

JDBCOURSE

Manual Detalhado de Instalação, Configuração e Uso do
Sistema de Gerenciamento de Cursos Online

Autor:

Gabriel Levi do Nascimento Costa

Universidade de Fortaleza
Curso de Ciência da Computação

Fortaleza - CE
Junho de 2025

Conteúdo

1	Introdução	2
2	Requisitos do Sistema	2
3	Instalação dos Componentes	3
3.1	Instalação do IntelliJ IDEA	3
3.2	Instalação do MySQL Server e MySQL Workbench	6
3.3	Instalação do Git	9
4	Configurando e Executando o Projeto	11
4.1	Clonando o Repositório	12
4.2	Configuração do Banco de Dados	13
4.3	Execução do Backend (Spring Boot)	14
4.4	Acessando o Frontend	15
5	Utilizando o Sistema	16
5.1	Visão Geral da Interface e Estrutura do Frontend	16
5.2	Operações CRUD via Interface do Frontend (Exemplo: Gerenciamento de Alunos)	18
5.2.1	Listar Alunos	18
5.2.2	Adicionar Novo Aluno	19
5.2.3	Visualizar Detalhes do Aluno	19
5.2.4	Editar Aluno	20
5.2.5	Excluir Aluno	20
5.3	Testando Operações CRUD no Backend (via Classe de Teste)	21
5.4	Testando as Rotas com o Postman	22
6	Diagramas do Sistema	25
6.1	Diagrama Entidade-Relacionamento (DER)	25
6.2	Diagrama de Classes UML	26
7	Considerações Finais	27
8	Referências	27

1 Introdução

Este manual detalhado tem como objetivo principal guiar usuários e desenvolvedores através de todas as etapas necessárias para a instalação, configuração, execução e utilização eficaz do sistema de gerenciamento para plataformas de cursos online, denominado "JDBC-Course". O sistema foi concebido e desenvolvido como um projeto acadêmico, utilizando uma arquitetura robusta composta por **Java** com o framework **Spring Boot** para a construção do backend, e tecnologias web fundamentais como **HTML**, **CSS** e **JavaScript** puro para a interface do frontend. A persistência e o gerenciamento dos dados são realizados através do Sistema de Gerenciamento de Banco de Dados Relacional (SGBDR) **MySQL**.

O "JDBCCourse" é uma plataforma que visa simular um ambiente de aprendizado online, permitindo o cadastro e a manipulação de diversas entidades essenciais, incluindo Alunos, os Cursos propriamente ditos, Módulos que compõem os cursos, Aulas individuais, Instrutores responsáveis pelo conteúdo, Matrículas de alunos nos cursos, Avaliações fornecidas pelos alunos e Certificados de conclusão. A camada de acesso a dados no backend foi implementada utilizando Data Access Objects (DAOs) com **JDBC (Java Database Connectivity)**, o que permite uma interação direta e controlada com o banco de dados MySQL. Este manual cobrirá desde os pré-requisitos de software até a utilização das funcionalidades CRUD (Create, Read, Update, Delete) através da interface gráfica e testes de API.

2 Requisitos do Sistema

Para garantir uma instalação e execução bem-sucedida do projeto JDBCCourse, é crucial que o ambiente de desenvolvimento e execução atenda aos seguintes requisitos mínimos e recomendados:

- **Sistema Operacional:**
 - Windows 10 (64-bit) ou superior.
 - macOS 10.15 (Catalina) ou superior.
 - Distribuições Linux modernas com kernel 4.x ou superior (ex: Ubuntu 20.04 LTS, Fedora 34+).
- **Java Development Kit (JDK):** Recomenda-se a versão 17 LTS ou a versão 21 LTS, pois o projeto foi configurado no `pom.xml` para utilizar Java 21. Certifique-se de que a variável de ambiente `JAVA_HOME` esteja configurada corretamente.
- **IntelliJ IDEA:**

- Community Edition (gratuita) é suficiente para este projeto.
- Ultimate Edition (paga) oferece funcionalidades adicionais que podem ser úteis, mas não são mandatórias.
- Versão 2023.x ou mais recente recomendada para melhor compatibilidade com as ferramentas e plugins.
- **MySQL Server:** Versão 8.0 ou superior é recomendada devido a potenciais funcionalidades e otimizações. Versões 5.7 também podem funcionar, mas a 8.0 é preferível.
- **MySQL Workbench** (ou ferramenta similar):
 - MySQL Workbench 8.0 ou superior.
 - Alternativas como DBeaver, SQL Developer, ou HeidiSQL também podem ser utilizadas para gerenciamento do banco de dados.
- **Git:** Versão 2.x ou mais recente para clonagem do repositório e controle de versão.
- **Navegador Web Moderno:**
 - Google Chrome (versão mais recente).
 - Mozilla Firefox (versão mais recente).
 - Microsoft Edge (baseado em Chromium, versão mais recente).

3 Instalação dos Componentes

Esta seção detalha o processo de instalação de cada componente de software necessário para o desenvolvimento e execução do sistema JDBCourse.

3.1 Instalação do IntelliJ IDEA

O IntelliJ IDEA é um IDE robusto e popular para desenvolvimento Java, oferecendo excelente suporte para projetos Spring Boot e Maven.

1. Acesso e Download:

- Abra seu navegador e acesse o site oficial da JetBrains para o IntelliJ IDEA: <https://www.jetbrains.com/idea/download/>.
- Na página de download, você encontrará opções para diferentes sistemas operacionais (Windows, macOS, Linux) e duas edições principais: Ultimate (paga, com trial disponível) e Community (gratuita e open-source).

- Para este projeto, a **Community Edition** é suficiente. Clique no botão de download correspondente à Community Edition para o seu sistema operacional.

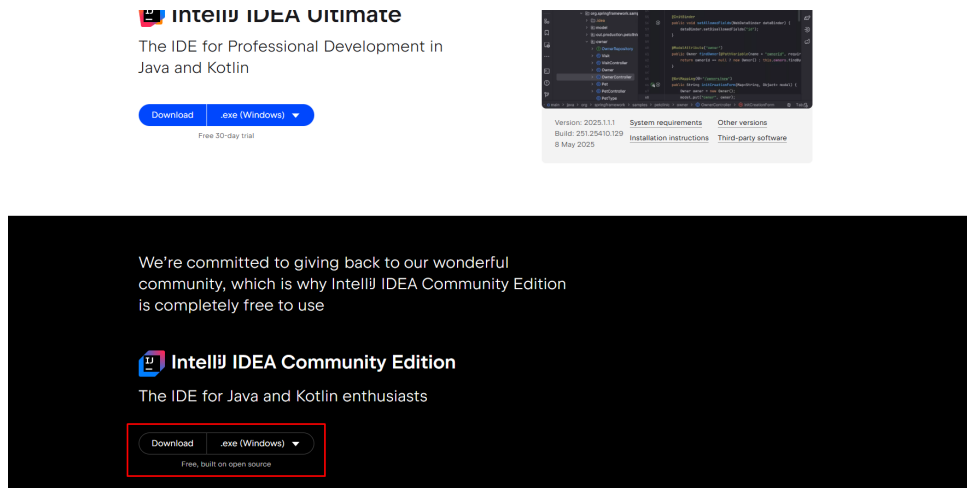


Figura 1: Seleção da IntelliJ IDEA Community Edition para download. Fonte: JetBrains (2025).

2. Execução do Instalador:

- Após o download do arquivo de instalação (ex: `ideaIC-xxxx.y.z.exe` para Windows, `.dmg` para macOS, `.tar.gz` para Linux), execute-o.
- **Windows:** Dê um duplo clique no arquivo `.exe`. Permita que o instalador faça alterações no seu dispositivo, se solicitado pelo Controle de Conta de Usuário.
- **macOS:** Abra o arquivo `.dmg` e arraste o ícone do IntelliJ IDEA para a pasta Aplicativos.
- **Linux:** Extraia o arquivo `.tar.gz` para um diretório de sua escolha (ex: `/opt/` ou `/jetbrains/`). Para iniciar, navegue até a pasta `bin` dentro do diretório extraído e execute `idea.sh`.

3. Passos da Instalação (Exemplo Windows):

- **Welcome Screen:** Clique em "Next".
- **Destination Folder:** Escolha o local de instalação ou mantenha o padrão. Clique em "Next".
- **Installation Options:** Você pode optar por criar um atalho na área de trabalho (32-bit ou 64-bit launcher), adicionar "Open Folder as Project" ao menu de contexto, e associar extensões de arquivo (como `.java`, `.groovy`, `.kt`). Marque as opções conforme sua preferência. Adicionar ao PATH não é usualmente necessário para o IntelliJ em si, mas criar um atalho é conveniente. Clique em "Next".

