



Sistema de Delivery

Universidade de Fortaleza

Ciência da Computação - Ambiente de Dados

Professor: Pedro Gabriel

Equipe

- Lucas de Vasconcelos Barreira Carvalho - 2410424
- André Lima - 2410378

Outubro de 2025

1. Introdução

Desenvolvimento e a funcionalidade do Sistema de Delivery, projetado para gerenciar o fluxo completo de pedidos, desde a seleção do restaurante pelo cliente até a atualização do status pelo estabelecimento. O sistema foi construído em Java, utilizando o padrão arquitetural MVC (Model-View-Controller) e persistência de dados via JDBC com MySQL.

Este documento detalha a construção do sistema. O objetivo é fornecer um guia completo para que o projeto seja replicável por qualquer desenvolvedor.

2. Objetivos do Sistema

- O Sistema de Delivery tem como objetivos: Gerenciar o Ciclo de Pedidos: Automatizar o registro, acompanhamento e finalização de pedidos de clientes.
- Controle de Estoque e Cardápio: Permitir que restaurantes gerenciem seus itens de cardápio de forma dinâmica (adicionar, editar, remover).
- Gestão de Usuários e Endereços: Manter o cadastro de clientes e seus múltiplos endereços para entrega.
- Separação de Responsabilidades: Implementar a arquitetura MVC/Camadas (Model, View, Controller, Service, DAO) para garantir código limpo, modular e de fácil manutenção.

3. Funcionalidades Principais

O sistema incluirá as seguintes funcionalidades:

3.1. Gestão de Clientes e Endereços

- Cadastro e login de novos clientes.
 - Gerenciamento de múltiplos endereços de entrega (**EnderecoModel**).
-

3.2. Gestão de Cardápio (Área do Restaurante)

- Cadastro de novos itens de cardápio (**CardapioModel**).
 - Edição e exclusão de itens existentes.
 - Controle de disponibilidade dos itens.
-

3.3. Fluxo de Pedidos e Acompanhamento

- Seleção de restaurante e visualização do cardápio (**ClienteCardapioView**).
- Adição de itens ao carrinho (**ClienteCarrinhoView**).
- Finalização do pedido com seleção de endereço e pagamento.
- Atualização do status do pedido pelo restaurante (**RestauranteAreaView**).
-

4. Tecnologias Utilizadas

O sistema será desenvolvido utilizando as seguintes tecnologias:

| Atributos | Tecnologias |
|------------------------------|-----------------------------------|
| Linguagem de Programação | Java(JDK 21) e SQL |
| Extensões | "Extension Pack for Java" |
| IDE | IntelliJ IDEA |
| Banco de Dados | MySQL 8+ |
| Framework | Nenhum framework utilizado |
| Interface do Usuário | JAVA SWING (JFrame) |
| Conexão com o banco de dados | JDBC (JAVA Database Connectivity) |
| Driver Connection | MySQL Connector/J |

4.1 Explicações das tecnologias

Nesse tópico será apresentado as descrições das ferramentas utilizadas para desenvolver esse projeto.

- **JAVA (JDK 21)** : Linguagem de programação principal do sistema. O JDK contém tudo o que é necessário para criar, compilar e executar programas Java. A versão 21 é moderna, com melhorias de desempenho e novos recursos.
- **MySQL 8+** : Banco de dados relacional utilizado para armazenar todas as informações do sistema (livros, usuários, empréstimos, multas, etc.). Utiliza tabelas e a linguagem SQL para manipulação dos dados.
- **IntelliJ IDEA**: IntelliJ IDEA é um ambiente de desenvolvimento integrado escrito em Java para o desenvolvimento de software de computador escrito em Java, Kotlin, Groovy e outras linguagens baseadas em JVM.
- **JDBC (Java Database Connectivity)**: Tecnologia que permite a comunicação entre o Java e o banco de dados MySQL. É usada para

enviar comandos SQL (como SELECT, INSERT e UPDATE) e receber os resultados diretamente no programa.

- **JFrame (Swing):** Biblioteca do Java utilizada para criar interfaces gráficas (telas). Permite montar janelas com botões, tabelas, menus e campos de texto, sendo a base visual do sistema.
- **MySQL Connector/J:** Driver JDBC específico para o MySQL, responsável por estabelecer a conexão real entre o programa Java e o banco de dados. É uma biblioteca (`.jar`) que precisa ser adicionada ao projeto para que o JDBC funcione corretamente.

5. Requisitos mínimos

Para garantir o correto funcionamento do sistema de gerenciamento de biblioteca, é necessário que o ambiente de desenvolvimento e execução atenda a alguns requisitos básicos de hardware e software. Esses requisitos asseguram que todas as ferramentas, bibliotecas e conexões com o banco de dados operem de forma estável e sem falhas.

5.1 Requisitos de Software

- **Sistema Operacional:** Windows 10 ou superior
- **Java Development Kit (JDK):** Versão 21 ou superior
- **Banco de Dados:** MySQL 8.0 ou superior
- **Driver JDBC:** MySQL Connector/J
- **IDE Recomendada:** Visual Studio Code (VS Code)
- **Extensões Necessárias:**
 - Extension Pack for Java
 - MySQL Shell (opcional para gerenciamento do banco)

Bibliotecas Utilizadas:

- `java.sql` (para conexão e manipulação de dados)
- `javax.swing` (para criação das interfaces gráficas)
- `java.time` (para manipulação de datas e prazos de empréstimos)

6. LINKS OFICIAIS DE DOWNLOAD

A seguir estão os links oficiais para download das ferramentas e dependências utilizadas no desenvolvimento e execução do sistema de gerenciamento de

biblioteca. Recomenda-se sempre utilizar fontes oficiais para garantir segurança e compatibilidade.

| | |
|-------------------|---|
| Java JDK 21 | https://www.oracle.com/java/technologies/javase/jdk21-archive-downloads.html |
| IntelliJ IDEA | https://www.jetbrains.com/idea/download/ |
| MySQL | https://dev.mysql.com/downloads/installer/ |
| MySQL Connector/J | https://dev.mysql.com/downloads/connector/j/ |
| | |

7. Instalação

A seguir, são apresentados os passos necessários para instalação e configuração dos programas utilizados no desenvolvimento do projeto. A ordem de instalação recomendada é: **Java 21**, IntelliJ IDEA, **MySQL** e **MySQL Connector/J**.

7.1 Java JDK

O **Java Development Kit (JDK 21)** é essencial para compilar e executar o código-fonte do projeto. Ele fornece as ferramentas necessárias para o desenvolvimento em linguagem Java.

1. Acesse o site oficial do **Java**, disponibilizado pela **Oracle**, conforme o link apresentado anteriormente.
2. Faça o download do instalador compatível com o seu sistema operacional (Windows, Linux ou macOS). Estaremos utilizando Windows como sistema operacional.

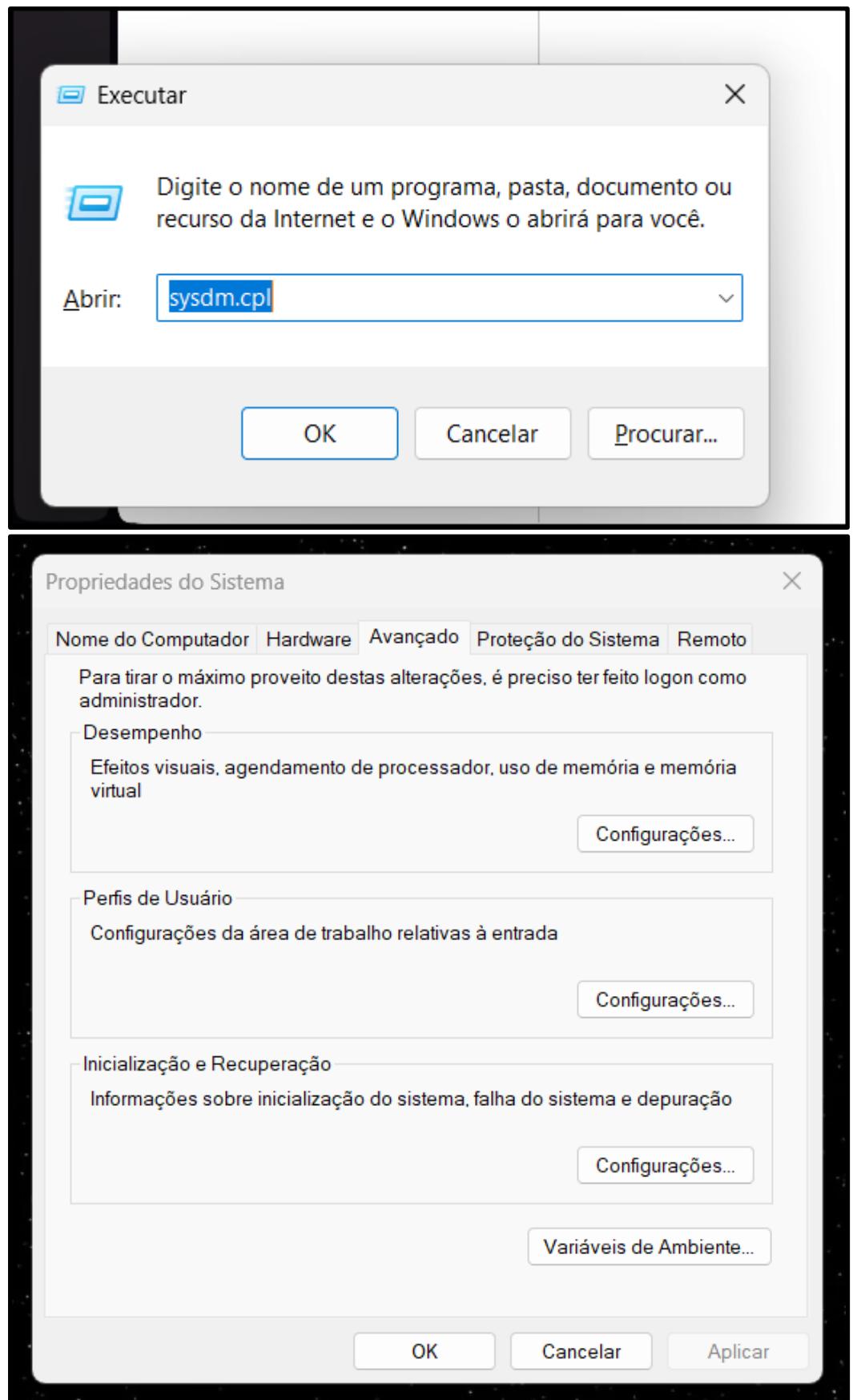
Windows x64 Installer

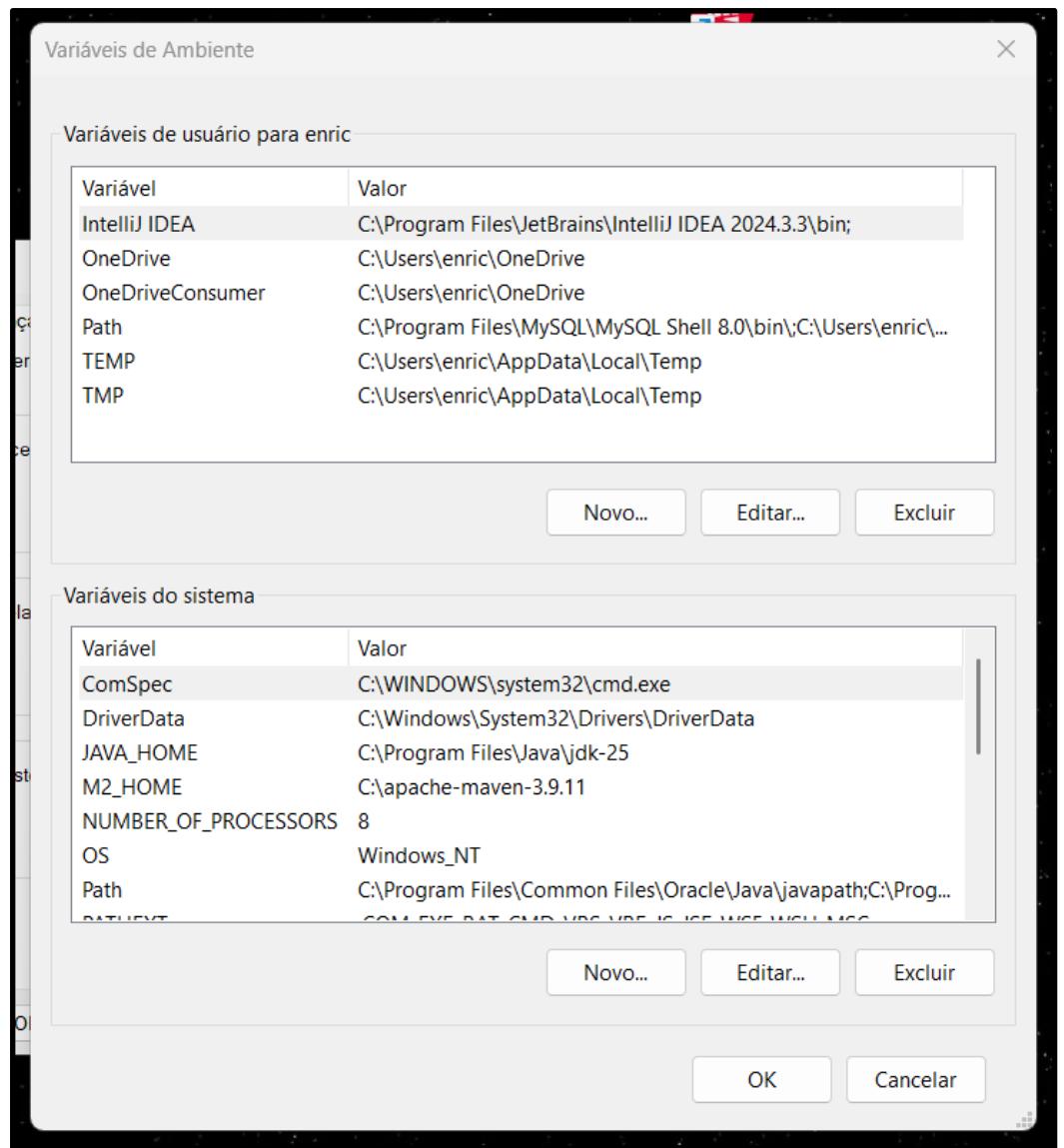
164.42 MB

https://download.oracle.com/java/21/archive/jdk-21.0.8_windows-x64_bin.exe (sha256)

