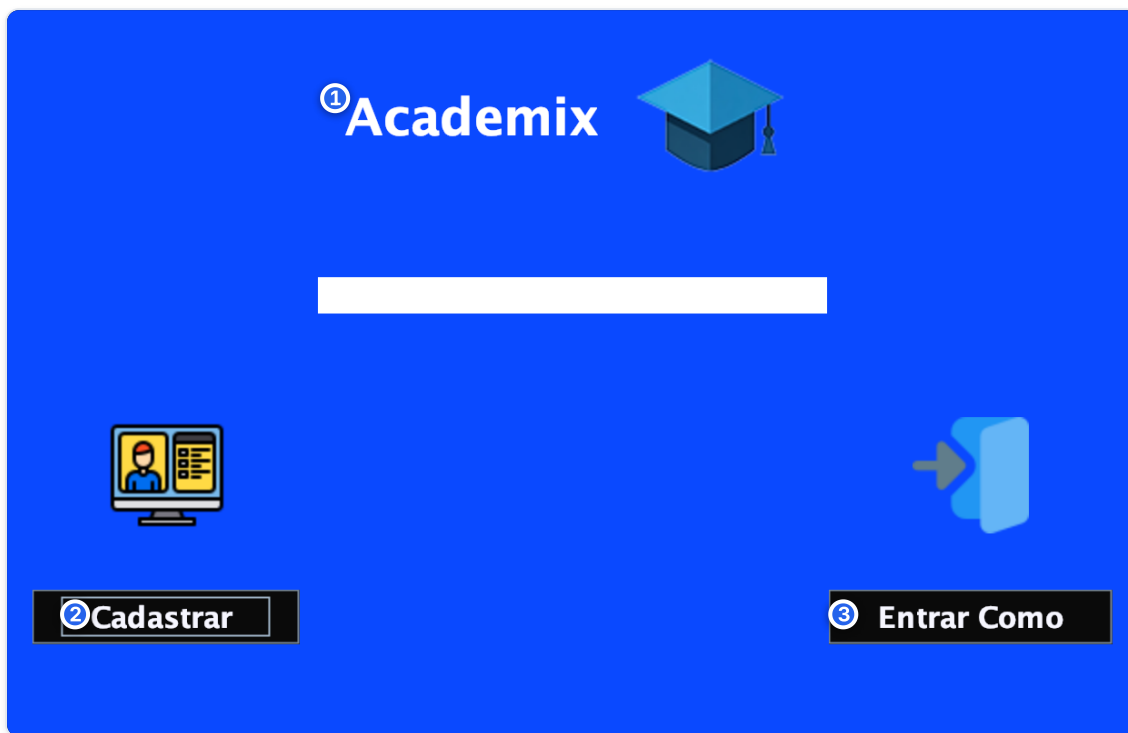
```
● ● ● escola.principal.EscolaApp  
  
Disciplina P00 adicionada ao professor Marina de Lara  
Nota registrada: 10.0  
Aluno Eduardo Henrique Fabri adicionado à turma A.  
Nota total da turma: 10.0  
Nota média da turma: 10.0
```

OBSERVAÇÃO

As telas deste manual foram reproduzidas lendo os arquivos `.form` e montando a árvore Swing com o mesmo mecanismo de layout do GUI Designer (`com.intellij.uiDesigner.core` , de `forms_rt.jar`). Textos, cores, fontes, ícones e tamanhos vêm dos próprios `.form` .

Menu principal

Janela inicial (`EscolaPrincipalUI`). Dois caminhos: **Cadastrar** abre o painel de cadastros do sistema; **Entrar Como** abre a seleção de papel para usar as funções de cada perfil.



- 1 Identidade do produto em texto: "Academix" + ícone de capelo.
- 2 Botão **Cadastrar** — leva ao Painel de Cadastro.
- 3 Botão **Entrar Como** — leva à seleção de papel.



Duas áreas

Cadastro (dados mestres) e Entrar Como (operação por papel) são os dois ramos do sistema.



Janelas Swing

Cada ação abre um novo `JFrame` de tamanho fixo, não redimensionável.



Sem autenticação

"Entrar Como" é escolha de perfil, não login: não há usuário nem senha.