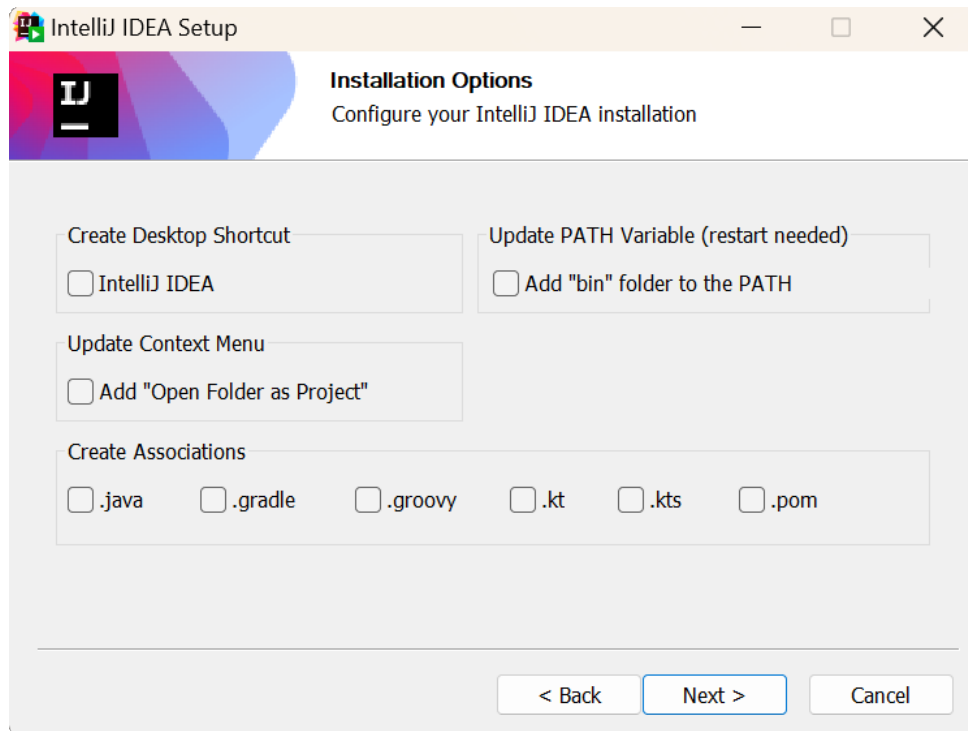


Figura 2: Opções de instalação do IntelliJ IDEA. Fonte: Elaborado pelo autor (2025).

- **Start Menu Folder:** Escolha a pasta do Menu Iniciar ou mantenha o padrão. Clique em "Install".
- **Installation Progress:** Aguarde a conclusão da instalação.
- **Completing Setup:** Após a conclusão, você pode marcar a opção "Run IntelliJ IDEA Community Edition" e clicar em "Finish".

4. Primeira Execução e Configuração Inicial (se aplicável):

- Ao iniciar pela primeira vez, você pode ser solicitado a importar configurações (se tiver uma versão anterior) ou aceitar os termos de privacidade e o acordo de usuário.
- Você também pode ser apresentado a opções de customização da interface (tema claro ou escuro, plugins). O tema escuro (Darcula) é popular. Plugins como Maven, Git são geralmente incluídos e ativados por padrão na Community Edition.

5. Possíveis Problemas e Soluções:

- *Download não inicia/corrompido:* Verifique sua conexão com a internet e tente baixar novamente. Use um gerenciador de downloads se tiver conexões instáveis.

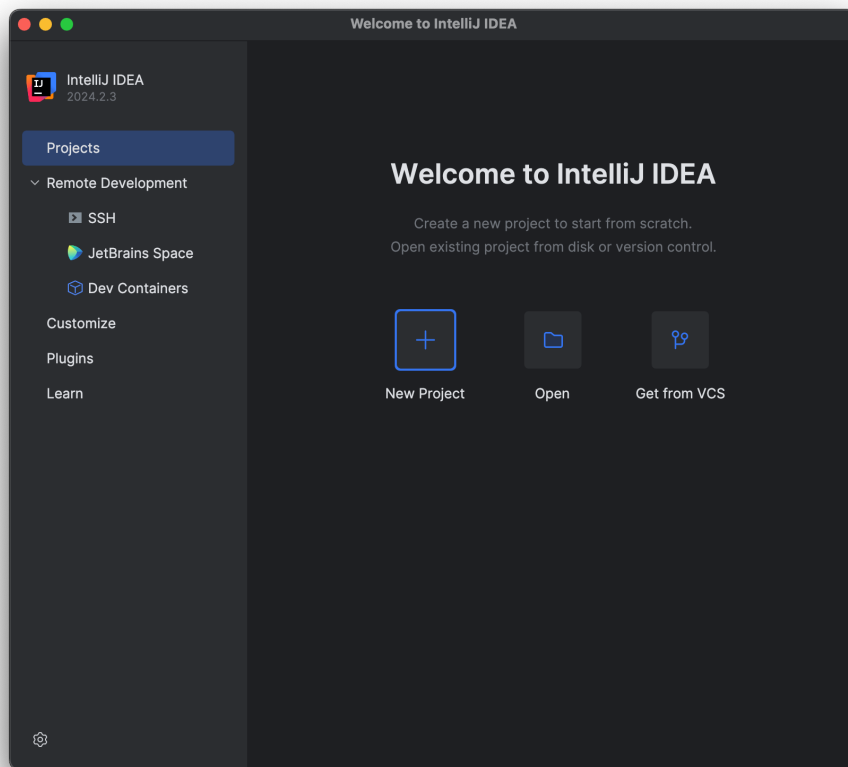


Figura 3: Tela inicial do IntelliJ IDEA após a instalação. Fonte: JetBrains (2025).

- *Erro durante a instalação (permissões):* No Windows, tente executar o instalador como administrador. No Linux, certifique-se de ter permissões de escrita no diretório de instalação.
- *JDK não encontrado pelo IntelliJ:* Embora o IntelliJ possa usar um JRE embutido para rodar a si mesmo, para desenvolvimento de projetos Java, ele precisará de um JDK configurado. Você pode configurar o JDK nas configurações do projeto ("File» "Project Structure» "SDKs") após abrir ou criar um projeto.

3.2 Instalação do MySQL Server e MySQL Workbench

O MySQL é o SGBDR escolhido para este projeto. O MySQL Workbench é uma ferramenta gráfica para interagir com o servidor.

1. Download do MySQL Installer (Windows) ou Pacotes Individuais:

- Acesse <https://dev.mysql.com/downloads/>.
- Para Windows, a maneira mais fácil é baixar o "MySQL Installer for Windows" da seção "MySQL Community Downloads". Ele permite instalar o Servidor, Workbench, Conectores, etc.

- Para macOS ou Linux, você pode baixar os pacotes individuais do "MySQL Community Server" e "MySQL Workbench" ou usar gerenciadores de pacotes (como Homebrew para macOS, apt/yum para Linux).

MySQL Community Downloads

MySQL Installer

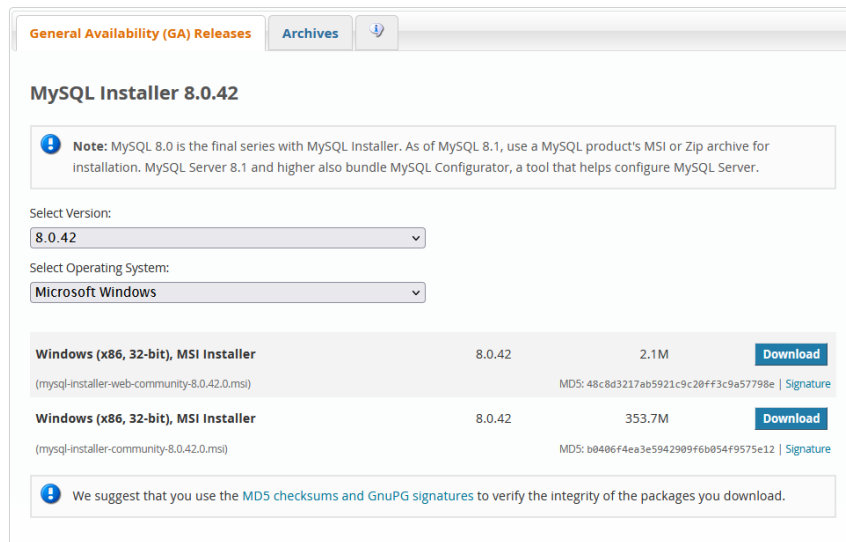


Figura 4: Página oficial de downloads do MySQL. Fonte: Oracle (2025).