3. Execute o instalador e siga o assistente de instalação.
4. Quando o download terminar, clique em close.
 - a. No Windows, acesse:

- i. Windows + R -> Digite sysdm.cpl -> Avançado -> Variáveis de Ambiente





- ii. Em Variáveis do sistema, crie uma nova variável (clique em "Novo..."), chamada **JAVA_HOME**, apontando para o diretório de instalação.
 1. Exemplo:
Nome da variável: **JAVA_HOME**
Valor da variável: **C:\Program Files\Java\jdk-21**
- iii. Em seguida, edite a variável Path e adicione:
 1. **%JAVA_HOME%\bin**

Variáveis do sistema

| Variável | Valor |
|----------------------|---|
| ComSpec | C:\WINDOWS\system32\cmd.exe |
| DriverData | C:\Windows\System32\Drivers\DriverData |
| JAVA_HOME | C:\Program Files\Java\jdk-25 |
| M2_HOME | C:\apache-maven-3.9.11 |
| NUMBER_OF_PROCESSORS | 8 |
| OS | Windows_NT |
| Path | C:\Program Files\Common Files\Oracle\Java\javapath;C:\Prog... COM;EVE;BAT;CMD;VBS;VBE;JS;JSE;WCF;WSH;MSC |
| SHELL | |

Novo... Editar... Excluir

OK Cancelar

iv.

Variáveis de Ambiente

Editar a variável de ambiente

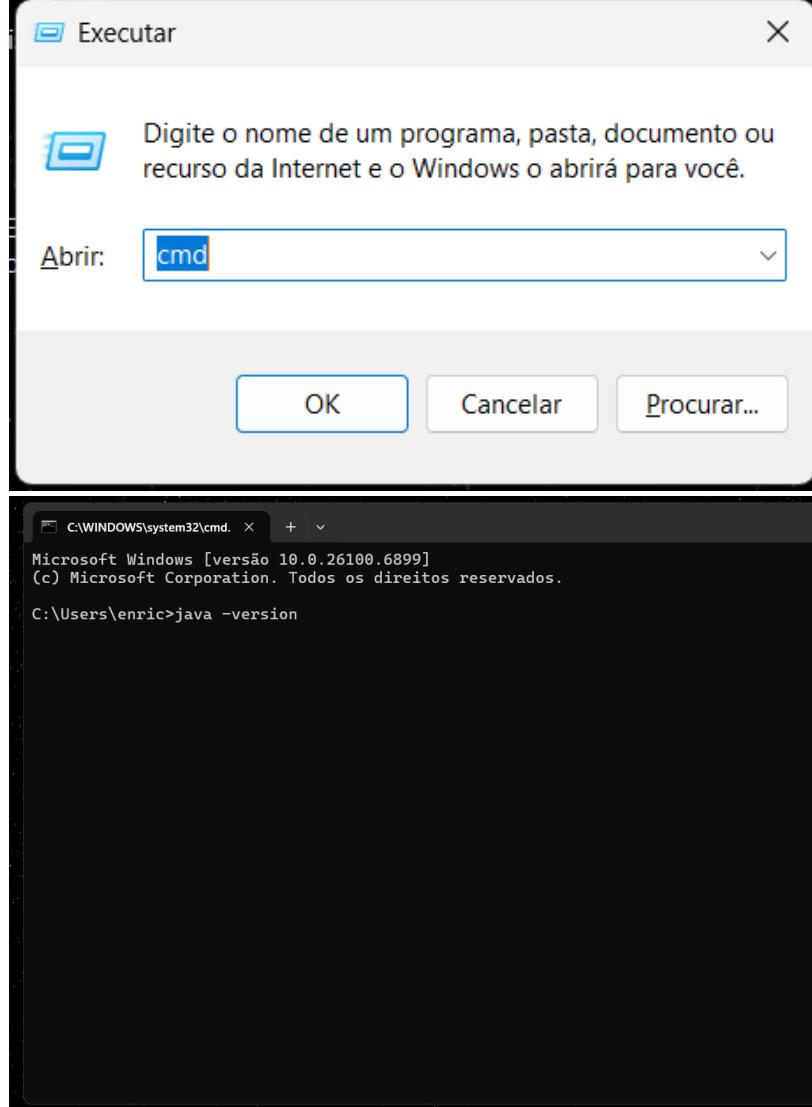
| Variável | Valor |
|----------|---|
| Path | C:\Program Files\Common Files\Oracle\Java\javapath C:\Program Files\Java\jdk-23.0.2\bin C:\WINDOWS\system32 C:\WINDOWS C:\WINDOWS\System32\Wbem C:\WINDOWS\System32\WindowsPowerShell\v1.0\ C:\WINDOWS\System32\OpenSSH\ C:\Program Files\Git\cmd c:\Users\enric\AppData\Local\Programs\cursor\resources\app\b... C:\Program Files\nodejs\ C:\Program Files\Microsoft SQL Server\150\Tools\Binn\ C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\170\To... C:\Program Files\dotnet\ C:\Program Files (x86)\Windows Kits\10\Windows Performance ... C:\msys64\mingw64\bin C:\apache-maven-3.9.11\bin %JAVA_HOME%\bin |

Novo Editar Procurar... Excluir Mover para Cima Mover para baixo Editar texto...

OK Cancelar

OK Cancelar

5. Para verificar se a instalação foi concluída corretamente, abra o **Prompt de Comando (cmd)** e digite:
- No Windows, acesse:
 - Windows + R -> Digite cmd



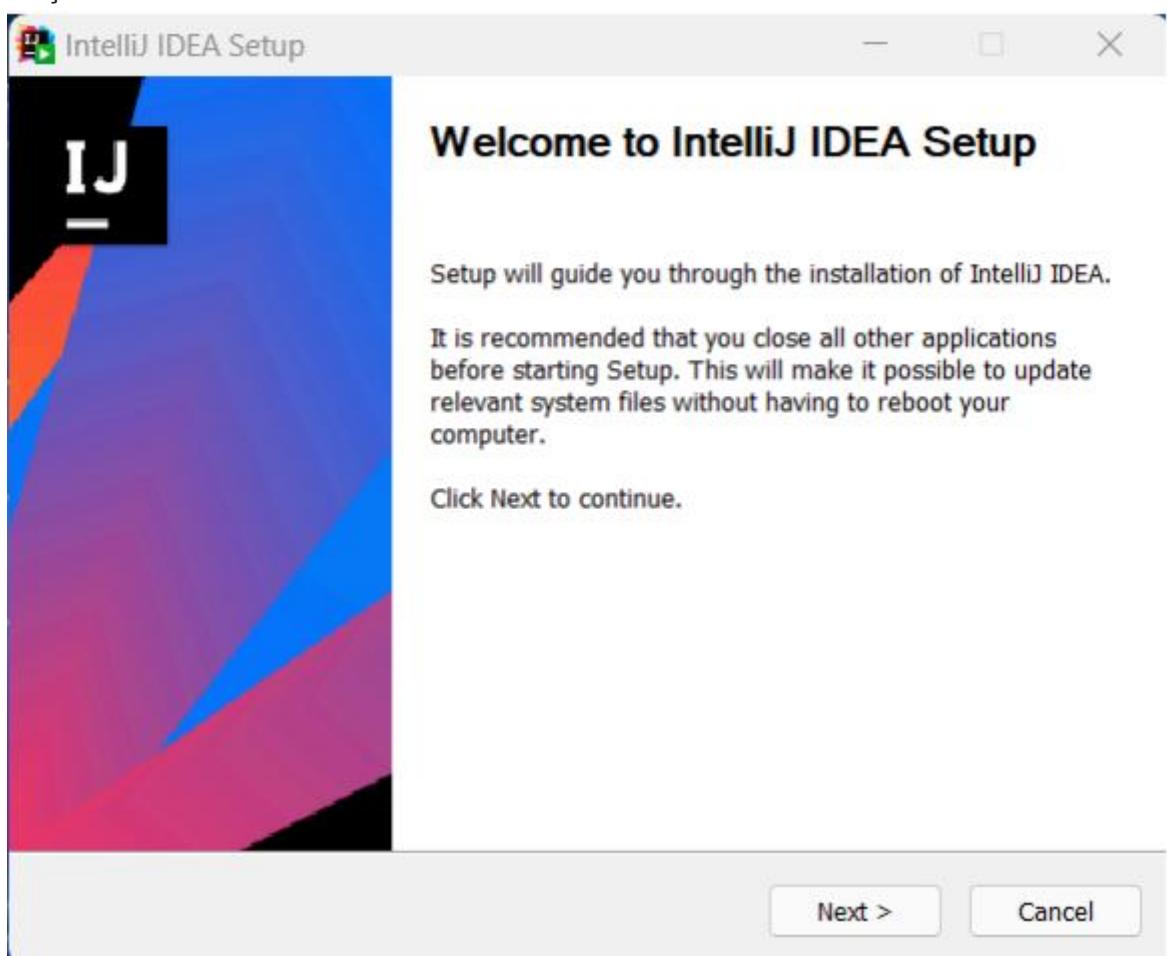
- Para verificar se a instalação foi concluída corretamente, abra o **Prompt de Comando** e digite: **java -version**

7.2 IntelliJ IDEA

- 7.2. Instalação da IDE (IntelliJ IDEA) A IDE (Integrated Development Environment) é o ambiente de trabalho onde o código será escrito,

compilado e executado. O IntelliJ IDEA é a IDE mais recomendada para projetos Java, devido à sua robustez e funcionalidades.

2. 7.2.1 Download e Execução do Instalador Acesse o site oficial da JetBrains e baixe a versão Community (gratuita) do IntelliJ IDEA para o seu sistema operacional (Windows, macOS ou Linux).
3. Link Oficial: <https://www.jetbrains.com/idea/download/>
4. Localize o arquivo de instalação (ex: ideaC-202X.X.exe) e execute-o. 3. Na tela de boas-vindas do Setup, clique em Next > para continuar.
5. Faça o **download** e execute o **instalador**.



7.2.2 Configuração do Diretório e Opções

1. Escolha do Local de Instalação: Mantenha o diretório padrão sugerido pelo instalador, a menos que você tenha uma razão específica para alterá-lo. Clique em Next >

2. Opções de Instalação: Esta etapa é crucial para facilitar o uso da IDE. Marque as seguintes opções: Create Desktop Shortcut: Marque a opção de 64-bit launcher para criar um atalho na área de trabalho. Update PATH variable (restart needed): Marque esta opção para poder iniciar o IntelliJ de qualquer lugar no terminal. Create Associations: Marque a opção .java para que o IntelliJ seja o programa padrão para abrir arquivos Java.