CAMPOS E CONTROLES

CAMPO / CONTROLE	TIPO	OBRIGATÓRIO	OBSERVAÇÕES
Cadastrar	Botão	—	Abre <code>CadastroUI</code> (Painel de Cadastro).
Entrar Como	Botão	—	Abre <code>EntrarComoUI</code> (seleção de papel).



OBSERVAÇÃO

A janela abre com `setSize(600, 350)` e `setResizable(false)`. O rótulo branco central do formulário fica sobre fundo branco e por isso não aparece — detalhe herdado do arquivo `.form`.

Painel de Cadastro

Concentra o cadastro dos dados mestres da escola: pessoas (funcionários e alunos) e recursos acadêmicos (sala, turma, disciplina, livro).



- 1 Título "Painel de Cadastro".
- 2 Coluna esquerda: Funcionários, Sala, Disciplina.
- 3 Coluna direita: Aluno, Turma, Livro.



Pessoas

Cadastrar Funcionarios abre o submenu de papéis; **Cadastrar Aluno** vai direto ao formulário.



Recursos

Sala, Turma, Disciplina e Livro são os recursos acadêmicos cadastráveis.



Padrão visual

Todos os itens são botões grandes com ícone; layout em grade de 2 colunas.

CAMPOS E CONTROLES

CAMPO / CONTROLE	TIPO	OBRIGATÓRIO	OBSERVAÇÕES
Cadastrar Funcionarios	Botão	—	Abre <code>CadastroFuncionariosUI</code> .
Cadastrar Aluno	Botão	—	Abre <code>CadastroAlunoUI</code> .
Sala · Turma · Disciplina · Livro	Botões	—	Abrem os formulários correspondentes em <code>cadastro/outros</code> .



OBSERVAÇÃO

É o "hub" de dados mestres. A operação do dia a dia (lançar nota, ver boletim, gerar relatório) fica do outro lado, em **Entrar Como**.

Cadastro de Aluno

Formulário para criar um aluno e vinculá-lo a uma turma. Campos de identificação pessoal mais matrícula e turma.

Cadastro Aluno

1

2

3

4

Nome

CPF

Nascimento

Endereço

Matrícula

Turma

Cadastrar

↺

- 1 Título e ícone da entidade sendo cadastrada.
- 2 Campos de texto: Nome, CPF, Nascimento, Endereço, Matrícula.
- 3 Turma: combo alimentado com as turmas já cadastradas.
- 4 Cadastrar grava; o botão ↺ ao lado recarrega / limpa.



Campos livres

Nome, CPF, Nascimento, Endereço e Matrícula são `UITextField` sem máscara.



Vínculo

A turma é escolhida por `JComboBox` — o aluno nasce já associado a uma turma.



Ações

Cadastrar persiste no “minibanco” serializado; ↺ atualiza a tela.

CAMPOS E CONTROLES

CAMPO / CONTROLE	TIPO	OBRIGATÓRIO	OBSERVAÇÕES
Nome	Texto	Sim	Sem validação de formato.
CPF	Texto	Sim	Campo livre; não há verificação de dígitos nem de duplicidade na tela.
Nascimento	Texto	Sim	Data digitada como texto (sem seletor de calendário).
Endereço · Matrícula	Texto	Sim	Livres.
Turma	Combo	Sim	Lista as turmas cadastradas; vazio se nenhuma turma existe.

⚠ OBSERVAÇÃO

A data de nascimento é texto puro. O domínio usa `LocalDate`, então o formato aceito depende de como o controlador converte a string — cadastre datas no padrão ISO (`AAAA-MM-DD`) para evitar erro.

Cadastro de Professor

Cadastro de um professor (subclasse de `Funcionario`). Mesma base de dados pessoais, acrescida de salário e disciplina.

- 1 Título "Cadastro Professor" e ícone do papel.
- 2 Dados pessoais + salário (campo de funcionário).
- 3 Cadastrar / atualizar.



É um Funcionário

`Professor` extends `Funcionario` extends `Pessoa`: herda dados pessoais e ponto.



Disciplina

O professor pode lecionar uma disciplina (definido depois, no Painel do Professor).



Campos

Nome, CPF, Nascimento, Endereço, Salário — todos texto livre.