2. Instalação do MySQL Server (usando MySQL Installer como exemplo):

- Execute o MySQL Installer.
- **Choosing a Setup Type:** Selecione "Developer Default" (inclui Servidor, Workbench, etc.) ou "Custom" para escolher componentes manualmente. "Server only" também é uma opção se você for instalar o Workbench separadamente. Para este projeto, "MySQL Server" e "MySQL Workbench" são os essenciais.
- **Check Requirements:** O instalador pode verificar por pré-requisitos (como Microsoft Visual C++ Redistributable). Instale-os se necessário.
- **Installation:** Nessa tela você verá todos os produtos MySQL. Apenas clique execute que automaticamente instalará tudo necessário.
- **Configuration Steps:**
 - **Type and Networking:** Mantenha o tipo "Development Computer" e a porta padrão (3306), a menos que você tenha um motivo específico para alterar.
 - **Authentication Method:** Recomenda-se usar "Use Strong Password Encryption for Authentication".

- **Accounts and Roles:** Defina uma senha para o usuário root do MySQL. Esta senha é crucial e será usada na configuração do `application.properties` do seu projeto **Spring Boot**. Você pode adicionar outros usuários se desejar, mas o root é suficiente para desenvolvimento local.

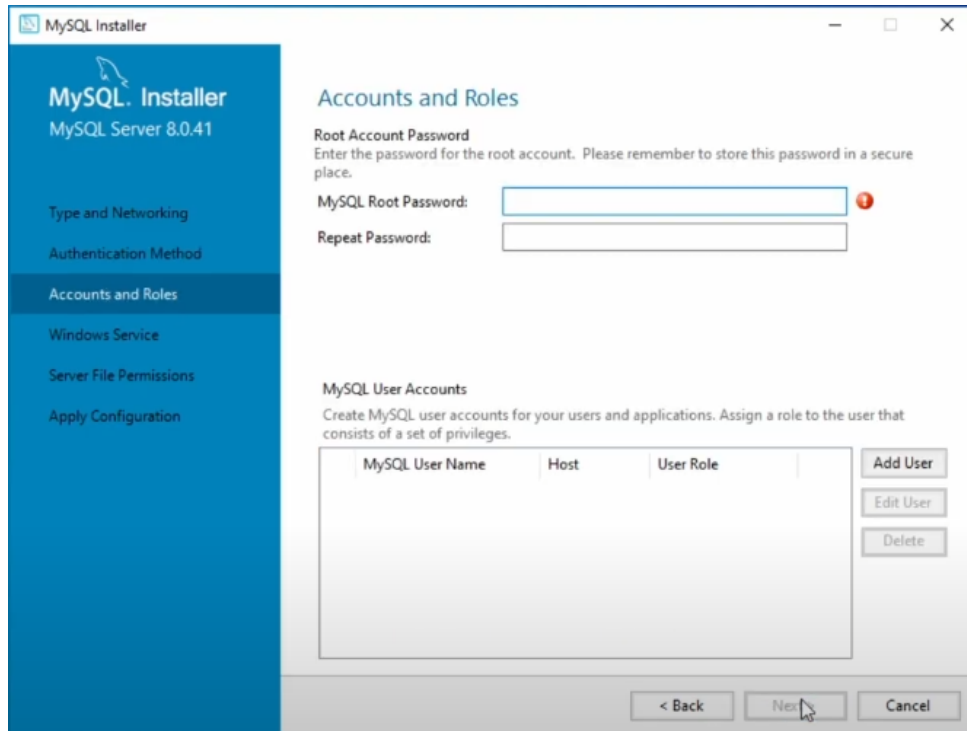


Figura 5: Definindo a senha do usuário root do MySQL. Fonte: Elaborado pelo autor (2025).

- **Windows Service (Windows):** Configure o MySQL Server para rodar como um serviço do Windows, nomeando o serviço (ex: MySQL80) e configurando-o para iniciar com o sistema.
 - **Server File Permission:** Permita completo de administrador
 - **Apply Configuration:** Apenas clique em execute que o instalador aplicará todas as configurações.
- **Installation Complete.**
3. **Instalação do MySQL Workbench (se não incluído ou instalado separadamente):**
 - Baixe do site do MySQL e siga o instalador padrão. É uma instalação direta.
 4. **Verificando a Instalação e Serviço:**
 - No Windows, você pode verificar se o serviço do MySQL está rodando em "Serviços"(procure por "mysql"no menu Iniciar).

- Abra o MySQL Workbench e tente criar uma nova conexão com o servidor local (localhost, porta 3306, usuário root, e a senha que você definiu). Se a conexão for bem-sucedida, o servidor está funcionando.

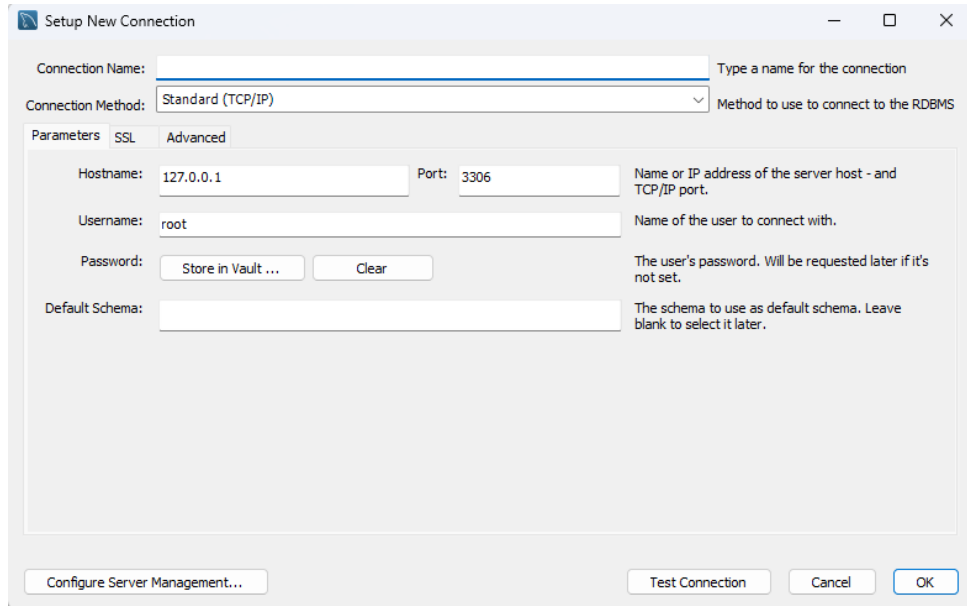


Figura 6: Conexão ao servidor MySQL local via MySQL Workbench. Fonte: Elaborado pelo autor (2025).

5. Possíveis Problemas e Soluções:

- *Porta 3306 em uso:* Se outro aplicativo estiver usando a porta 3306, o MySQL Server pode não iniciar. Você precisará parar o outro serviço ou configurar o MySQL para usar uma porta diferente (e ajustar a URL de conexão no seu `application.properties`).
- *Firewall bloqueando conexões:* Certifique-se de que seu firewall permite conexões na porta do MySQL (3306 por padrão).
- *Senha root esquecida:* Existem procedimentos para resetar a senha root do MySQL, mas são mais complexos. É melhor anotá-la durante a instalação.
- *Falha na instalação de componentes:* Às vezes, o MySQL Installer pode falhar ao instalar um componente específico. Tente reinstalar apenas esse componente ou verifique os logs do instalador para mais detalhes.

3.3 Instalação do Git

Git é o sistema de controle de versão distribuído usado para gerenciar o código-fonte do projeto.

1. Download do Git:

- Acesse o site oficial: <https://git-scm.com/downloads>.
- O site geralmente detecta seu sistema operacional e sugere o download apropriado.