3. Clique em Next >.

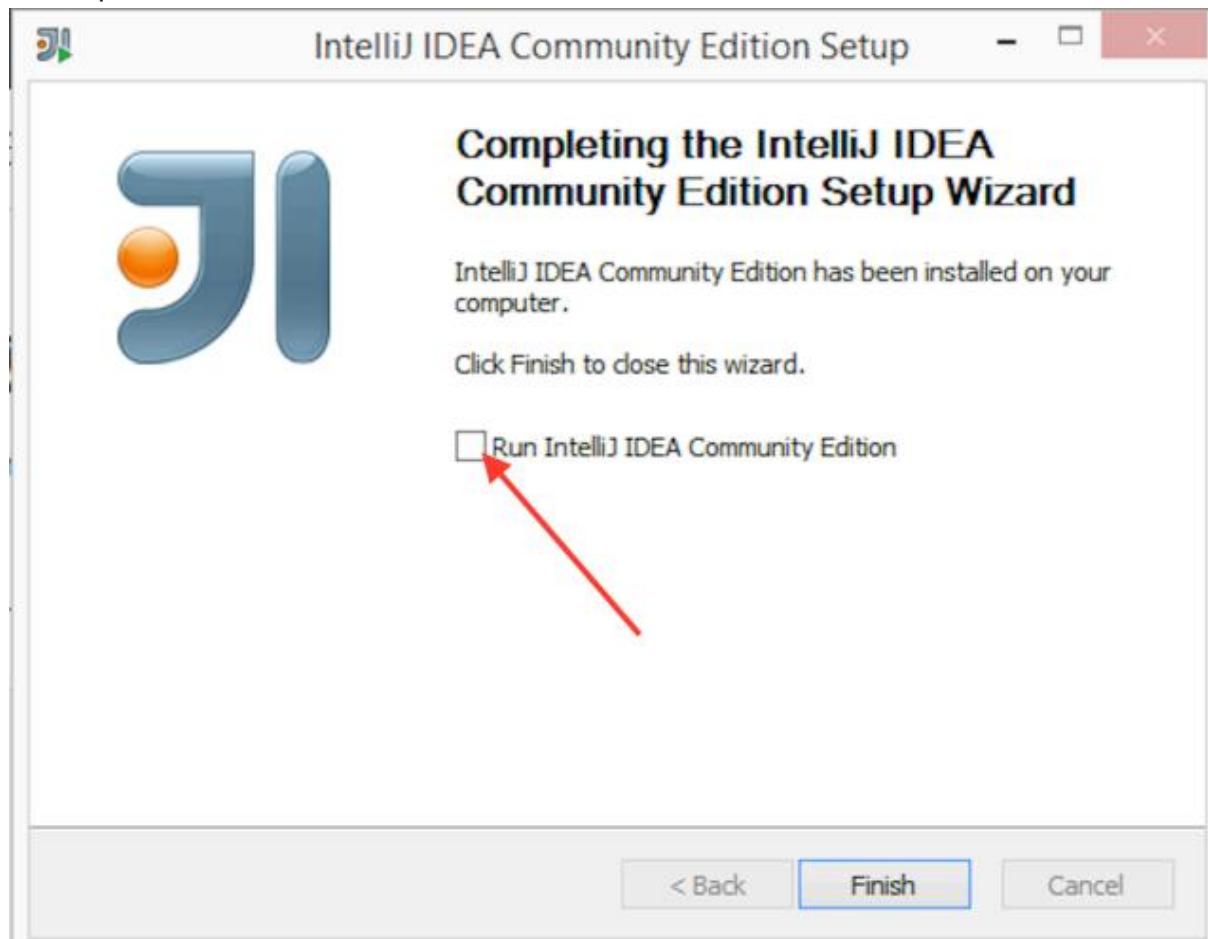
7.2.3 Início da Instalação

1. Escolha da Pasta do Menu Iniciar: Mantenha a pasta padrão (JetBrains) e clique em Install.
2. Aguarde o processo de instalação.

7.2.4 Finalização e Execução

1. Após a conclusão, o instalador exibirá a tela "Completing the IntelliJ IDEA Community Edition Setup".
2. Finalize a Instalação: Marque a opção Run IntelliJ IDEA Community Edition se desejar abrir a IDE imediatamente

3. Clique em finish



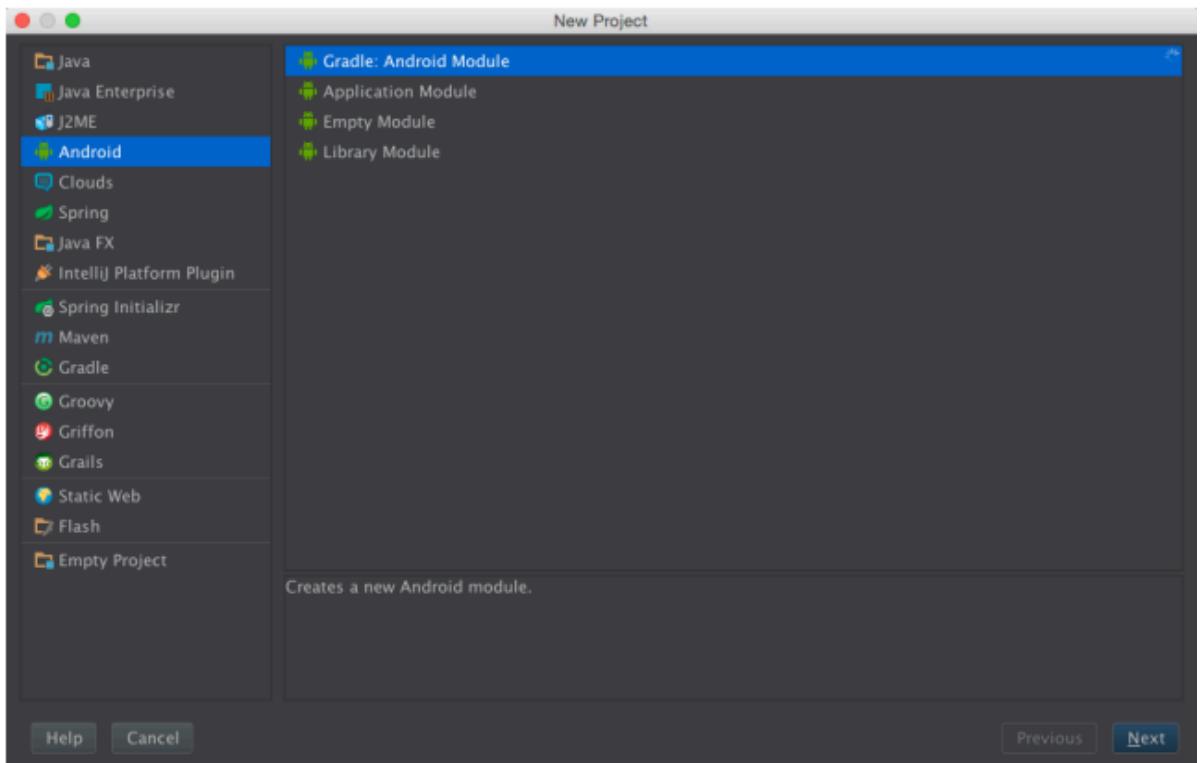
7.2.5 Configuração Inicial do Projeto (Associação do JDK)

1. Primeira Execução: Ao abrir o IntelliJ pela primeira vez ou ao criar um novo projeto, você precisará configurar o JDK (Java Development Kit) que foi instalado anteriormente.

2. Configuração do SDK: Vá em File -> Project Structure -> Project SDK .

Selecione o JDK instalado (ex: 19 Oracle OpenJDK version 19). Se o IntelliJ não detectar automaticamente, clique em Add SDK e navegue até a pasta de instalação do seu JDK.

Clique em OK para salvar as configurações



Com isso, o IntelliJ IDEA está pronto para ser utilizado no desenvolvimento do seu Sistema de Delivery.

7.4 MySQL Workbench

O **MySQL Workbench** é uma ferramenta gráfica oficial da Oracle utilizada para **administrar bancos de dados MySQL, criar tabelas, executar comandos SQL e gerenciar conexões**.

1. Acesse o link disponibilizado do MySQL nos links recomendados.

2. Escolha o seu sistema operacional e a versão, escolha a opção de 558.3M

MySQL Installer 8.0.44

Note: MySQL 8.0 is the final series with MySQL Installer. As of MySQL 8.1, use a MySQL product's MSI or Zip archive for installation. MySQL Server 8.1 and higher also bundle MySQL Configurator, a tool that helps configure MySQL Server.

Select Version:

8.0.44

Select Operating System:

Microsoft Windows

| | | | |
|--|--------|--------|--------------------------|
| Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-8.0.44.0.msi) | 8.0.44 | 2.1M | Download |
| Windows (x86, 32-bit), MSI Installer (mysql-installer-community-8.0.44.0.msi) | 8.0.44 | 558.3M | Download |

3. Depois de clicar em download, o site pode pedir para fazer login na conta Oracle, mas **isso não é obrigatório**. Basta clicar em “**No thanks, just start my download**” para iniciar o download direto.

MySQL Community Downloads

[Login Now](#) or [Sign Up](#) for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

[Login »](#)

using my Oracle Web account

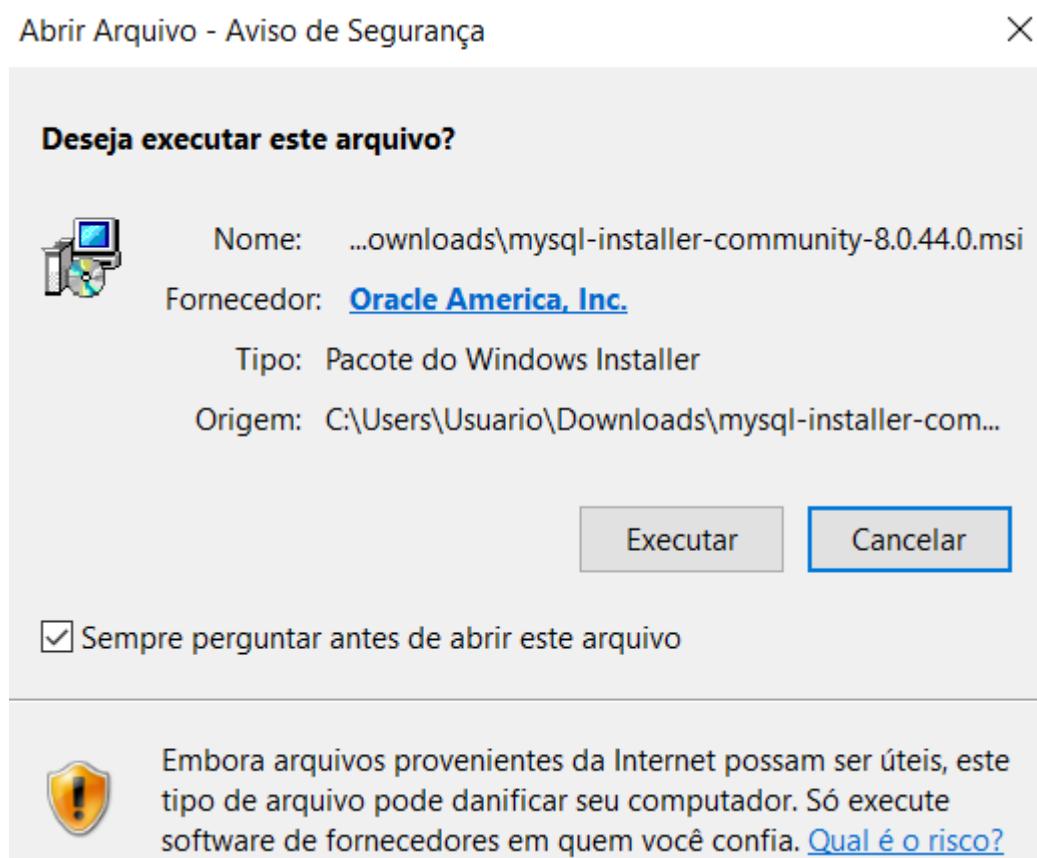
[Sign Up »](#)

for an Oracle Web account

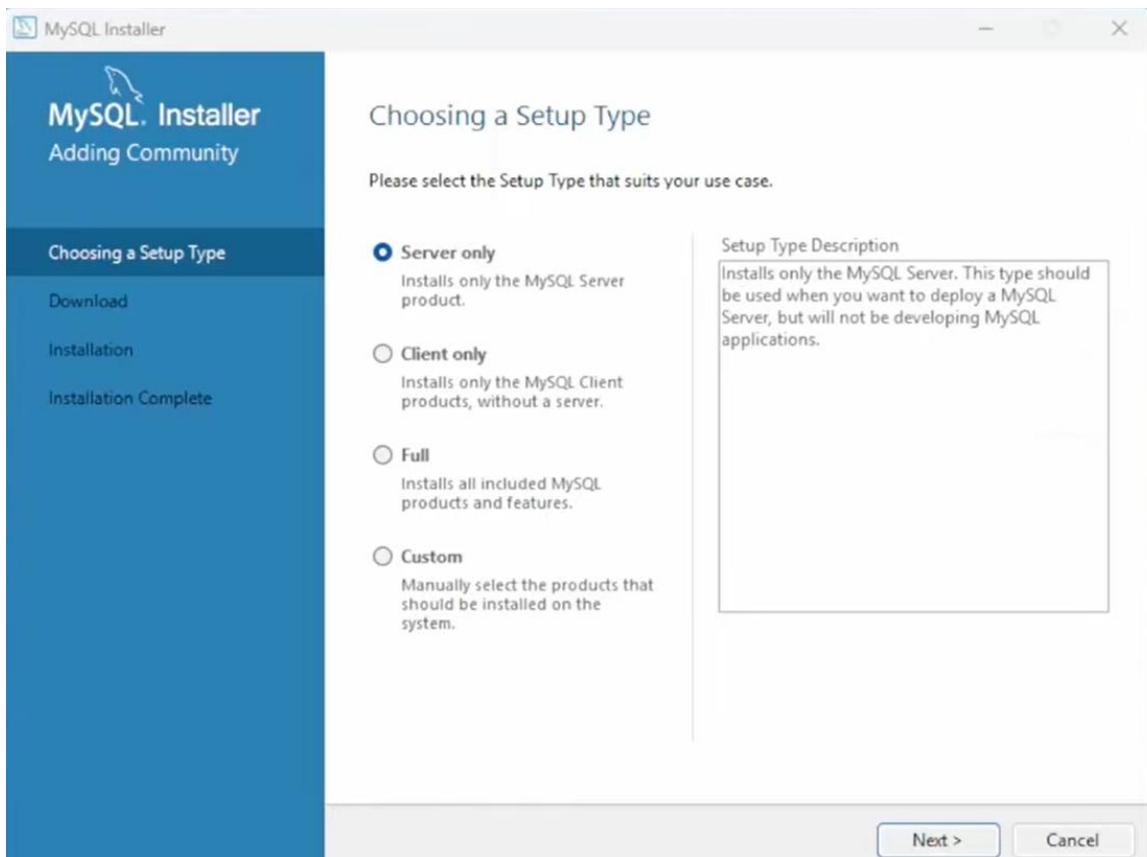
MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

[No thanks, just start my download.](#)

4. Clique no instalador do MySQL e aceite as permissões.



5. Quando terminar o download do instalador, clique em executar, após isso, irá aparecer uma tela para escolher o tipo de download do sql, selecione a opção full e clique em next.