CAMPOS E CONTROLES

CAMPO / CONTROLE	TIPO	OBRIGATÓRIO	OBSERVAÇÕES
Nome · CPF · Nascimento · Endereço	Texto	Sim	Herdados de <code>Pessoa</code> .
Salário	Texto	Sim	Específico de <code>Funcionario</code> ; valor numérico digitado como texto.
Cadastrar ·	Botões	—	Gravar / recarregar.



OBSERVAÇÃO

Os demais papéis de funcionário (Diretor, Secretário, Pedagogo, Bibliotecário, Porteiro, Faxineiro, Merendeiro) usam formulários equivalentes — mesma base de `Pessoa` + `salário`.

Cadastro de Funcionários

Submenu que escolhe qual tipo de funcionário cadastrar. Cada botão abre o formulário específico daquele papel.



- 1 Título "Cadastro de Funcionarios".
- 2 Coluna esquerda: Professor, Pedagogo, Secretario, Porteiro.
- 3 Coluna direita: Faxineiro, Bibliotecario, Diretor, Merendeiro.



Oito papéis

Reflete diretamente as oito subclasses concretas de `Funcionario`.



Um formulário por papel

Cada botão abre `CadastroXxxUI` (ex.: `CadastroDiretorUI`).



Sem lista

A tela não mostra os funcionários já cadastrados — é só um seletor de tipo.

CAMPOS E CONTROLES

CAMPO / CONTROLE	TIPO	OBRIGATÓRIO	OBSERVAÇÕES
Professor · Pedagogo · Secretario · Porteiro	Botões	—	Abrem os formulários da coluna esquerda.
Faxineiro · Bibliotecario · Diretor · Merendeiro	Botões	—	Abrem os formulários da coluna direita.



OBSERVAÇÃO

A modelagem separa "pessoas da administração/ensino" (Professor, Pedagogo, Diretor, Secretário, Bibliotecário) de "apoio" (Porteiro, Faxineiro, Merendeiro), mas todas herdam de `Funcionario`.

Cadastro de Disciplina

Cria uma disciplina — nome, carga horária e o professor responsável.

Cadastro Disciplina

①

Nome

②

Carga Horár

Sala de Aula

▼

Professor

▼

③

Cadastrar

- 1 Título "Cadastro Disciplina".
- 2 Campos: nome da disciplina e carga horária.
- 3 Cadastrar / ↺.



Dados

Nome e carga horária (horas). A disciplina entra no domínio como `Disciplina`.



Professor

A disciplina é associada a um professor que a leciona (`lecionarDisciplina`).



Uso

Disciplinas cadastradas alimentam o combo do **Cadastro de Turma**.

CAMPOS E CONTROLES

CAMPO / CONTROLE	TIPO	OBRIGATÓRIO	OBSERVAÇÕES
Nome	Texto	Sim	Identifica a disciplina (ex.: "POO").
Carga horária	Texto	Sim	Número de horas, digitado como texto.
Cadastrar · ↺	Botões	—	Gravar / recarregar.



OBSERVAÇÃO

No teste de domínio embutido (`EscoLaApp`), a disciplina "POO" com 120 h é atribuída à professora "Marina de Lara" — ver a página **Modelo de domínio**.

Cadastro de Turma

Cria uma turma identificada por nome e ano, ligada a uma disciplina.



- 1 Título "Cadastro Turma".
- 2 Campos: **Nome da turma** e **Ano**.
- 3 **Disciplina**: combo com as disciplinas cadastradas.
- 4 **Cadastrar** / .



Identificação

Nome (ex.: "A") e Ano definem a turma.



Disciplina

Escolhida em `JComboBox`; a turma referencia a disciplina.



Alunos

A turma recebe alunos depois, pelo **Cadastro de Aluno** (combo Turma).

CAMPOS E CONTROLES

CAMPO / CONTROLE	TIPO	OBRIGATÓRIO	OBSERVAÇÕES
Nome da turma	Texto	Sim	Curto (ex.: "A", "3B").
Ano	Texto	Sim	Ano letivo, digitado como texto.
Disciplina	Combo	Sim	Lista as disciplinas cadastradas.