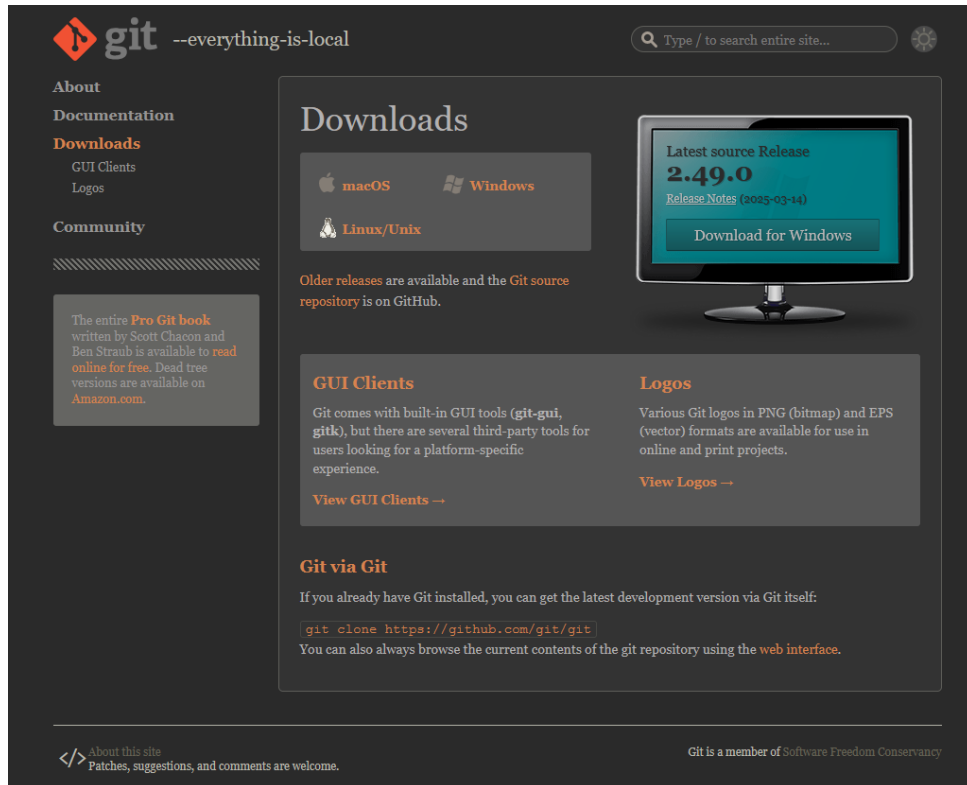


Figura 7: Página oficial para download do Git. Fonte: git-scm.com (2025).

2. Executando o Instalador (Exemplo Windows):

- Após o download, execute o instalador.
- **Information (GNU General Public License):** Leia (opcional) e clique em "Next".
- **Select Components:** As opções padrão geralmente são boas. Você pode optar por adicionar um ícone na área de trabalho. "Git Bash Here" e "Git GUI Here" (integração com o menu de contexto do Windows Explorer) são úteis.
- **Choosing the default editor used by Git:** Vim é o padrão, mas você pode escolher outro como Notepad++, VS Code, etc., se estiverem instalados e o instalador os detectar. Para iniciantes, manter Vim ou escolher um editor mais simples pode ser melhor.
- **Adjusting the name of the initial branch in new repositories:** O padrão "master" está mudando para "main". Você pode escolher qual o instalador deve configurar como padrão para novos repositórios que você criar.

- **Adjusting your PATH environment:** Esta é uma etapa importante. A opção recomendada é geralmente "Git from the command line and also from 3rd-party software". Isso adiciona o Git ao seu PATH do sistema, permitindo que você use comandos `git` de qualquer terminal (CMD, PowerShell).

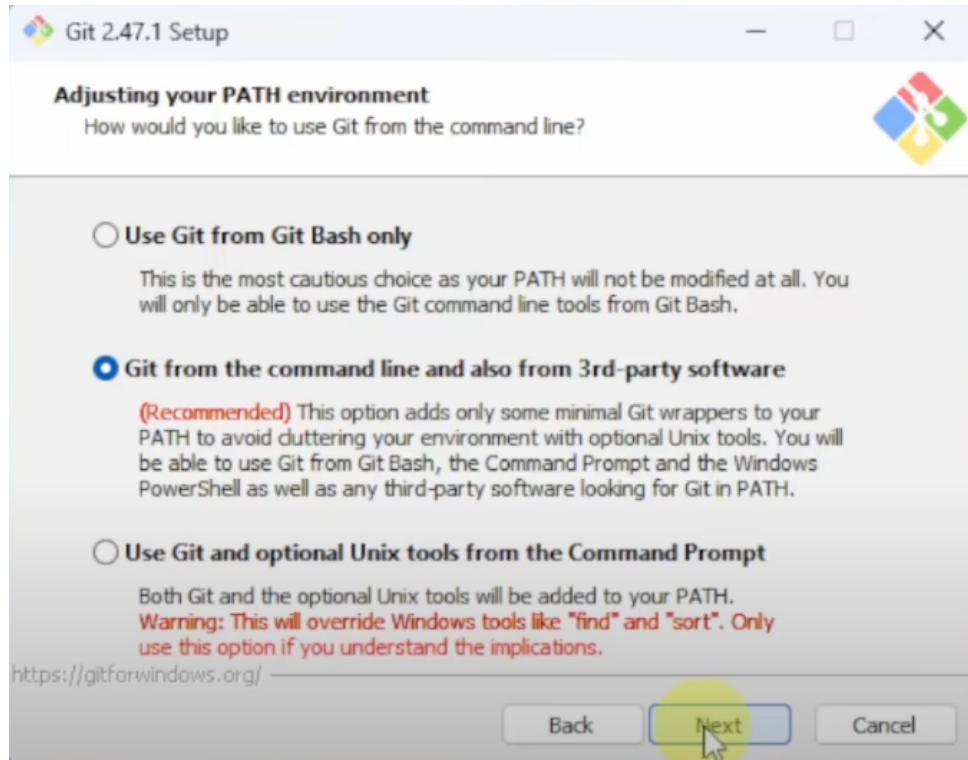


Figura 8: Opção de configuração do PATH durante a instalação do Git. Fonte: Elaborado pelo autor (2025).

- **Choosing HTTPS transport backend, Configuring line ending conversions, Configuring the terminal emulator, etc.:** Para as demais telas, as opções padrão são geralmente seguras e adequadas para a maioria dos usuários. Clique em "Next" em cada uma delas.
- **Configuring extra options, Configuring experimental options:** Mantenha desmarcado, a menos que saiba o que está fazendo.
- **Install:** Clique em "Install".
- **Completing Setup:** Após a conclusão, você pode desmarcar "View Release Notes" e clicar em "Finish".

4 Configurando e Executando o Projeto

Após a instalação dos componentes, siga estas etapas para configurar e executar o projeto JDBCcourse.

4.1 Clonando o Repositório

1. **Obtenha a URL do Repositório:** Localize a URL HTTPS ou SSH do seu projeto JDBCcourse no GitHub.
2. **Clonando via IntelliJ IDEA (Método Recomendado):**
 - Abra o IntelliJ IDEA. Na tela de boas-vindas, selecione "Get from VCS" (Obter do Controle de Versão). Se um projeto já estiver aberto, vá em "File» "New» "Project from Version Control...".
 - Na caixa de diálogo, certifique-se de que "Git" ou "GitHub" esteja selecionado na aba da esquerda (ou que a URL seja para um repositório Git).
 - Cole a URL do repositório no campo "URL".
 - Escolha o "Directory" (diretório local) onde o projeto será clonado em sua máquina.
 - Clique em "Clone". O IntelliJ IDEA fará o download do projeto.

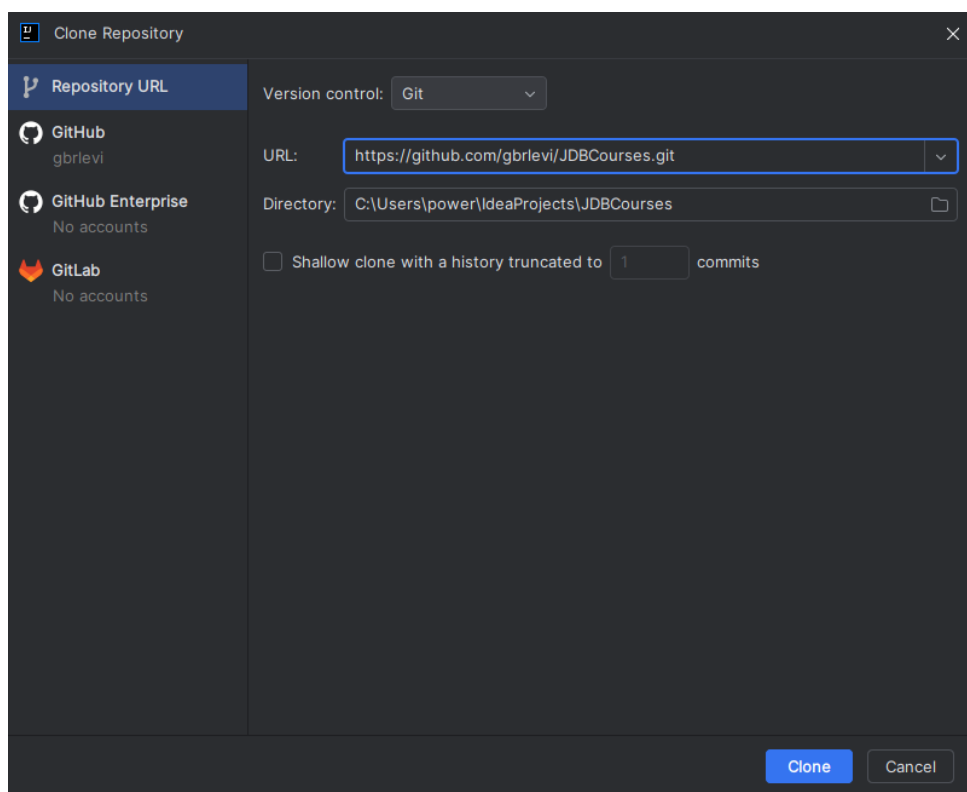


Figura 9: Clonando o repositório JDBCcourse via IntelliJ IDEA. Fonte: Elaborado pelo autor (2025).