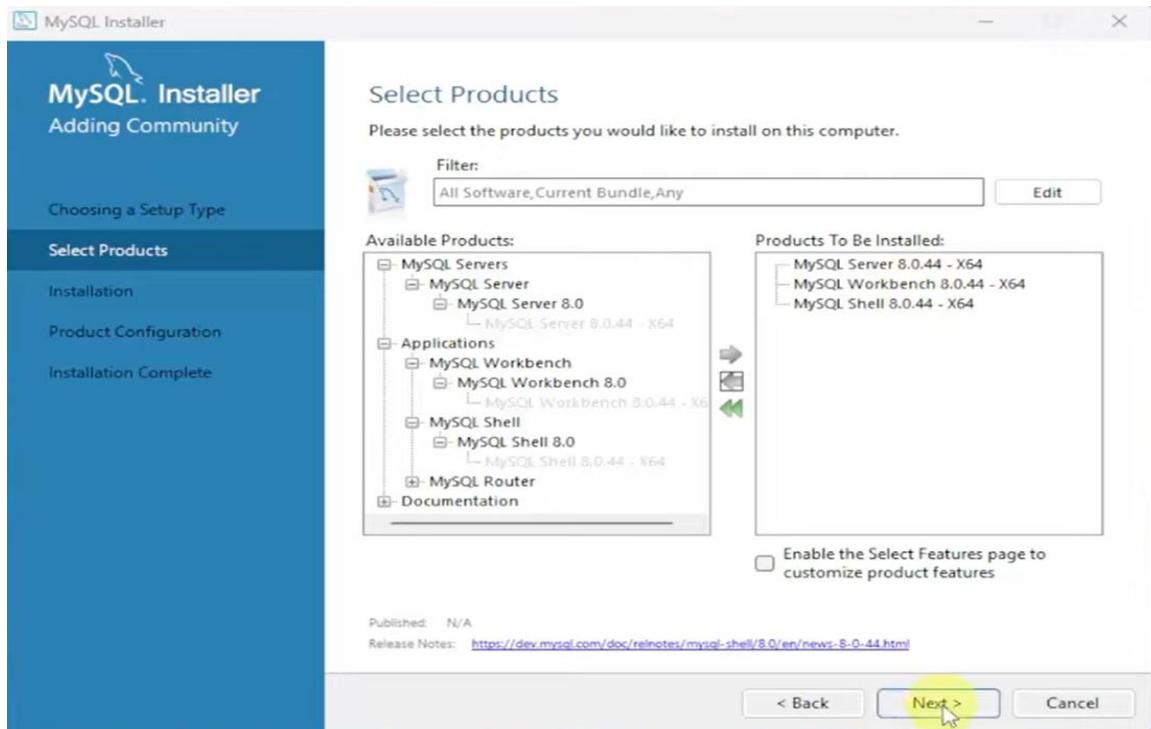


6. Agora vamos selecionar os produtos a serem instalados, clique no quadrado com o símbolo de "+" do MySQL Servers, depois MySQL Server, selecione MySQL Server 8.0 e depois clique na seta apontando para a direita "->". Clique no símbolo de "+" do Applications, em seguida MySQL Workbench, selecione MySQL Workbench 8.0 e clique na seta "->", ainda em applications, clique em MySQL Shell e depois selecione MySQL Shell 8.0 e clique na seta e finalize em next.
 - a. MySQL Servers -> MySQL Server -> MySQL Server 8.0.44 x64
 - b. Applications -> MySQL Workbench -> MySQL Workbench 8.0 -> MySQL Workbench 8.0.44 x64
 - c. Applications -> MySQL Shell -> MySQL Shell 8.0 -> MySQL Shell 8.0.44 x64

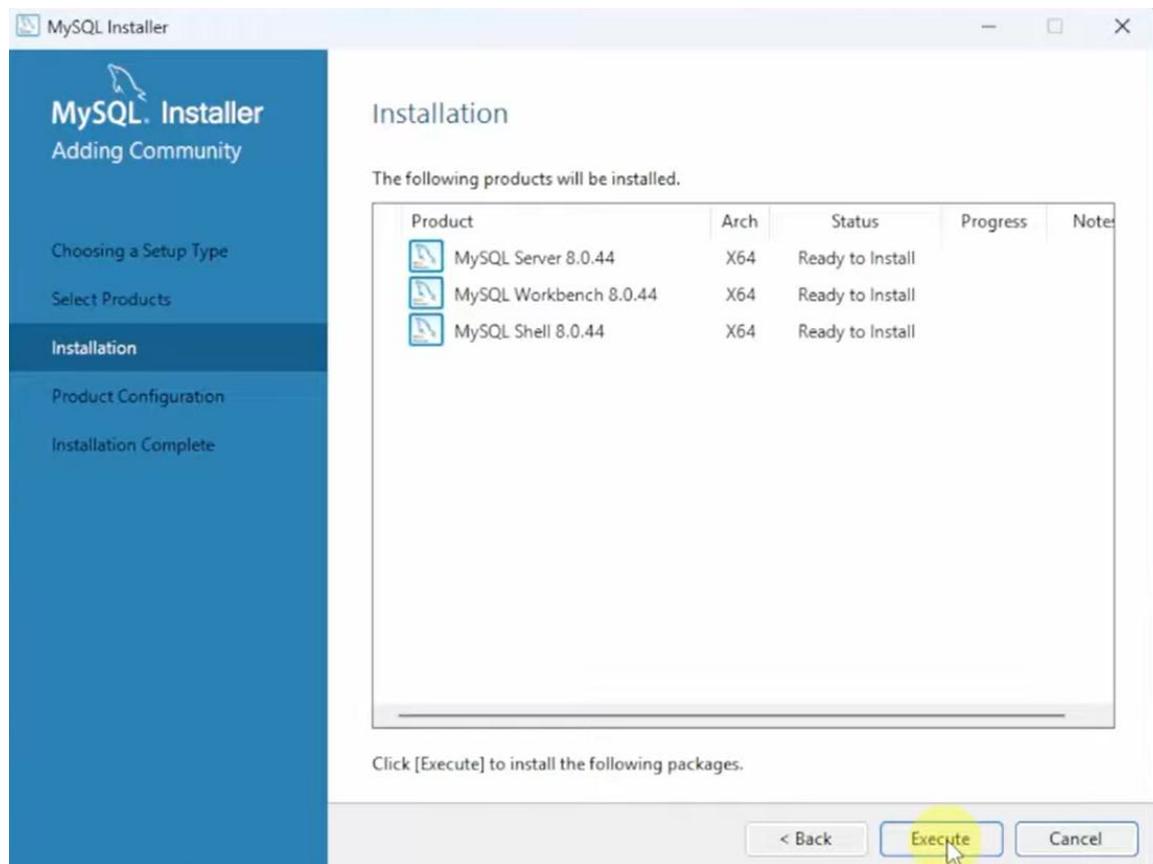
De forma mais compreensível e visual de entender:

 - d. MySQL Servers
 - i. MySQL Server

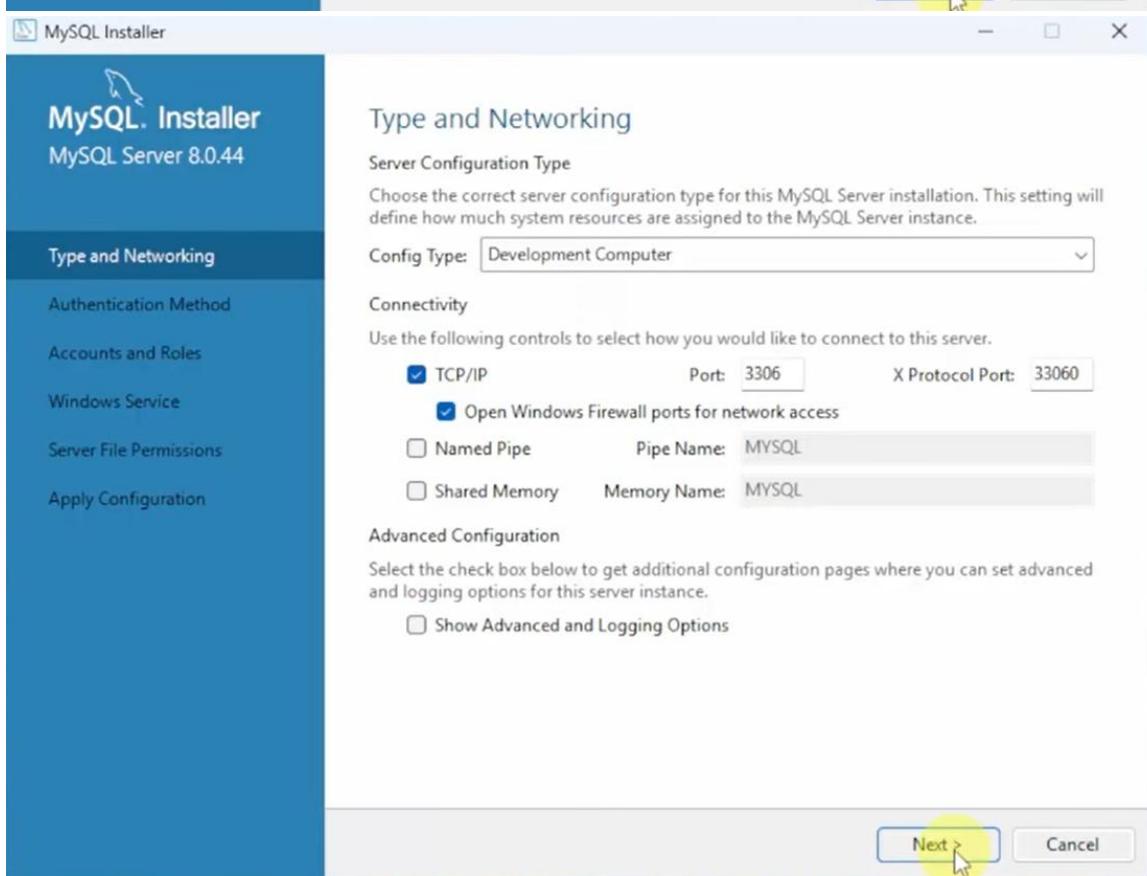
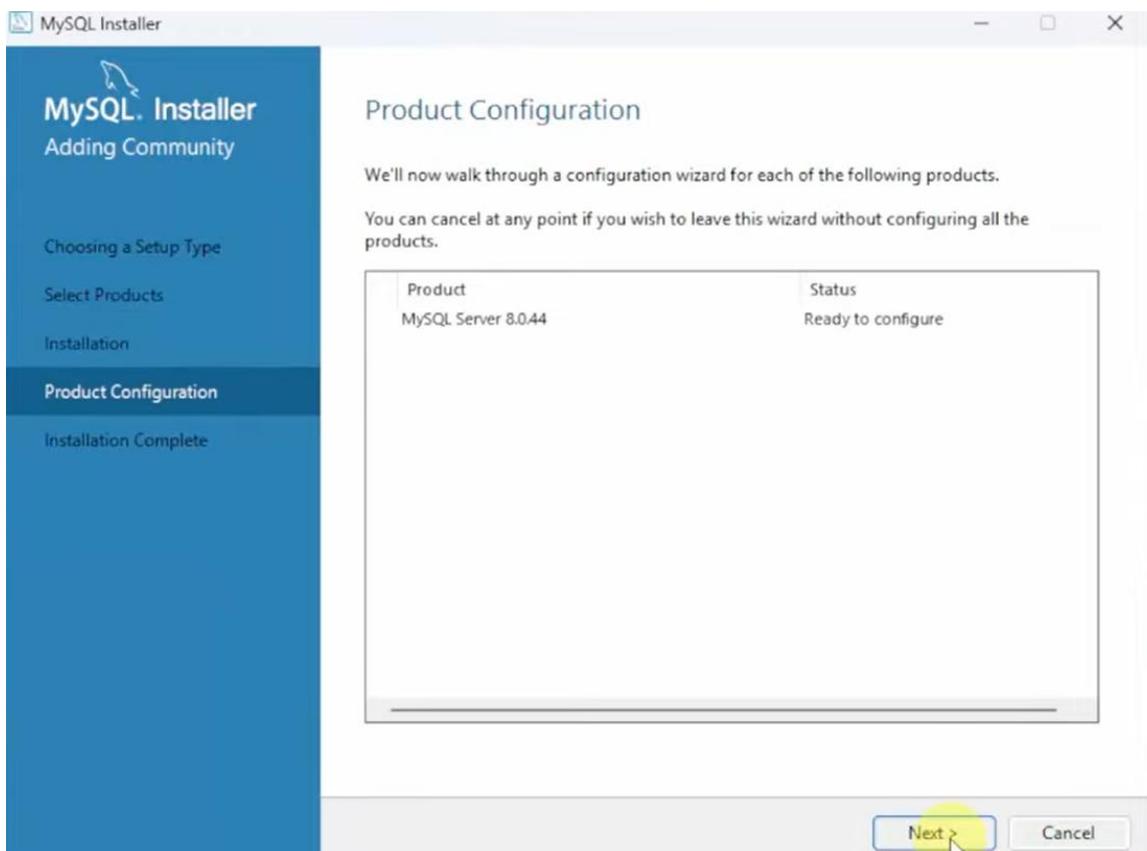
1. MySQL Server 8.0
 - a. MySQL 8.0.44 x64
- e. Applications
 - i. MySQL Workbench
 1. MySQL Workbench 8.0
 - a. MySQL Workbench 8.0.44 x64
 - ii. MySQL Shell
 1. MySQL MySQL Shell 8.0
 - a. MySQL Shell 8.0.44 x64