OBSERVAÇÃO

Os rótulos aparecem com um leve fundo colorido (definido no `.form`) — não são campos, apenas etiquetas.

Cadastro de Sala e de Livro

Dois formulários de recursos físicos. **Sala** tem tipo (aula ou biblioteca — `Sala` é abstrata com duas subclasses). **Livro** alimenta o acervo da biblioteca.

Cadastro Sala



Nome da Sala

Capacidade

ID

Cadastrar

CadastroSalaUI

Cadastro Livro



Título

Autor

Ano de publicação

Cadastrar

CadastroLivroUI



Sala abstrata

`SalaAula` e `SalaBiblioteca` herdam de `Sala` (`Serializable`).



Livro → Biblioteca

Livros cadastrados compõem o acervo consultado pelo Bibliotecário.



Campos simples

Sala: número/capacidade/tipo. Livro: título e dados de exemplar.

CAMPOS E CONTROLES

CAMPO / CONTROLE	TIPO	OBRIGATÓRIO	OBSERVAÇÕES
Sala	Formulário	—	Número, capacidade e tipo (aula / biblioteca).
Livro	Formulário	—	Título e identificação do exemplar; entra em <code>Biblioteca</code> .

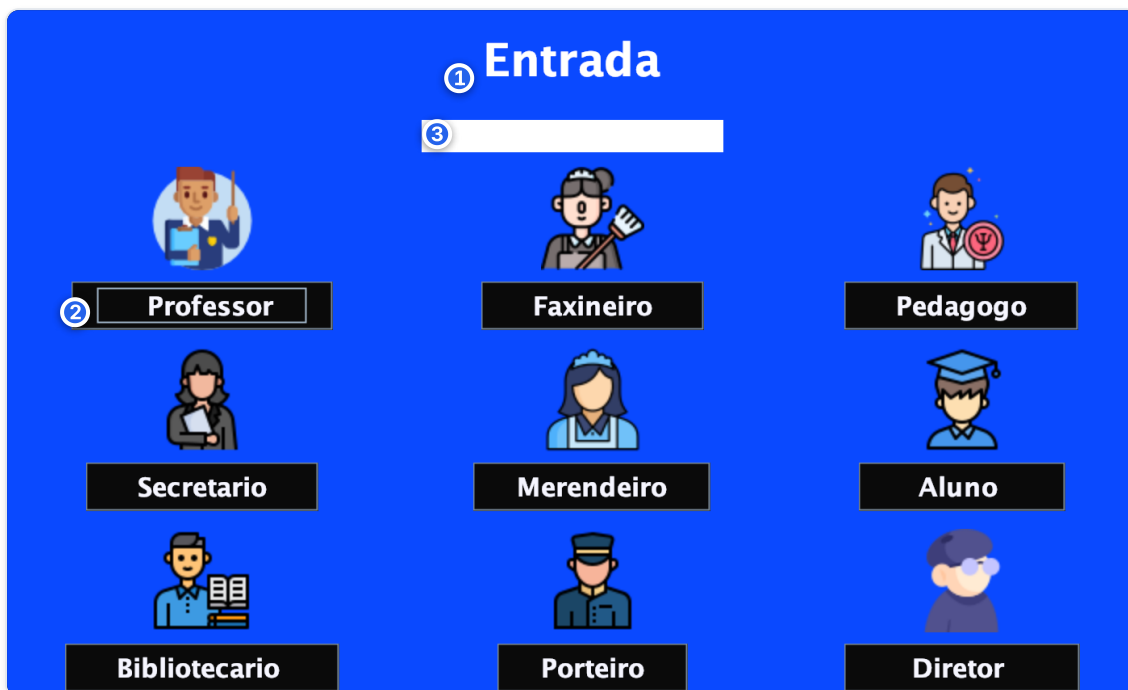


OBSERVAÇÃO

São os únicos recursos “físicos” do modelo. O restante do domínio (Boletim, Nota, Cardápio) é gerado a partir da operação, não cadastrado nessas telas.

Entrar Como — seleção de papel

Escolhe com qual papel operar o sistema. Cada botão abre o painel de operações daquele perfil (Painel do Professor, do Aluno, etc.).