3. Abrindo o Projeto e Build Inicial:

- O IntelliJ IDEA detectará o arquivo `pom.xml` e reconhecerá o projeto como um projeto Maven. Ele automaticamente tentará "sincronizar" o projeto, o

que envolve baixar todas as dependências listadas no `pom.xml` (como Spring Boot, MySQL Connector, etc.) do repositório central do Maven.

- Este processo de "Resolving Dependencies" ou "Importing Maven project" pode levar alguns minutos, especialmente na primeira vez, pois depende da velocidade da sua internet e do número de dependências. Você verá uma barra de progresso no canto inferior direito do IntelliJ.
- Se o IntelliJ perguntar se você confia no projeto Maven, clique em "Trust Project".

[INSERIR SCREENSHOT: IntelliJ IDEA com o projeto JDBCcourse aberto]

Figura 10: Projeto JDBCcourse aberto no IntelliJ IDEA com o Maven resolvendo dependências. Fonte: Elaborado pelo autor (2025).

4.2 Configuração do Banco de Dados

1. Criação do Banco de Dados no MySQL:

- Abra o MySQL Workbench e conecte-se ao seu servidor MySQL local.
- Se ainda não o fez, crie o banco de dados que será utilizado pela aplicação. O nome configurado no projeto é `cursoonline`. Execute o seguinte script SQL:

```
1 CREATE DATABASE IF NOT EXISTS cursoonline
2 CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
3
```

Listing 1: Criação do Banco de Dados

`CHARACTER SET` e `COLLATE` são recomendados para suportar uma ampla gama de caracteres.

2. Configuração das Credenciais no `application.properties`:

- No IntelliJ IDEA, navegue até o arquivo `src/main/resources/application.properties`.
- Certifique-se de que as propriedades de `datasource` estejam corretas para o seu ambiente MySQL local:

```
1 # Database Configuration
2 spring.datasource.url=jdbc:mysql://localhost:3306/cursoOnline
3 spring.datasource.username=SEU_USUARIO_MYSQL # Ex: root
4 spring.datasource.password=SUA_SENHA_MYSQL
5 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
6
```

Listing 2: Configuração do Datasource

4.3 Execução do Backend (Spring Boot)

Com o projeto clonado, dependências resolvidas e banco de dados configurado, você pode executar a aplicação backend.

1. **Localize a Classe Principal da Aplicação:** No IntelliJ IDEA, encontre a classe `CursosOnlineApplication.java`. Ela está localizada em:

```
src/main/java/com/example/cursosonline/.
```

Esta classe contém o método `main` que inicializa o Spring Boot.

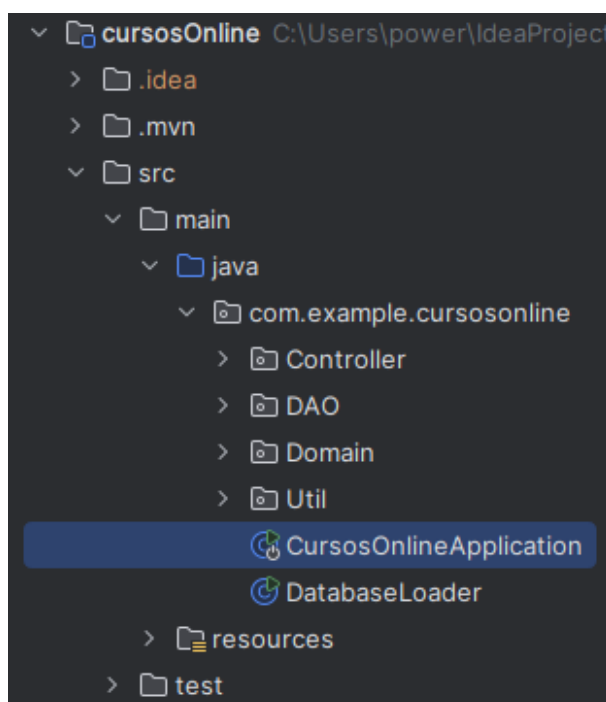


Figura 11: Classe principal `CursosOnlineApplication.java` no explorador de projeto. Fonte: Elaborado pelo autor (2025).

2. Execute a Aplicação:

- Clique com o botão direito do mouse sobre o nome do arquivo `CursosOnlineApplication.java` no explorador de projeto ou diretamente no editor de código.
- No menu de contexto, selecione "Run 'CursosOnlineApplication.main()'".
- Alternativamente, se o IntelliJ já criou uma configuração de execução, você pode selecioná-la na barra de ferramentas superior e clicar no botão verde "Play" (Run).

3. Acompanhe o Console de Saída:

- A aba "Run" ou "Debug" no painel inferior do IntelliJ IDEA exibirá os logs de inicialização do Spring Boot.
- Procure por mensagens indicando que o contexto do Spring foi carregado, que a conexão com o banco de dados foi estabelecida, e finalmente, uma mensagem similar a: `Tomcat started on port(s): 8080 (http) with context path ''Started CursosOnlineApplication in X.XXX seconds (JVM running for Y.YYY)`
- Isso indica que o servidor backend está rodando e escutando na porta 8080.
- **Importante:** Mantenha esta aba do console e o processo do Spring Boot em execução enquanto estiver utilizando o sistema. Fechar esta janela ou parar o processo interromperá o backend.

```
INFO 29472 --- [cursosOnline] [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available (set 'hibernate.transaction.jta.platform'
INFO 29472 --- [cursosOnline] [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
WARN 29472 --- [cursosOnline] [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may
INFO 29472 --- [cursosOnline] [main] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page: class path resource [static/index.html]
INFO 29472 --- [cursosOnline] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'
INFO 29472 --- [cursosOnline] [main] c.e.c.CursosOnlineApplication : Started CursosOnlineApplication in 3.177 seconds (process running for 3.661)
```

Figura 12: Logs de console confirmando a inicialização do backend Spring Boot. Fonte: Elaborado pelo autor (2025).

4. Execução do DatabaseLoader.java (Opcional/Primeira Vez):

- O arquivo `DatabaseLoader.java` contém código para popular o banco de dados com dados de exemplo. Para que esses dados sejam inseridos, você precisa executar o método `main` deste arquivo **após** o backend Spring Boot ter sido iniciado uma vez (para que o `ddl-auto=update` crie as tabelas, se configurado).
- Localize `DatabaseLoader.java`, clique com o botão direito e selecione "Run 'DatabaseLoader.main()'".
- Verifique o console para mensagens de sucesso da inserção dos dados. Execute este loader apenas uma vez, a menos que queira resetar os dados de exemplo.

4.4 Acessando o Frontend

Com o backend Spring Boot em execução:

1. Abra seu navegador web preferido.
2. Na barra de endereços, digite: <http://localhost:8080/>
3. A página `index.html` do seu dashboard, localizada em `src/main/resources/static/`, deve ser carregada.
4. A partir daqui, você pode navegar pelas diferentes seções da aplicação clicando nos links da sidebar.