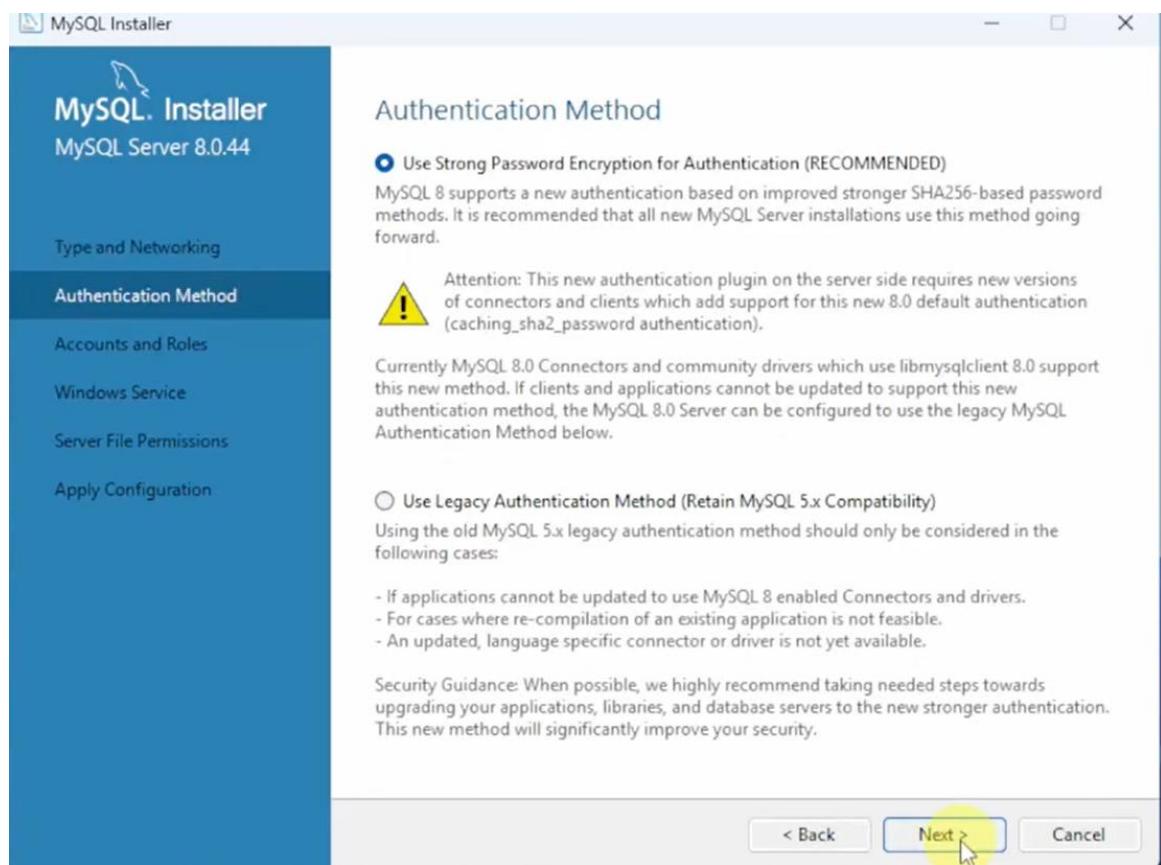
7. Agora vamos, **instalar os produtos**. clique em executar e espere, quando terminar prossiga.



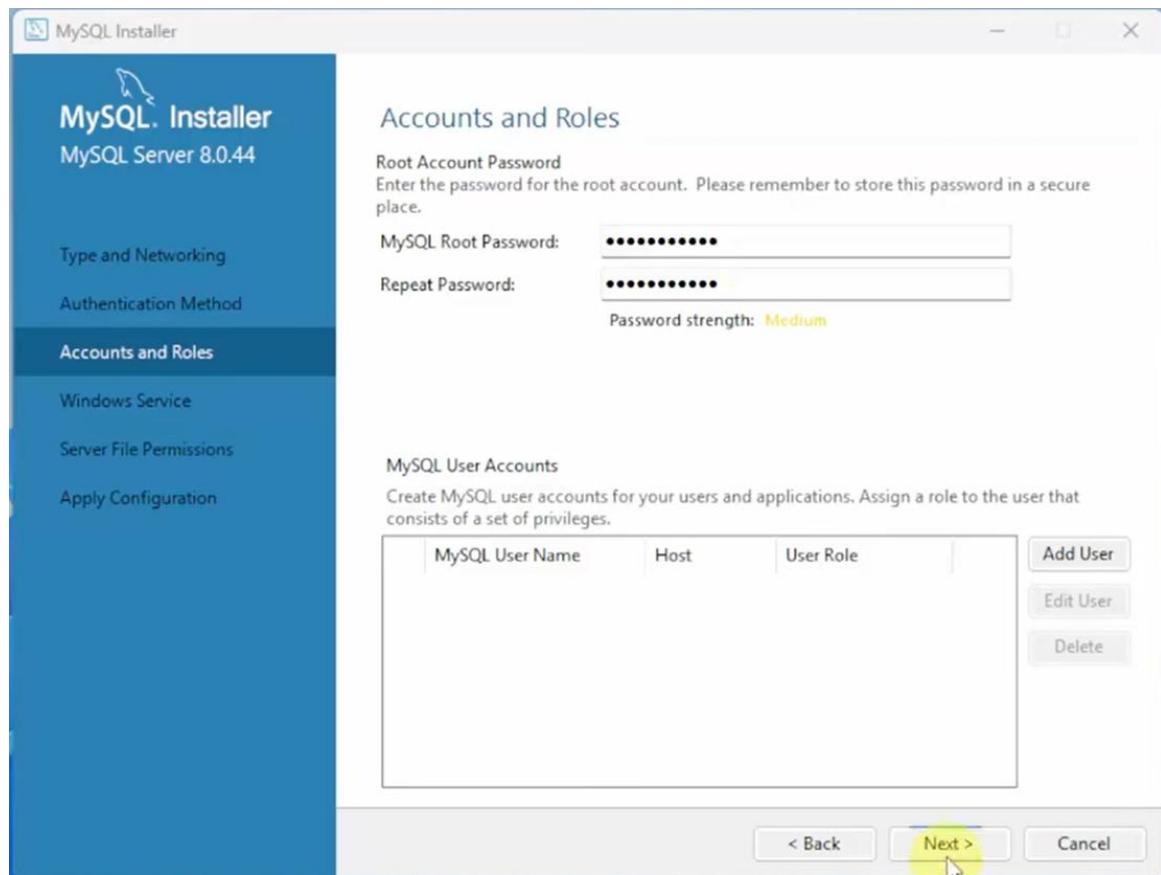
8. Em Product Configuration e Type and Networking, clique em next.



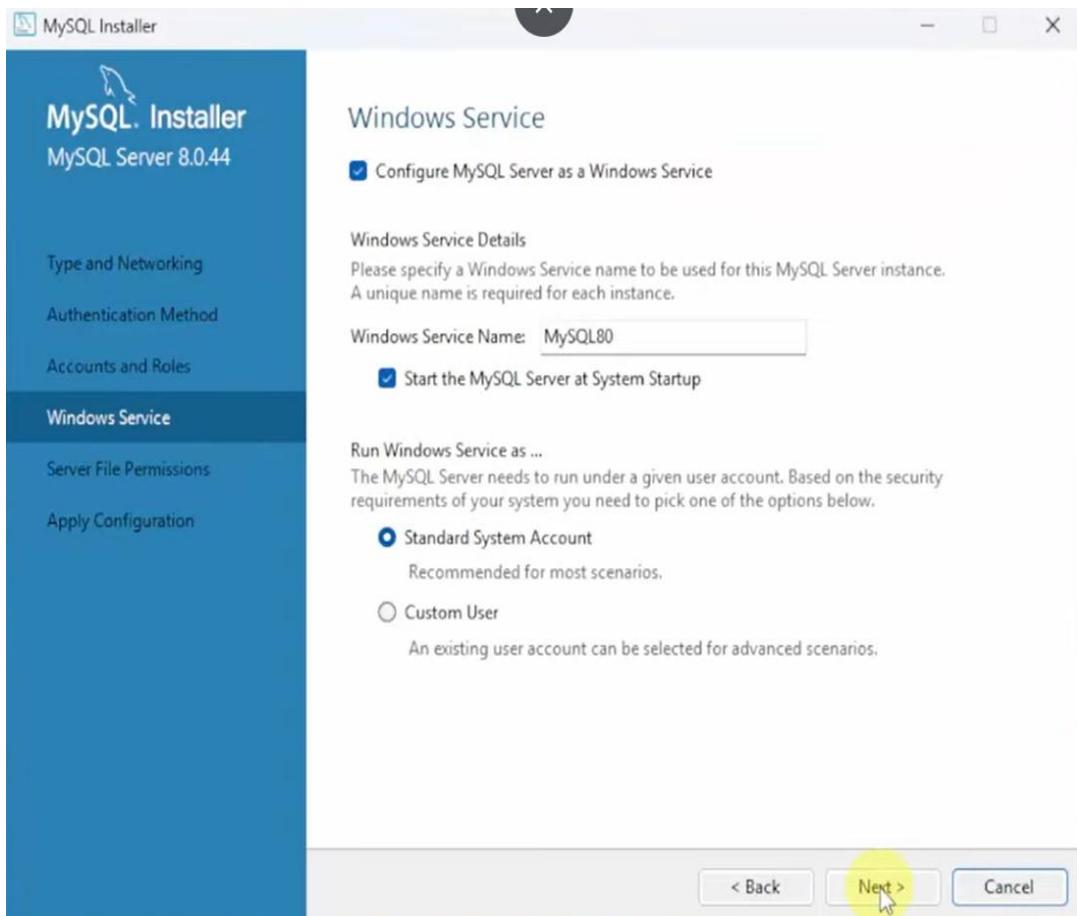
9. Em Authentication Method, selecione a opção (**use strong password encryption**), para garantir mais segurança no acesso dos bancos de dados.