- 1 Título "Entrada".
- 2 Grade de papéis: Professor, Faxineiro, Pedagogo, Secretario, Merendeiro, Aluno, Bibliotecario, Porteiro, Diretor.
- 3 Rótulo "Escolha o que quer ser" (texto branco sobre fundo branco — não visível).



Papel = visão

Não há credencial: o papel escolhido só determina quais funções aparecem em seguida.



Nove papéis

Um botão por perfil de pessoa do sistema, cada um com seu ícone.



Próximo passo

O painel aberto pede "Qual <papel> você é" num combo, para carregar os dados da pessoa.

CAMPOS E CONTROLES

CAMPO / CONTROLE	TIPO	OBRIGATÓRIO	OBSERVAÇÕES
Professor · Aluno · Diretor · ...	Botões	—	Abrem os painéis ProfessorUI, AlunoUI, DiretorUI, etc.



OBSERVAÇÃO

A separação é intencional: **Cadastrar** cria as pessoas e recursos; **Entrar Como** as usa. Não é possível operar um perfil sem antes cadastrar a pessoa correspondente.

Painel do Professor

Operações do professor sobre notas, disciplinas e turmas. Começa escolhendo, num combo, qual professor está operando.

- 1 Título "Painel do Professor".
- 2 "Qual professor você é" — combo que carrega a pessoa.
- 3 Ações: Atribuir Nota, Consultar Nota, Lecionar Disciplina, Consultar Turma.
- 4 Gerar Relatorio Desempenho — relatório da turma numa disciplina.

Atribuir / Consultar Nota
Lança e consulta notas de alunos (entra no **Boletim** de cada aluno).

Lecionar Disciplina
Associa o professor a uma disciplina que ele passa a lecionar.

Relatório de desempenho
Calcula nota total e média da turma numa disciplina — pode lançar exceções de regra.

CAMPOS E CONTROLES

CAMPO / CONTROLE	TIPO	OBRIGATÓRIO	OBSERVAÇÕES
Qual professor você é	Combo	Sim	Sem seleção, as ações não têm sobre quem operar.
Atribuir / Consultar Nota	Botões	—	Notas por aluno + disciplina.
Lecionar Disciplina · Consultar Turma	Botões	—	Vínculo com disciplina / listagem da turma.
Gerar Relatorio Desempenho	Botão	—	Total e média da turma.



OBSERVAÇÃO

O relatório usa as exceções do domínio: `DisciplinaInvalidaException`, `TurmaSemAlunosException`, `BoletimNaoRegistradoException`. São validações de regra de negócio, não de formulário.

Painel do Aluno

Visão do aluno sobre seu próprio desempenho: boletim, frequência, média e turma.

- 1 Título "Painel do Aluno".
- 2 "Qual aluno você é" — combo de seleção.
- 3 Ações: Visualizar Boletim, Consultar Frequencia, Calcular Média, Consultar Turma.

Boletim Mostra as notas registradas para o aluno (objeto <code>Boletim</code> com <code>Nota</code>s).	Média e frequência Calcula a média das notas e consulta a frequência do aluno.	Turma "Consultar Turma" mostra a turma à qual o aluno está vinculado.
---	---	--

CAMPOS E CONTROLES

CAMPO / CONTROLE	TIPO	OBRIGATÓRIO	OBSERVAÇÕES
Qual aluno você é	Combo	Sim	Define de quem são o boletim e a frequência exibidos.
Visualizar Boletim · Calcular Média	Botões	—	Leitura do <code>Boletim</code> .
Consultar Frequencia · Consultar Turma	Botões	—	Consultas somente leitura.

⚠️ OBSERVAÇÃO

Sem boletim registrado, "Visualizar Boletim" tende a acionar `BoletimNaoRegistradoException` — a regra do domínio exige um boletim criado antes da consulta.

Painel do Diretor

Funções de gestão: acompanhar professores e alunos, conduzir reuniões e emitir relatórios gerais e de desempenho de turma.

- 1 Título "Painel do Diretor".
- 2 "Qual diretor você é" — combo de seleção.
- 3 Gerenciar Professores / Alunos, Realizar Reunião, Relatório Geral.
- 4 Avaliar Desempenho de Turma.