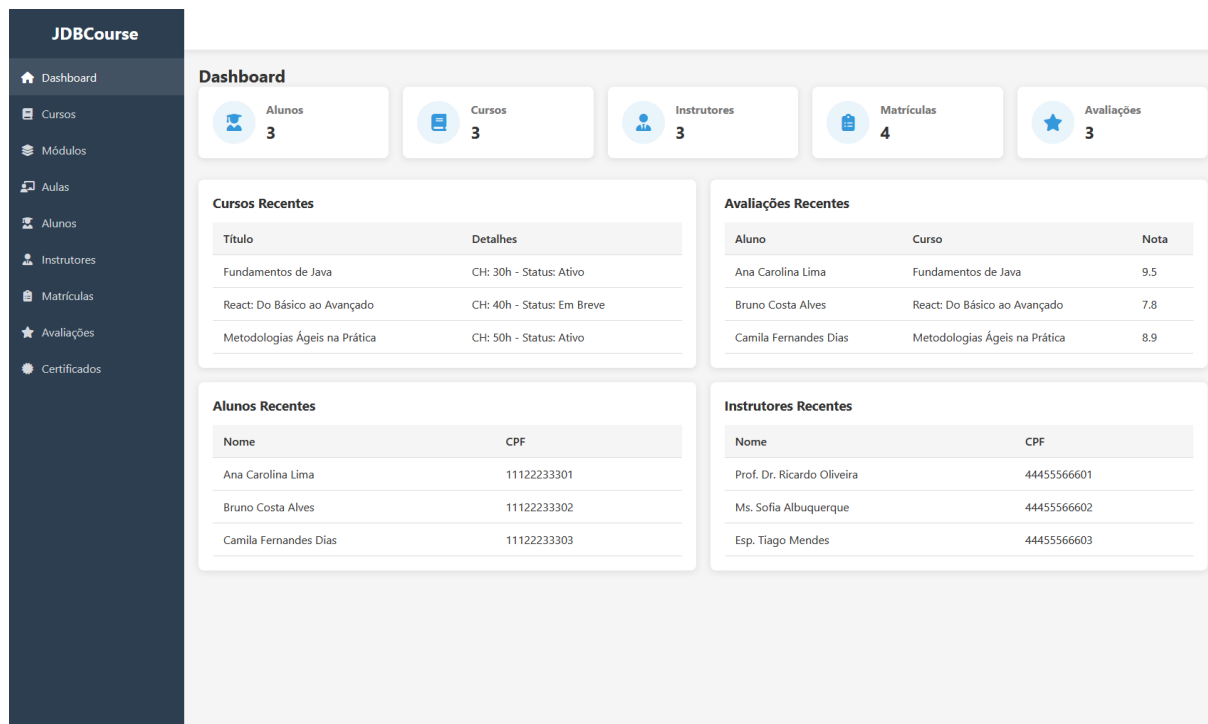


Figura 13: Página inicial do sistema JDBCOURSE acessada via navegador. Fonte: Elaborado pelo autor (2025).

5 Utilizando o Sistema

5.1 Visão Geral da Interface e Estrutura do Frontend

O frontend do JDBCOURSE é construído com HTML, CSS e JavaScript puro, servido estaticamente pelo Spring Boot a partir da pasta `src/main/resources/static/`.

Estrutura dos Arquivos Frontend:

- `static/index.html`: A página principal do dashboard.
- `static/pages/`: Contém as páginas HTML para cada entidade (ex: `alunos.html`, `cursos.html`).
- `static/css/styles.css`: Arquivo CSS principal para estilização de todas as páginas.
- `static/js/`: Contém os arquivos JavaScript:
 - `api.js`: Responsável por todas as chamadas AJAX (via `fetch`) para os endpoints da API REST do backend. Ele centraliza a comunicação com o servidor.
 - `utils.js`: Contém funções utilitárias reutilizáveis, como formatação de datas, exibição de notificações, criação de modais e formulários dinâmicos.

- `dashboard.js`: Lógica específica para popular os cards de estatísticas e as listas de itens recentes na página `index.html`.
- Arquivos específicos por entidade (ex: `alunos.js`, `cursos.js`): Cada um contém a lógica para carregar dados, popular tabelas, manipular formulários e interagir com o usuário para a respectiva página HTML (ex: `alunos.html`). Eles utilizam as funções de `api.js` para buscar/enviar dados e de `utils.js` para operações comuns de UI.

Interação das Partes do Frontend:

1. O usuário navega para uma página HTML (ex: `alunos.html`).
2. O script JavaScript associado (ex: `alunos.js`) é executado quando o DOM está carregado.
3. O script JS (ex: `alunos.js`) chama funções em `api.js` (ex: `window.getAllAlunos()`) para buscar dados do backend.
4. `api.js` usa a API `fetch` para fazer uma requisição HTTP ao endpoint do Spring Boot (ex: `GET /api/alunos`).
5. O backend processa a requisição e retorna dados (geralmente JSON).
6. `api.js` recebe a resposta e a retorna para o script chamador (ex: `alunos.js`).
7. O script JS da página (ex: `alunos.js`) então manipula o DOM para exibir os dados em tabelas, popular formulários, etc.
8. Interações do usuário (cliques em botões, submissão de formulários) disparam funções no JS da página, que podem chamar `utils.js` (para modais, notificações) e `api.js` (para enviar dados ao backend).

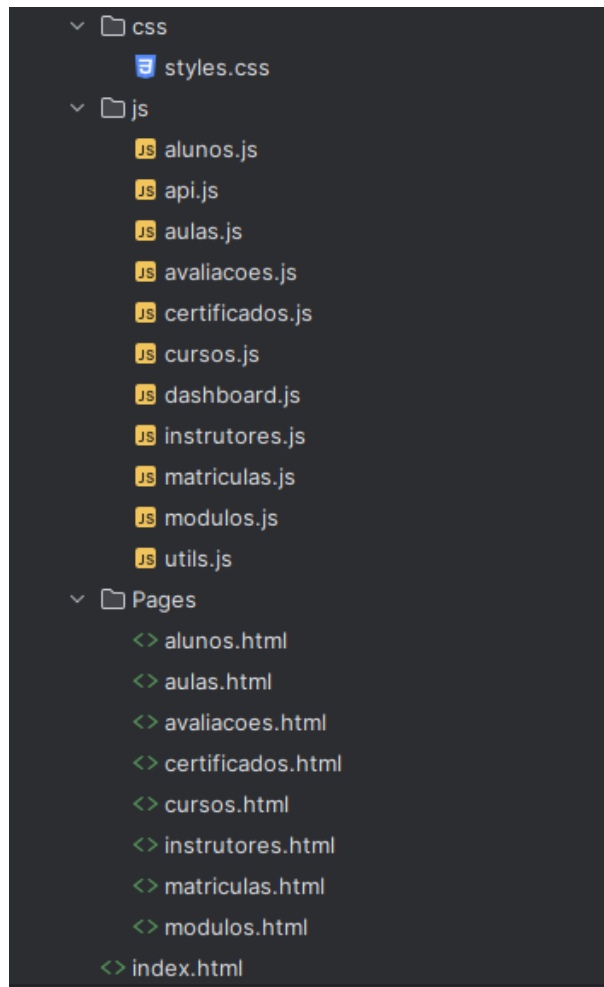


Figura 14: Componentes principais da interface do sistema JDBCourse. Fonte: Elaborado pelo autor (2025).

5.2 Operações CRUD via Interface do Frontend (Exemplo: Gerenciamento de Alunos)

(Esta seção permanece similar à anterior, mas você pode adicionar mais detalhes sobre os campos específicos do formulário de Aluno, por exemplo)

5.2.1 Listar Alunos

Ao clicar em "Alunos" na sidebar, a página `alunos.html` é carregada. O script `alunos.js` busca e exibe uma tabela com ID, Nome, CPF e Data de Cadastro. A barra de pesquisa no topo da página permite filtrar a lista por nome ou CPF.

Pesquisar Alunos










ID	Nome	CPF	Data de Cadastro	Ações
1	Ana Carolina Lima	11122233301	01/06/2025	  
2	Bruno Costa Alves	11122233302	01/06/2025	  
3	Camila Fernandes Dias	11122233303	01/06/2025	  

Figura 15: Página de listagem de Alunos no sistema JDBCourse. Fonte: Elaborado pelo autor (2025).

5.2.2 Adicionar Novo Aluno

O botão "+ Novo Aluno" abre um modal (gerenciado por `utils.js` e `alunos.js`) com um formulário para Nome, CPF e Senha. Após submissão, `alunos.js` chama `api.js` para enviar os dados ao endpoint `POST /api/alunos`.

ma

ves

les Dia

stro

Novo Aluno

×

Nome Completo *

João da Silva

CPF (somente números) *

12312312377

Senha *

●●●●●●●●●●

Criar Aluno

Figura 16: Modal para adição de um novo Aluno. Fonte: Elaborado pelo autor (2025).

5.2.3 Visualizar Detalhes do Aluno

(Conforme descrito anteriormente, mostrando informações e tabelas de Matrículas, Avaliações, Certificados, populadas por chamadas específicas em `alunos.js` via `api.js`).

Detalhes do Aluno
✕ Fechar

ID:

1

Nome:

Ana Carolina Lima

CPF:

11122233301

Data de Cadastro:

01/06/2025

Matrículas

ID	Curso (Título)	Data de Matrícula	Ativo	Ações
1	Fundamentos de Java	01/06/2025	Sim	
4	React: Do Básico ao Avançado	01/06/2025	Sim	

Avaliações

ID	Curso (Título)	Nota	Feedback	Ações
1	Fundamentos de Java	9.5	Curso excelente, didática impecável e conteúdo muito relevante!	

Certificados

ID	Curso (Título)	Data de Conclusão	Ações
1	Fundamentos de Java	01/06/2025	

Figura 17: Visualização detalhada de um Aluno e suas informações associadas. Fonte: Elaborado pelo autor (2025).

5.2.4 Editar Aluno

(Conforme descrito anteriormente, usando um modal pré-preenchido e chamando PUT /api/alunos/id).