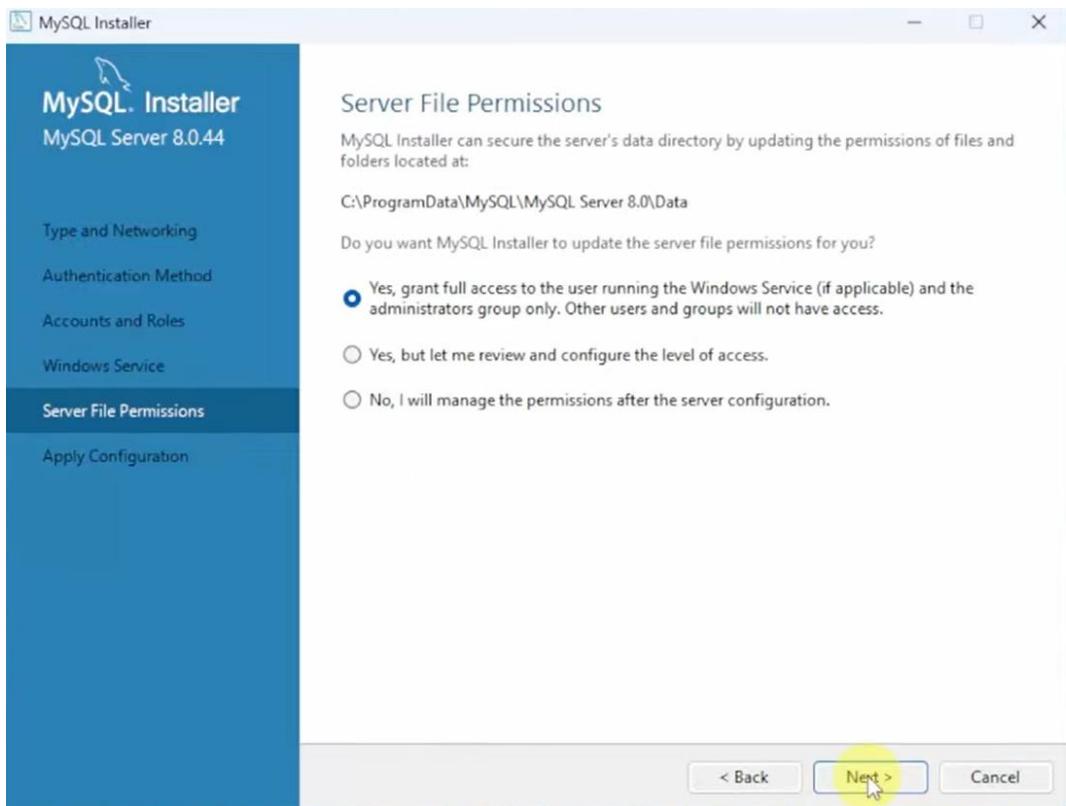
10. Agora, em Accounts and Roles, crie sua senha, guarde ela em algum lugar seguro, ela vai ser a chave de acesso pros bancos de dados relacionais.



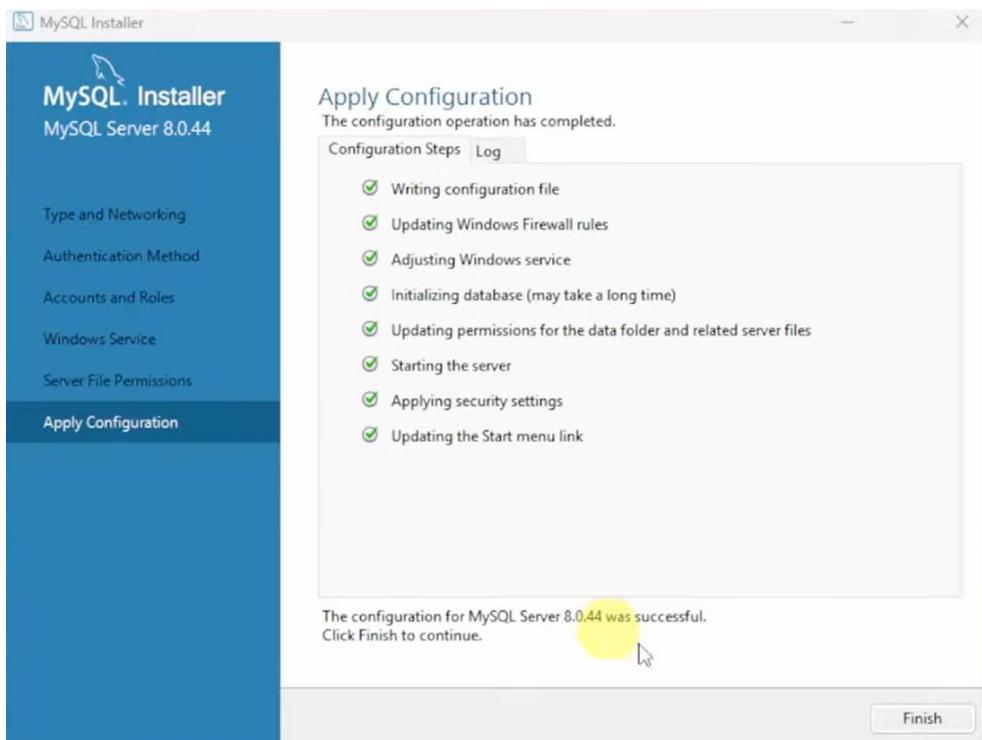
11. Após o processo de criar a senha, em Windows Service, localize “Run Windows Service as...” e deixe em “Standard System Account”.



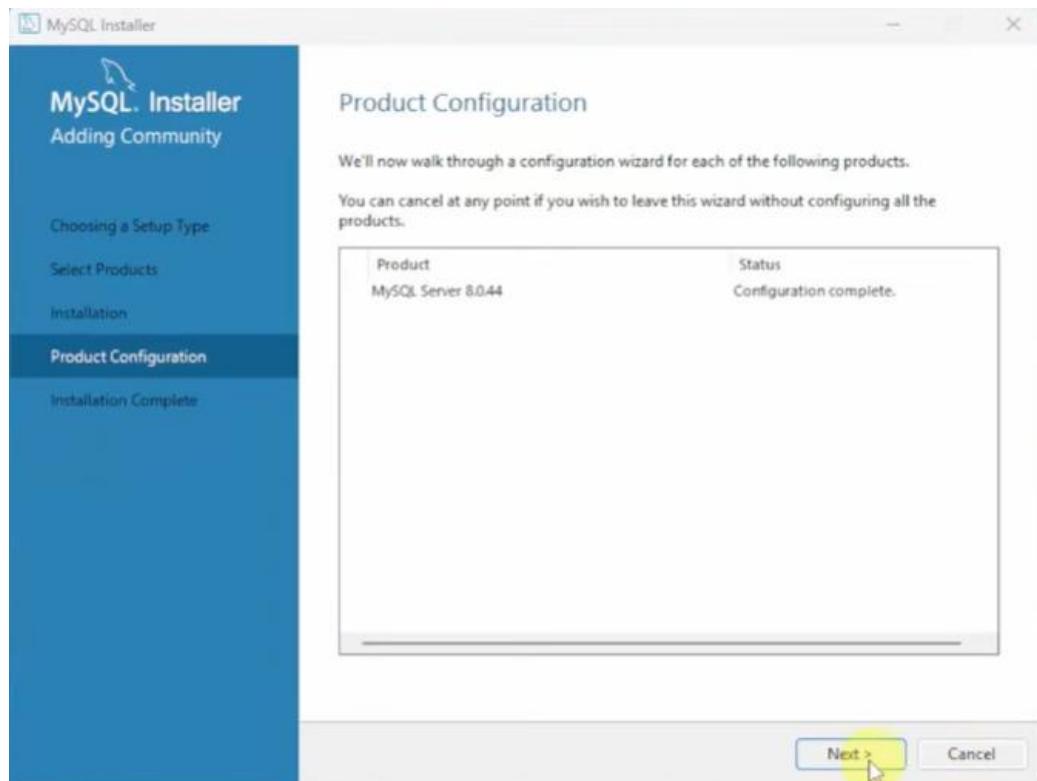
12. Em Server File Permissions, deixe a primeira opção marcada.



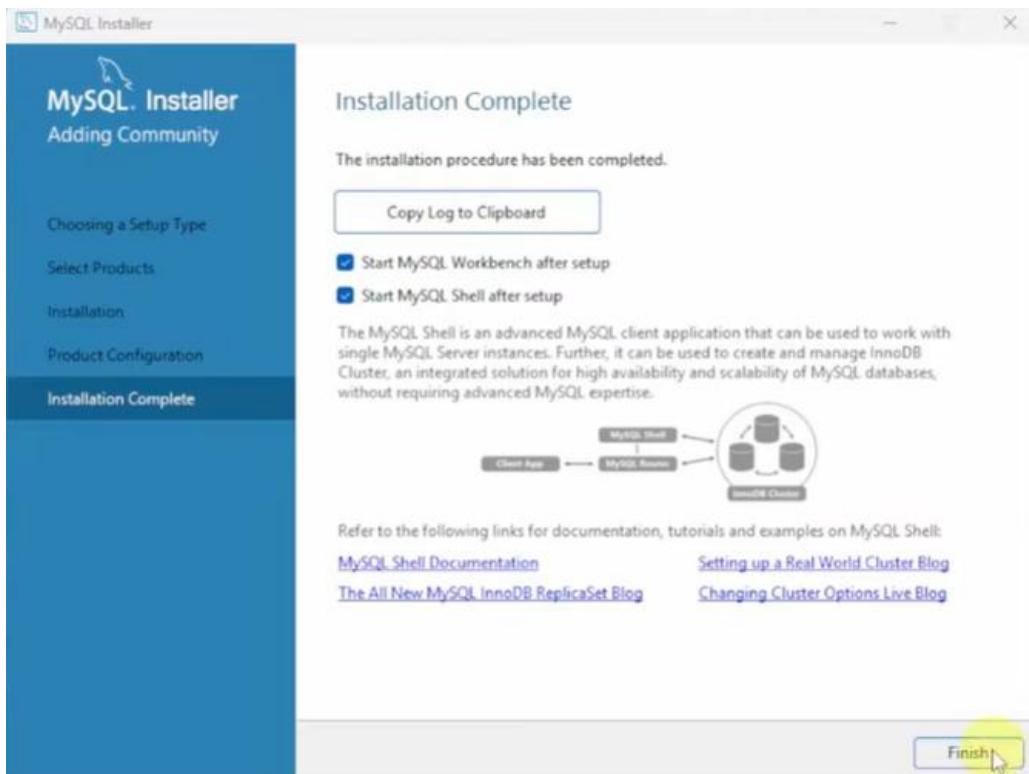
13. Agora em Apply Configurations, clique em execute e quando terminar clique em finish.