Gestão de pessoas Acompanha os cadastros de professores e alunos da escola.	Relatórios "Relatório Geral" e "Avaliar Desempenho de Turma" consolidam informações acadêmicas.	Reunião "Realizar Reunião" é uma ação de processo do papel de direção.
--	--	---

CAMPOS E CONTROLES

CAMPO / CONTROLE	TIPO	OBRIGATÓRIO	OBSERVAÇÕES
Qual diretor você é	Combo	Sim	Seleciona a pessoa que opera o painel.
Gerenciar Professores / Alunos	Botões	—	Acompanhamento dos cadastros.
Realizar Reunião · Relatório Geral · Avaliar Desempenho de Turma	Botões	—	Ações de gestão e relatório.

⚠️ OBSERVAÇÃO

Bibliotecário, Secretário, Pedagogo, Porteiro, Faxineiro, Merendeiro, Responsável e Monitor têm painéis equivalentes, cada um com o conjunto de ações do seu papel.

Regras de negócio e decisões de UX

Consolidação do que se observa nas telas e no código de domínio.

TEMA	COMPORTAMENTO OBSERVADO
Identificação	"Entrar Como" é escolha de papel, não login. Não há usuário, senha nem controle de acesso real.
Papéis	Nove perfis de pessoa (8 funcionários + aluno), cada um com um painel de ações próprio.
Cadastro x operação	Dados mestres entram por Cadastrar ; nota, boletim, relatório e reunião ficam em Entrar Como .
Vínculos	Aluno nasce ligado a uma turma; turma referencia uma disciplina; disciplina referencia um professor.
Validação de formulário	Mínima: campos são <code>TextField</code> livres, sem máscara de CPF/data nem checagem de duplicidade na tela.
Validação de regra	Feita no domínio, por exceções (disciplina inválida, turma sem alunos, boletim não registrado).
Datas	Digitadas como texto; o domínio usa <code>LocalDate</code> — usar <code>AAAA-MM-DD</code> .
Persistência	Serialização Java. Cada área grava um <code>.ser</code> ; apagar o arquivo zera aquela área.
Janelas	Tamanho fixo (<code>setResizable(false)</code>); cada ação abre um novo <code>JFrame</code> .
Aparência	Fundo azul, fonte Calibri, ícones coloridos. Alguns rótulos ficam invisíveis (branco sobre branco).

⚠ OBSERVAÇÃO

O sistema não tem trilha de auditoria, backup nem migração de dados. Para uso além de estudo, a persistência serializada precisaria ser trocada por um banco e os formulários ganhariam validação e máscaras.

Estado do projeto e limitações

O próprio README marca o Academix como **não finalizado**. Ele funciona como demonstração de POO e de construção de telas Swing, mas não como produto pronto para uso escolar real.

LIMITAÇÕES CONHECIDAS

1. Sem autenticação ou controle de acesso — “Entrar Como” apenas troca a visão.
2. Formulários sem validação de formato (CPF, data) nem verificação de duplicidade.
3. Submenus (Cadastro de Funcionários, Painel de Cadastro) não listam os registros já criados.
4. Rótulos com texto branco sobre fundo branco em várias telas (herdado dos `.form`).
5. Persistência frágil: arquivos `.ser`, sensíveis a mudança de classe e sem backup.
6. Fluxos de operação (lançar nota, gerar relatório) dependem de dados previamente cadastrados e podem lançar exceções de regra se a ordem não for seguida.

COMO ESTE DOCUMENTO FOI PRODUZIDO

1. Clonagem do repositório e leitura do código (domínio + classes `*UI` + arquivos `.form`).
2. Renderização de cada `.form` com o mecanismo de layout do GUI Designer (`forms_rt.jar`), preservando textos, cores, fontes e ícones do arquivo.
3. Execução real de `EscolaApp.main()` para capturar a saída do modelo de domínio.
4. Anotação das regras de negócio e decisões de UX visíveis nas telas e no código.

github.com/eduardofabii/academix

Autores: Eduardo Henrique Fabri · João Vitor Correa · Renan Herculano

Stack: Java · Swing · GUI Designer (`.form`) · serialização · POO

Documento de referência — Versão 1.0 — Agosto de 2026