5.2.5 Excluir Aluno

(Conforme descrito anteriormente, com modal de confirmação e chamando DELETE /api/alunos/id).

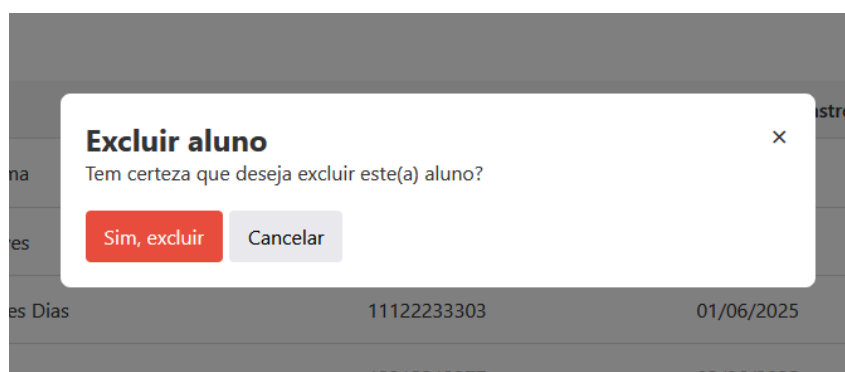


Figura 18: Visualização do modal de delete. Fonte: Elaborado pelo autor (2025).

5.3 Testando Operações CRUD no Backend (via Classe de Teste)

Para validar a camada de acesso a dados (DAOs), utiliza-se a classe `CrudTest.java`. Esta classe permite executar as operações CRUD (Criar, Ler, Atualizar, Deletar) diretamente no backend, focando na interação com os DAOs de uma entidade específica.

Esse teste tem como objetivo:

- Verificar o funcionamento individual dos métodos CRUD de um DAO (ex: `AlunoDAO`).
- Confirmar a correta interação com o banco de dados (consultas SQL, mapeamento de resultados).
- Auxiliar na depuração da camada de persistência de forma isolada.

Como Utilizar:

1. **Localize e Abra:** No IntelliJ IDEA, encontre a classe de teste

`src/main/java/CrudTest.java`.

2. **Analise e Execute:** O método `main` da classe conterá exemplos de chamadas aos métodos do DAO. Você pode executar a classe diretamente ("Run '`CrudTest.main()`'").

```
1 public class CrudTest {
2     public static void main(String[] args) {
3         AlunoDAO alunoDAO = new AlunoDAO();
4
5         // Exemplo: Criar Aluno
6         Aluno aluno = new Aluno(); /* ... configurar aluno ... */
7         alunoDAO.save(aluno);
8         System.out.println("Aluno criado com ID: " + aluno.getId());
9
10        ;
11
12        // Outros testes para findById, findAll, update, delete...
13    }
14 }
```

Listing 3: Estrutura básica de `CrudTest.java`

3. **Pré-requisitos:** O servidor MySQL deve estar ativo, o banco `cursoonline` e as tabelas devem existir, e as credenciais em `application.properties` devem estar corretas.
4. **Observe o Console:** A saída no console do IntelliJ IDEA mostrará os resultados das operações CRUD executadas.

Figura 19: Exemplo de saída no console após executar TestAluno.java. Fonte: Elaborado pelo autor (2025).

Estas classes de teste são ferramentas de desenvolvimento para verificar a lógica do backend, não sendo parte da interface do usuário final.

5.4 Testando as Rotas com o Postman

Ferramentas de cliente HTTP como Postman (ou Insomnia) são valiosas para testar diretamente os endpoints da API RESTful do backend, independentemente do frontend. Isso permite verificar o comportamento de cada rota, os dados enviados e recebidos, e os códigos de status HTTP.

Configuração Básica no Insomnia/Postman:

- **Base URL:** Você pode configurar uma variável de ambiente na ferramenta com a URL base da sua API, por exemplo, `BASE_URL = http://localhost:8080/api`.
- **Headers:** Para requisições POST e PUT que enviam dados JSON, certifique-se de adicionar o header `Content-Type: application/json`.

Exemplos de Requisições:

1. **Listar todos os Alunos (GET):**

- Método: GET
- URL: {BASE_URL}/alunos (ou <http://localhost:8080/api/alunos>)
- Resultado esperado: Código 200 (OK) com um array JSON de cursos, ou 204 (No Content) se não houver cursos.

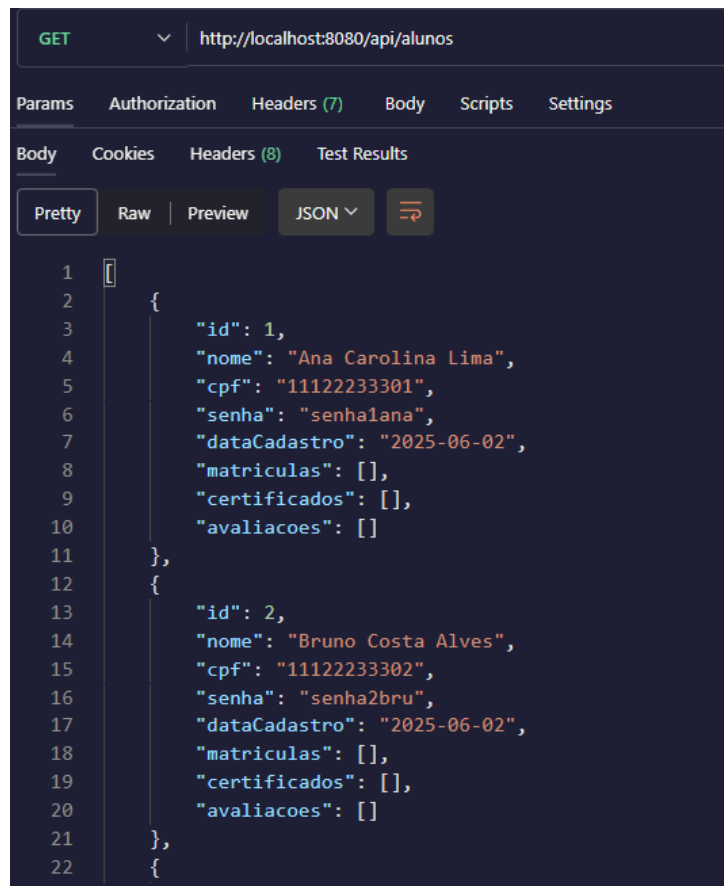


Figura 20: Testando o endpoint GET /api/cursos. Fonte: Elaborado pelo autor (2025).

2. Criar um novo Módulo (POST):

- Método: POST
- URL: {BASE_URL}/modulos (ou <http://localhost:8080/api/modulos>)
- Corpo (Body) > JSON:

```

1  {
2    "conteudo": "Introducao ao JDBC",
3    "cargaHoraria": 10,
4    "qtdAulas": 5
5  }
6

```

Listing 4: Exemplo de corpo JSON para criar Módulo

- Resultado esperado: Código 201 (Created) com o objeto JSON do módulo criado (incluindo seu novo ID).

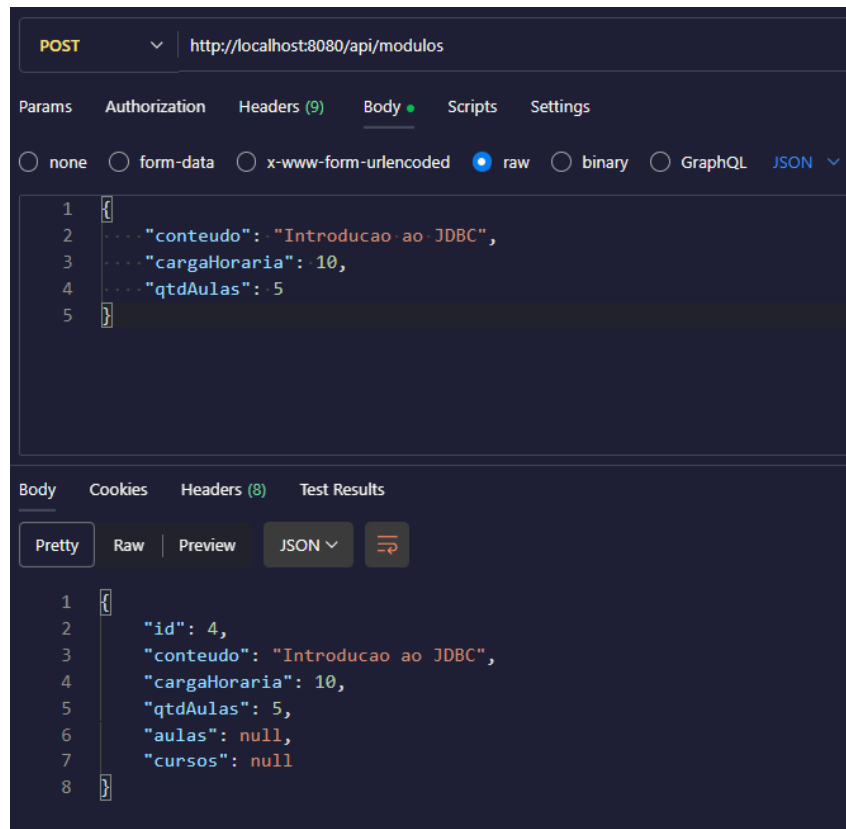


Figura 21: Testando o endpoint POST `/api/modulos`. Fonte: Elaborado pelo autor (2025).