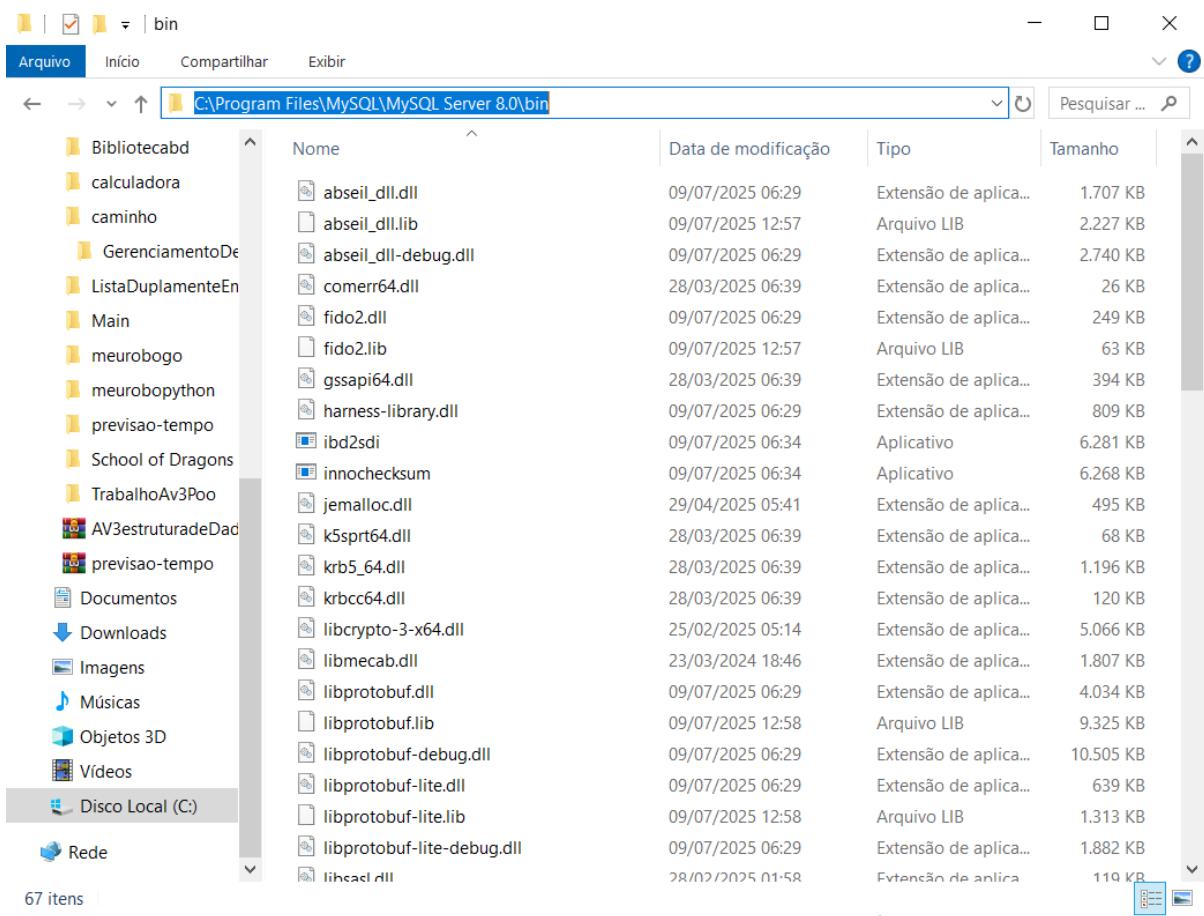
14. Clique em next



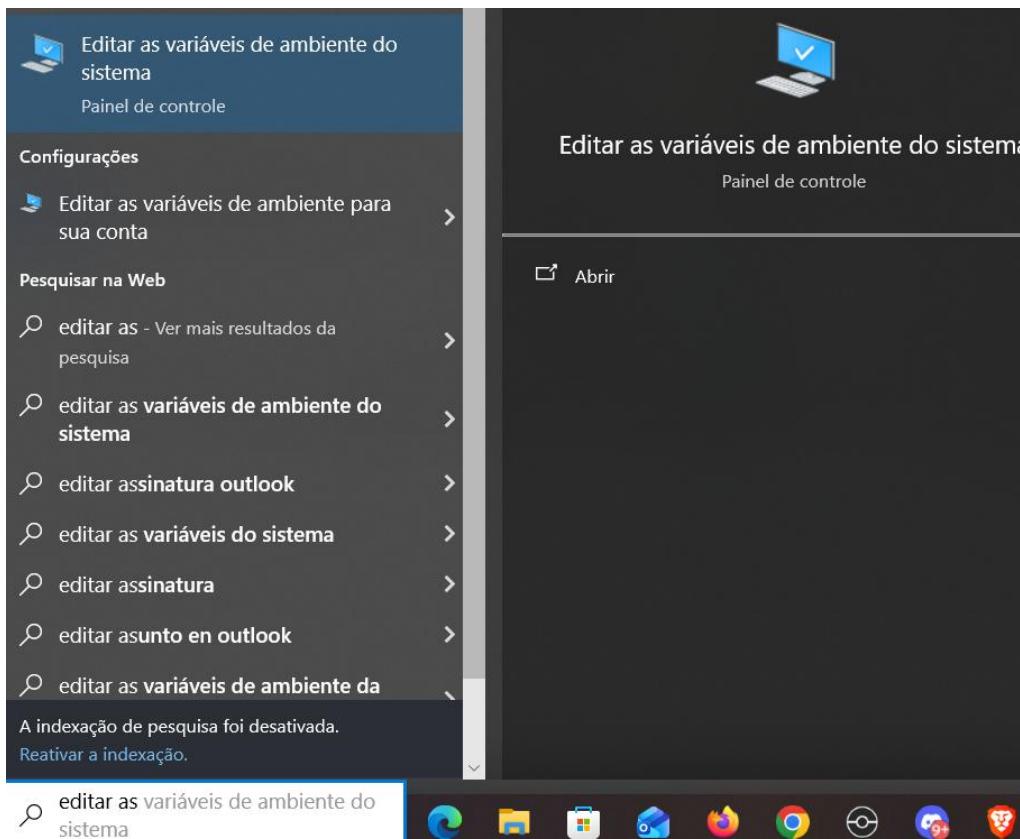
15. Em Installation Complete, deixe as duas opções selecionadas. clique em finish, e espere o MySQL abrir.



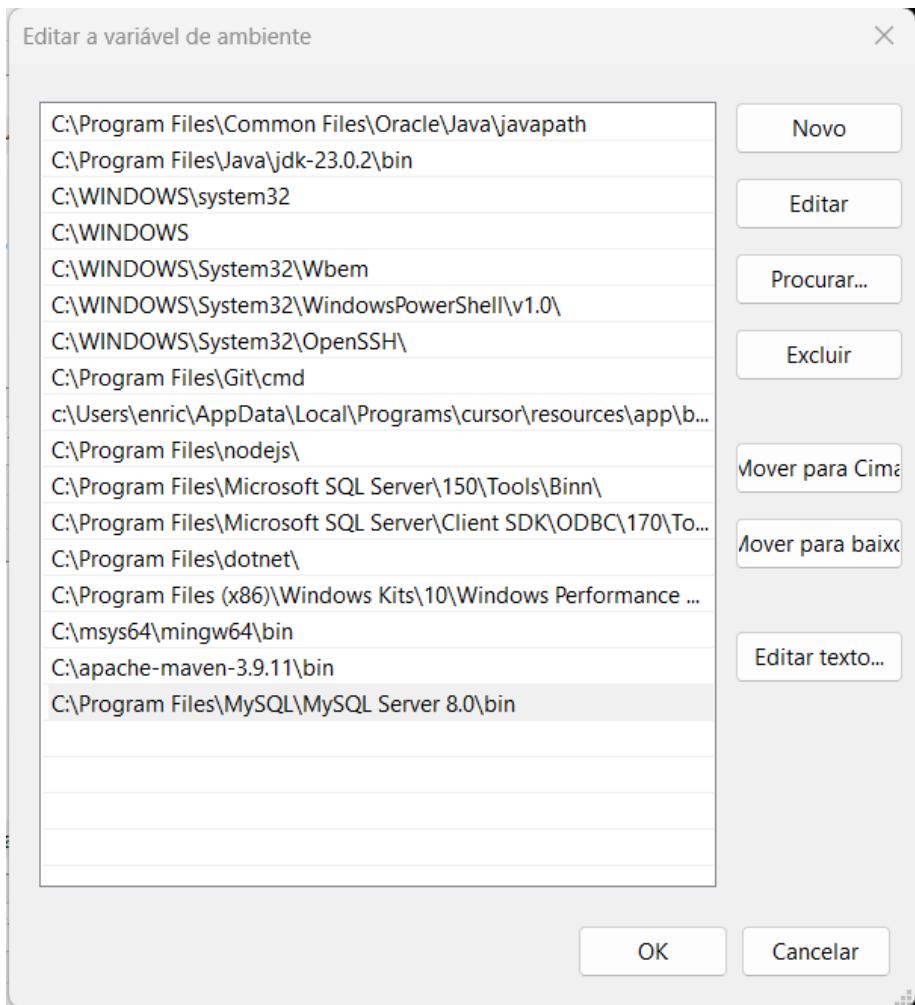
16. Agora vamos adicionar o bin da pasta do MySQL server 8.0 ao path.
abra seu explorador de arquivos, copie o caminho onde o bin do MySQL
está criada.
- Exemplo a seguir:**
Disco Local -> Arquivos de programas -> MySQL -> MySQL Server 8.0 -
> bin. Copie o caminho **(C:\Program Files\MySQL\MySQL Server
8.0\bin)**



17. Clique na barra de pesquisa do windows e escreva “editar as variáveis de ambiente”, clique em abrir.



18. Em Propriedades do Sistema, vá em Avançado -> Variáveis de Ambiente -> Variáveis do Sistema. Em Path, adicione o caminho da pasta bin do MySQL.
- **Usando o exemplo anterior do passo 16:**



7.5 MySQL Connector/J (JDBC)

O **JDBC (Java Database Connectivity)** é a API padrão do Java que permite a comunicação entre aplicações Java e bancos de dados relacionais (como o MySQL). Ele funciona por meio de **drivers**, e o **MySQL Connector/J** é o driver JDBC oficial desenvolvido pela Oracle para conectar o Java ao MySQL.

1. Acesse o link disponibilizado anteriormente do MySQL Connector/J, ele o levará ao site da Oracle.
2. Selecione a versão mais recente e o sistema operacional do seu computador, como estamos usando Windows, escolha a opção Platform Independent.
3. Após isso, realize o download.
4. Clique em "**No thanks, just start my download**".
5. Aguarde o download do arquivo .zip.
6. Descompacte o arquivo zip clicando com o botão direito -> extrair tudo.

7. Dentro da pasta descompactada procure o arquivo principal do driver: **mysql-connector-j-8.x.x.jar**.
8. Coloque esse arquivo na área de trabalho para facilitar a implementação.

8. Configuração

Nesta etapa, será configurada a estrutura inicial do projeto Java no **IntelliJ IDEA**. A criação correta das pastas e arquivos é essencial para manter a **organização, modularidade e fácil manutenção** do código.

1. Iniciando o Assistente de Novo Projeto

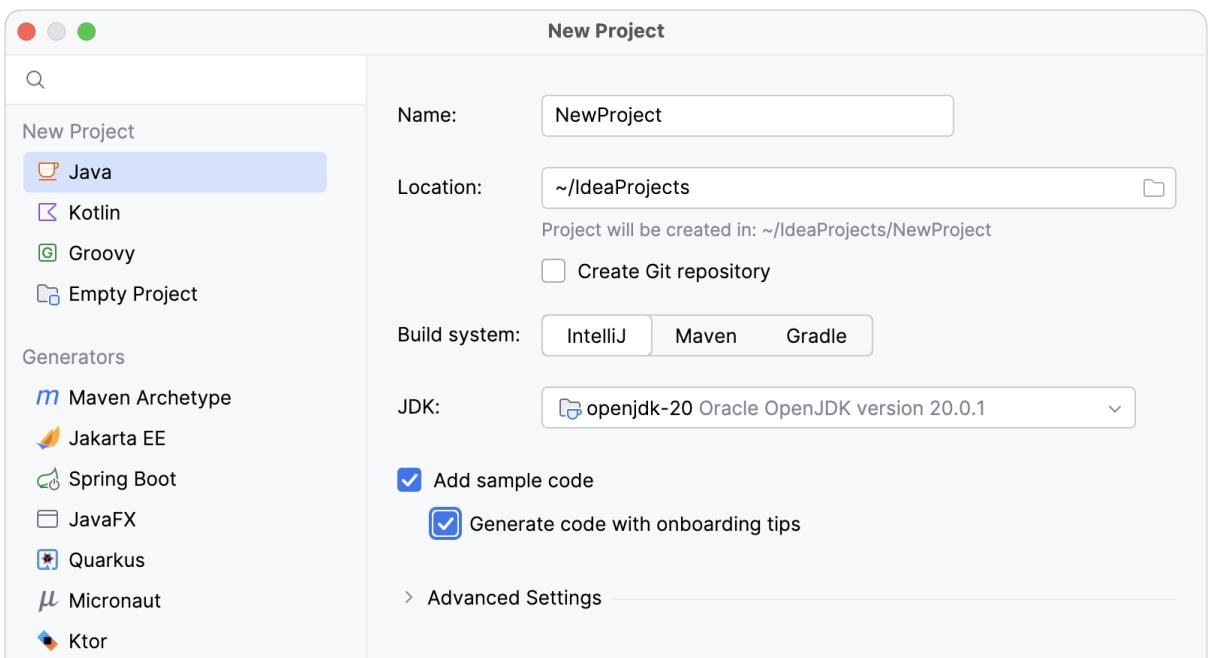
Existem duas maneiras principais de iniciar o assistente de criação de projeto:

- 1.A partir da Tela de Boas-Vindas: Se você acabou de abrir o IntelliJ IDEA e a tela de Boas-Vindas (Welcome Screen) está visível, clique na opção New Project (Novo Projeto).
- 2.A partir do Menu Principal: Se você já tem um projeto aberto, navegue até o menu superior e selecione File (Arquivo) > New (Novo) > Project (Projeto).

1. 2. Configurando as Opções Básicas do Projeto

2. O assistente de "Novo Projeto" será aberto, permitindo que você configure as principais opções do seu projeto.

3.



4. Preencha os seguintes campos:
5. •Name (Nome): Digite um nome para o seu projeto. Este nome será usado para o diretório raiz do projeto.
6. •Location (Localização): Especifique o diretório onde o projeto será salvo. O IntelliJ IDEA sugere um local padrão, mas você pode alterá-lo.
7. •Language (Linguagem): Selecione a linguagem principal do seu projeto (por exemplo, Java, Kotlin, Groovy).
8. •Build System (Sistema de Build): Escolha o sistema de build que deseja usar. As opções mais comuns são:
9. •IntelliJ: O construtor nativo da IDE, ideal para projetos simples ou para quem está começando.
10. •Maven: Um popular sistema de gerenciamento de projetos e build.
11. •Gradle: Outro sistema de build moderno e flexível.

3. Selecionando o Kit de Desenvolvimento (JDK)

A próxima etapa crucial é selecionar o Kit de Desenvolvimento Java (JDK) que será usado para compilar e executar seu código.

- JDK: No campo JDK, selecione a versão do Java que você deseja usar.
- Se o JDK já estiver configurado na IDE, ele aparecerá na lista.
- Se o JDK estiver instalado no seu computador, mas não na IDE, clique em Add JDK e navegue até o diretório de instalação.
- Se você não tiver o JDK instalado, você pode usar a opção Download JDK para que o IntelliJ IDEA baixe e configure automaticamente.

5. Finalizando a Criação do Projeto

Após preencher todas as configurações, clique no botão Create (Criar).

O IntelliJ IDEA irá configurar o ambiente, baixar as dependências (se você escolheu Maven ou Gradle) e abrir a janela principal da IDE com o seu novo projeto pronto para codificação.