3. Buscar um Instrutor por ID (GET):

- Método: GET
- URL: `{BASE_URL}/instrutores/1` (ou <http://localhost:8080/api/instrutores/1>, substituindo '1' pelo ID desejado)
- Resultado esperado: Código 200 (OK) com o objeto JSON do instrutor, ou 404 (Not Found) se o ID não existir.

4. Atualizar uma Aula (PUT):

- Método: PUT
- URL: `{BASE_URL}/aulas/1` (ou <http://localhost:8080/api/aulas/1>, substituindo '1' pelo ID da aula a ser atualizada)

- Corpo (Body) > JSON:

```
1 {  
2     "titulo": "Aula de Java Avan ado - Editada",  
3     "url": "http://link-da-aula.com/java-avancado-editado",  
4     "duracao": "01:00:00",  
5     "ordem": 1,  
6     "modulo": { "id": 1 }  
7 }  
8
```

Listing 5: Exemplo de corpo JSON para atualizar Aula

(Assumindo que a aula pertence ao módulo com ID 1)

- Resultado esperado: Código 200 (OK) com o objeto JSON da aula atualizada, ou 404 se o ID não existir.

5. Excluir uma Matrícula (DELETE):

- Método: DELETE
- URL: {BASE_URL}/matriculas/1 (ou <http://localhost:8080/api/matriculas/1>, substituindo '1' pelo ID da matrícula)
- Resultado esperado: Código 204 (No Content) se a exclusão for bem-sucedida, ou 404 se o ID não existir.

Explore os arquivos de Controller no seu projeto para descobrir todos os endpoints disponíveis e seus respectivos métodos HTTP e estruturas de payload esperadas.

6 Diagramas do Sistema

6.1 Diagrama Entidade-Relacionamento (DER)

O Diagrama Entidade-Relacionamento (DER) é uma representação visual da estrutura do banco de dados *cursoonline*. Ele ilustra as tabelas (que correspondem às entidades do sistema), as colunas (atributos) de cada tabela, as chaves primárias e estrangeiras, e os relacionamentos de cardinalidade (um-para-um, um-para-muitos, muitos-para-muitos) entre as tabelas. Este diagrama é fundamental para entender como os dados estão organizados e interconectados. *O DER deve incluir representações para as tabelas: Aluno, Aula, Avaliacao, Certificado, Curso, Instrutor, Matricula, Modulo, e tabelas de junção para relações Muitos-para-Muitos como curso_instrutor e curso_modulo.*

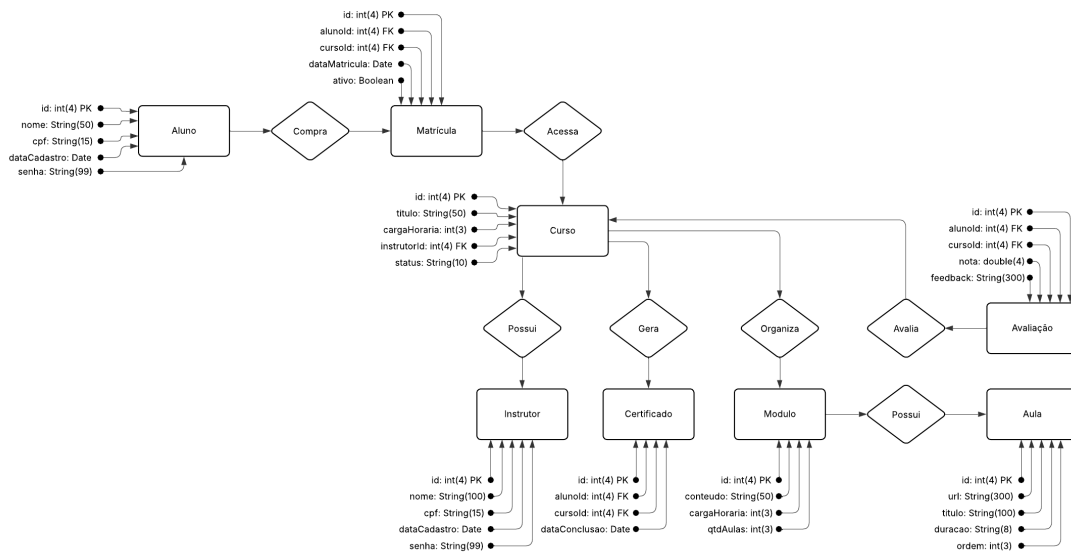


Figura 22: Diagrama Entidade-Relacionamento (DER) do sistema JDBCcourse. Fonte: Elaborado pelo autor (2025).

6.2 Diagrama de Classes UML

O Diagrama de Classes UML foca na estrutura das classes Java do domínio do seu projeto (localizadas no pacote `com.example.cursosonline.Domain`). Ele mostra os atributos de cada classe, seus métodos principais (getters/setters podem ser omitidos para simplificar) e os relacionamentos de associação (incluindo multiplicidade) entre elas, como herança ou composição, se aplicável.

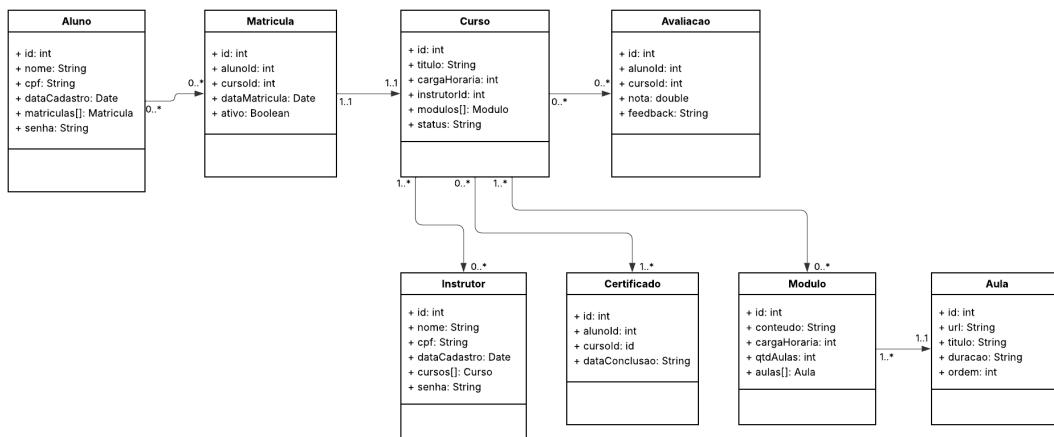


Figura 23: Diagrama de Classes UML simplificado das entidades do sistema JDBCcourse. Fonte: Elaborado pelo autor (2025).

7 Considerações Finais

O desenvolvimento do sistema "JDBCcourse" proporcionou uma aplicação prática dos conceitos de modelagem de dados, desenvolvimento backend com Spring Boot e JDBC, e construção de interfaces frontend com tecnologias web básicas. O resultado é uma plataforma funcional para gerenciamento de cursos online, cobrindo as operações CRUD essenciais para as entidades principais do domínio.

Este manual foi elaborado com o intuito de ser um guia compreensivo, facilitando tanto a instalação e configuração inicial do ambiente de desenvolvimento quanto a utilização e o entendimento da arquitetura do sistema por outros desenvolvedores ou usuários técnicos. Espera-se que os detalhes fornecidos sobre a instalação de componentes, execução do projeto, interação com a interface e testes de API sejam suficientes para uma experiência fluida com o JDBCcourse.

Para dúvidas ou sugestões, entre em contato: gablevi@edu.unifor.br ou via redes social presentes em <https://github.com/gbrlevi>

8 Referências

- Spring Boot. *Spring Framework*. Disponível em: <https://spring.io/projects/spring-boot>.

- MySQL Documentation. *Oracle*. Disponível em: <https://dev.mysql.com/doc/>.
- IntelliJ IDEA. *JetBrains*. Disponível em: <https://www.jetbrains.com/idea/documentation/>.
- Git Documentation. *Git SCM*. Disponível em: <https://git-scm.com/doc>.
- MDN Web Docs. *Mozilla Developer Network*. Disponível em: <https://developer.mozilla.org/>.