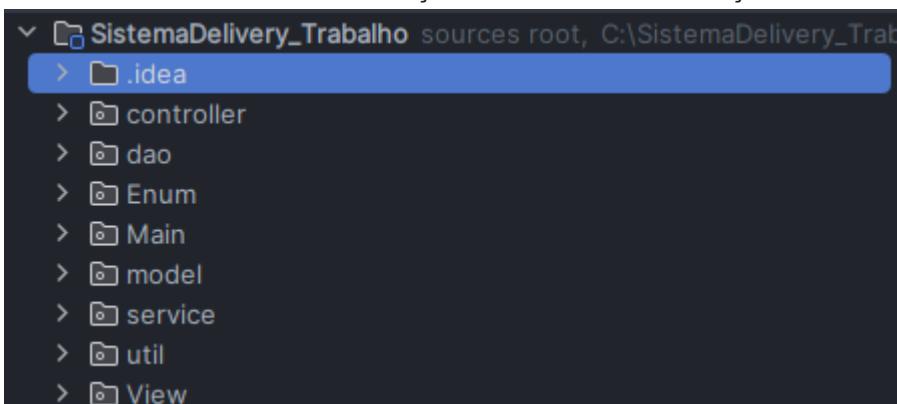
Se você marcou a opção "Add sample code", você verá o arquivo Main.java (ou similar) aberto no editor, contendo o código de exemplo.

Parabéns! Você criou com sucesso seu primeiro projeto no IntelliJ IDEA. Agora você pode começar a escrever seu código.

1. Na pasta **src** crie quatro pastas, **model**, **dao**, **service**, **controller**, **view**. Elas são as **principais pastas para a organização do projeto** em si e responsáveis por guardar os códigos em java. Elas servem,

respectivamente, para:

- a. **Conexão:** Responsável pela conexão com o banco de dados. Normalmente contém classes utilitárias para abrir e fechar conexões JDBC, configurar `DriverManager`, gerenciar `Connection`, `Statement`, etc. Exemplo: `Conexao.java` que retorna uma `Connection` pronta para uso.
- b. **controller:** Controla o fluxo entre a interface gráfica (view) e as regras de negócio (service). Ele recebe as ações do usuário (como clicar em um botão ou preencher um campo) e decide o que o sistema deve fazer em seguida.
- c. **dao:** É a camada de acesso ao banco de dados. Nela ficam as classes responsáveis por inserir, atualizar, excluir e consultar informações no banco MySQL.
- d. **model:** Armazena todas as classes que representam os objetos principais do sistema, ou seja, as entidades.
- e. **service:** Contém as regras de negócio e a lógica do sistema. O service é o intermediário entre o controller e o dao, garantindo que todas as ações sigam as regras corretas antes de acessar o banco.
- f. **view:** Contém todas as interfaces gráficas criadas com JFrame (Swing). É onde o usuário interage diretamente com o sistema, visualizando informações e executando ações.



8.1.2 Criação dos scripts

2. Dentro de **model**, crie os seguintes arquivos `CardapioModel`, `ClienteModel`, `EnderecoModel`, `ItemPedidoModel`, `PagamentoModel`, `PedidoModel`, `RestauranteModel`, `TipoRestaurante`

3. Para as pastas, **dao**, **service** e **controller** vamos fazer a mesma coisa, mas a diferença é que vamos pegar o **nome da classe + nome da pasta.java**. Na pasta dao, vamos criar a classe para a conexão do JDBC, e view vamos deixar por último.
 - a. Para a pasta dao, criaremos um arquivo chamado **ConnectionFactory.java**, ela será a principal classe de conexão com o banco de dados e, também, as classes, CardapioDAO, ClienteDAO, EnderecoDAO, Uma interface GenericDAO, ItemPedidoDAO, PagamentoDAO, PedidoDAO, RestauranteDAO e TipoRestauranteDAO:
4. Faremos o mesmo para a pasta **service**, criando as classes **CardapioService**, **ClienteService**, **EnderecoService**, **ItemPedidoService**, **PagamentoService**, **PedidoService**, **RestauranteService**, **TipoRestauranteService** **UsuariosService**.
5. O mesmo para a pasta **controller**, criaremos as classes **CardapioController**, **ClienteController**, **EnderecoController**, **ItemPedidoController**, **PagamentoController**, **RestauranteController**, **TipoRestauranteController**
6. Para fechamos a criação de classes, só falta **view**, na pasta view temos muitas classes e elas são, **CardapioItemDialog**, **ClienteView**, **ItemPedidoView**, **MenuPrincipalView**, **PedidoView**, **RestauranteAreaView**, **RestauranteLoginView**, **RestauranteView**

e a parte de View clientes, elas são, **ClienteAcompanhamentoView**, **ClienteCardapioView**, **ClienteCarrinhoView**, **ClienteCheckoutView**,

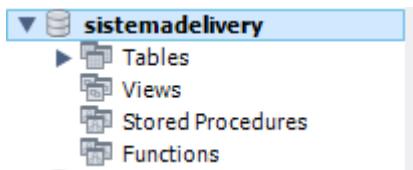
ClienteHistoricoDePedidosView, ClienteListagemRestaurantesView.

8.2 Configuração do MySQL

O objetivo desta seção é fornecer um **guia passo a passo claro e detalhado** para que qualquer outro desenvolvedor consiga preparar o ambiente do banco de dados em sua máquina local.

Crie uma connection, Clique no botão de “+” e coloque o nome da connection de sistemadelivery e clique em ok.

- 1.
2. Dentro da área de script, nós vamos criar o banco e as tabelas para executar scripts individualmente símbolo de raio com um traço branco para fazer esse processo, mas nós só vamos executar o script em manuseio, porquê ainda estamos configurando o projeto.



3. Para criar o banco de dados, escreva **CREATE DATABASE IF NOT EXISTS delivery_db;**. Para rodar essa linha, clique na linha onde o script está e aperte o símbolo do raio mencionado a cima.
4. Depois escreva **USE sistemadelivery;** para criar ou alterar componentes e elementos do banco de dados.
5. Agora, vamos criar as tables. Para criarmos, precisamos utilizar os seguintes comandos:

```
create database sistemaDelivery;  
use sistemaDelivery;
```

```
CREATE TABLE clientes (
```

```
    idCliente INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100),  
    telefone varchar(20),  
    endereco varchar(200)  
);
```

```
CREATE TABLE restaurante (  
    idRestaurante INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100),  
    tipo_cozinha varchar(50),  
    telefone varchar(200)  
);
```

```
CREATE TABLE tipoRestaurante(  
    idTipoRestaurante int auto_increment primary key,  
    nome_tipo varchar(100),  
    descricao varchar(300)  
);
```

```
CREATE TABLE pedido (  
    idPedido INT AUTO_INCREMENT PRIMARY KEY,  
    idCliente int,  
    idRestaurante int,  
    data_hora datetime,  
    valorTotal DECIMAL(10, 2) NOT NULL DEFAULT 0.00,  
    enderecoEntrega VARCHAR(255),  
    status enum('EM_PREPARO', 'A_CAMINHO', 'ENTREGUE',  
    'CANCELADO'),  
    FOREIGN KEY (idCliente) REFERENCES clientes(idCliente),  
    FOREIGN KEY (idRestaurante) REFERENCES  
    restaurante(idRestaurante)  
);
```

```
CREATE TABLE itemPedido (  
    idItem INT AUTO_INCREMENT PRIMARY KEY,  
    idPedido INT,  
    descricao varchar(150),  
    quantidade int,  
    preco decimal(10,2),
```

```
    FOREIGN KEY (idPedido) REFERENCES pedido(idPedido)
);
```

```
create table Pagamento(
    idPagamento int auto_increment primary key,
    idPedido int,
    valor_total decimal(10,2),
    metodo_pagamento enum('PIX', 'DINHEIRO',
    'CARTAO_DE_DEBITO', 'CARTAO_DE_CREDITO'),
    status_pagamento enum('PENDENTE', 'PROCESSANDO',
    'CONCLUIDO', 'CANCELADO'),
    data_pagamento date,
    foreign key(idPedido) references pedido(idPedido)
);
```

```
CREATE TABLE entregador (
    idEntregador INT PRIMARY KEY AUTO_INCREMENT,
    nome VARCHAR(100),
    telefone VARCHAR(20),
    tipo_veiculo VARCHAR(30),
    idRestaurante INT,
    FOREIGN KEY (idRestaurante) REFERENCES
restaurante(idRestaurante)
);
```

```
create table entrega(
    idEntrega int auto_increment primary key,
    idPedido INT NOT NULL,
    idEntregador INT,
    status_entrega enum('PENDENTE', 'PROCESSANDO',
    'EM_TRANSITO', 'ENTREGUE', 'CANCELADO'),
    data_saida DATETIME,
    data_entrega DATETIME,
    tempo_estimado varchar(10),
    observacao VARCHAR(255),
    foreign key(idPedido) references pedido(idPedido),
    foreign key (idEntregador) references entregador(idEntregador)
);
```

```
create table produto(
```

```
idProduto int auto_increment primary key,  
nome varchar(100),  
preco decimal(10,2),  
descricao varchar(200),  
categoria varchar(100),  
idRestaurante int,  
foreign key (idRestaurante) references restaurante(idRestaurante)  
Para a estrutura do código em Java, visite o repositório e veja a  
estrutura e os códigos armazenados lá:  
https://github.com/Lucasvbc-Dev/SistemaDelivery.git
```

9. Execução do Código

Com o banco de dados configurado e o projeto carregado no IntelliJ IDEA, a execução pode ser feita da seguinte forma:

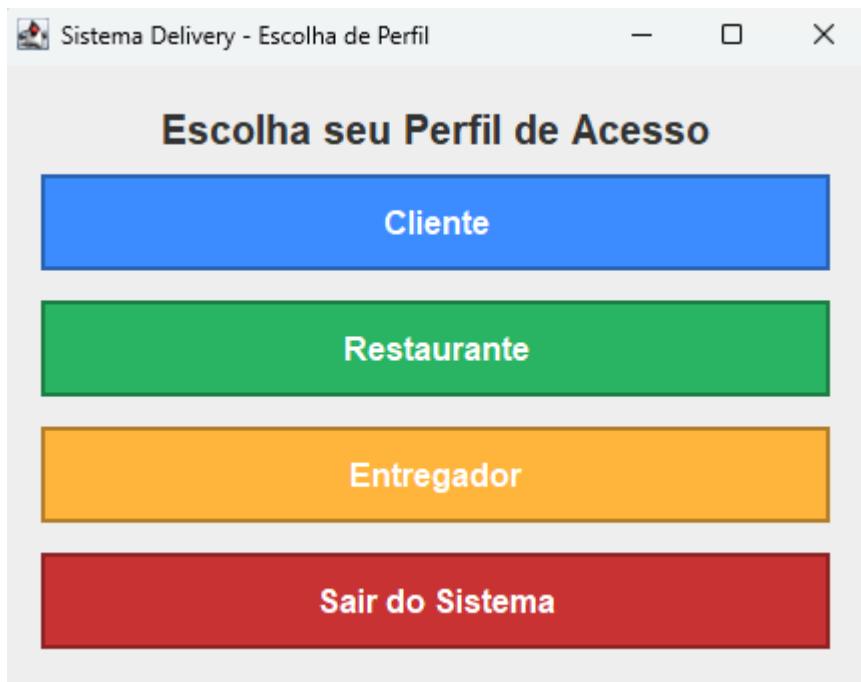
1. Localize a Classe Principal: Encontre a classe que contém o método `public static void main(String[] args)`. Tipicamente, ela terá um ícone verde de "play" (executar) ao lado da declaração da classe ou do método.

2. Executar:

- Clique no ícone verde de "play" e selecione Run '`NomeDaClassePrincipal.main()`'.
- Alternativamente, vá para o menu Run e selecione Run... (atualmente Shift+F10).

O IntelliJ IDEA irá compilar o código e iniciar a aplicação. A saída da execução será exibida na janela Run (Execução) na parte inferior da IDE.

Observação: Se o código for um projeto web (como Spring Boot ou Jakarta EE), a execução iniciará um servidor local (ex: Tomcat), e você poderá acessar a aplicação através de um navegador (geralmente em <http://localhost:8080>).



Tela Inicial



Tela do Cliente



Tela do Restaurante

| ID Entrega | ID Pedido | Status | ID Entregador | Cliente | Endereço |
|------------|-----------|-------------|---------------|---------------|----------|
| 1 | 1 | ENTREGUE | 1 | Cliente ID: 1 | |
| 4 | 8 | EM_TRANSITO | 4 | Cliente ID: 6 | |

[Atualizar Lista \(Todas as Entregas\)](#) [Assumir Entrega \(EM TRÂNSITO\)](#) [Marcar como ENTREGUE](#)

Tela do Entregador

Após seguir esses passos, tanto no back, como no banco e no front, sua máquina estará pronta para rodar nosso sistema de delivery.