

Daniel Ferreira Martins Felix

Manual de Instalação e Uso do Sistema de Clínica Veterinária

Universidade de fortaleza
Curso de Ciência da Computação

Fortaleza – CE

Maio de 2025

Sumário

Sumário	2	
Introdução	3	
1	REQUISITOS DO SISTEMA	4
2	INSTALAÇÃO DOS COMPONENTES	5
2.1	Instalação do IntelliJ IDEA	5
2.2	Instalação do XAMPP e MySQL	7
2.3	Instalação do Git	9
3	CLONADO E EXECUTANDO O PROJETO	12
3.1	Passo a passo para executar o projeto	12
3.2	Execução do Sistema Completo	15
3.3	Operações CRUD via Interface do Frontend	17
3.4	Execução de Operações CRUD via Comandos no Backend	19
3.5	Testando as Rotas com o Insomnia	20
4	DIAGRAMAS DO SISTEMA	21
Diagrama Entidade-Relacionamento (DER)		21
Diagrama de Classes UML		21
5	CONSIDERAÇÕES FINAIS	23
Referências		24

Introdução

Este manual tem como objetivo orientar o processo de instalação e utilização do sistema de gerenciamento para clínicas veterinárias, desenvolvido com Java (Spring Boot), HTML, CSS, JavaScript e MySQL. O sistema permite o cadastro de tutores, animais, produtos, consultas e diagnósticos.

1 Requisitos do Sistema

- Sistema Operacional: Windows 10 ou superior
- IntelliJ IDEA (Community ou Ultimate)
- XAMPP (Apache + MySQL)
- Git
- Navegador web moderno (Google Chrome, Firefox)

2 Instalação dos Componentes

2.1 Instalação do IntelliJ IDEA

1. Acesse: <https://www.jetbrains.com/pt-br/idea/download/?section=windows> ou entre no Google e pesquise por “intellij download”.

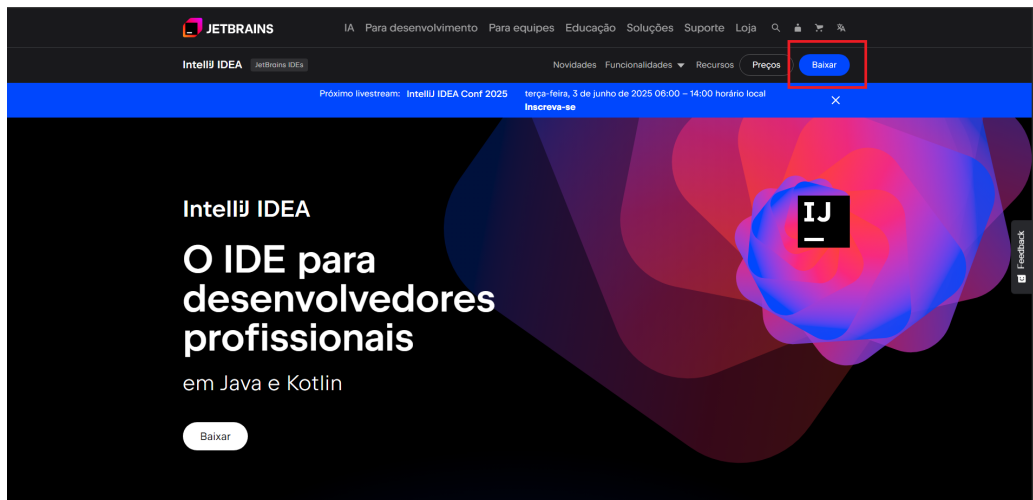


Figura 1 – Página inicial de download do IntelliJ IDEA

2. Nessa parte do site você terá a opção da versão Community (gratuita) ou Ultimate (paga).

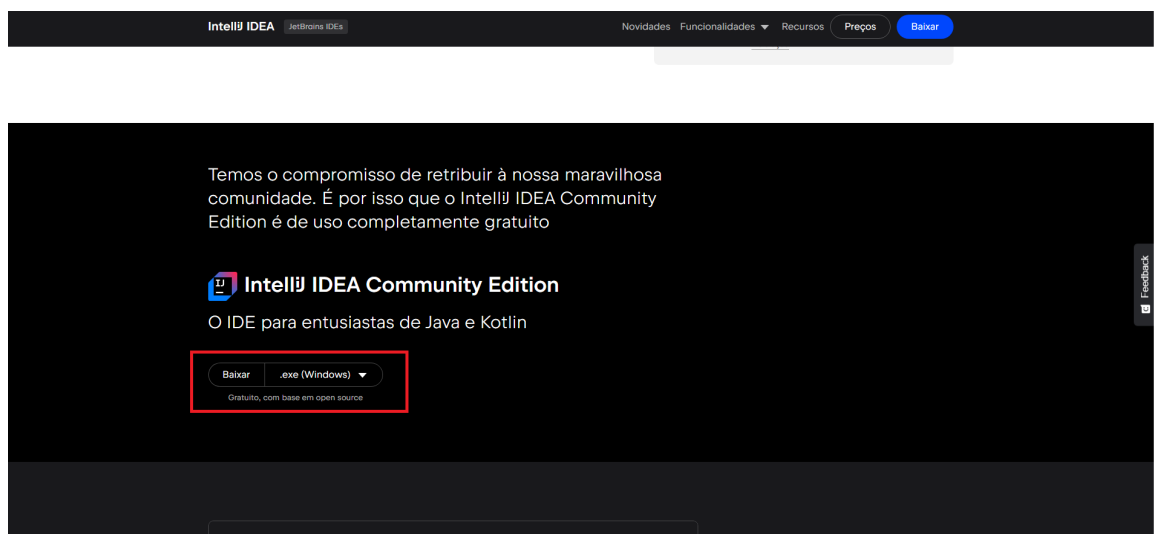


Figura 2 – Botão para baixar o instalador

3. Execute o instalador e siga as instruções. OBS: Se o download não começar sozinho você pode clicar no "link direto" marcado em azul, para encontrar o executável procure no canto superior esquerdo ou pelo atalho ctrl+j

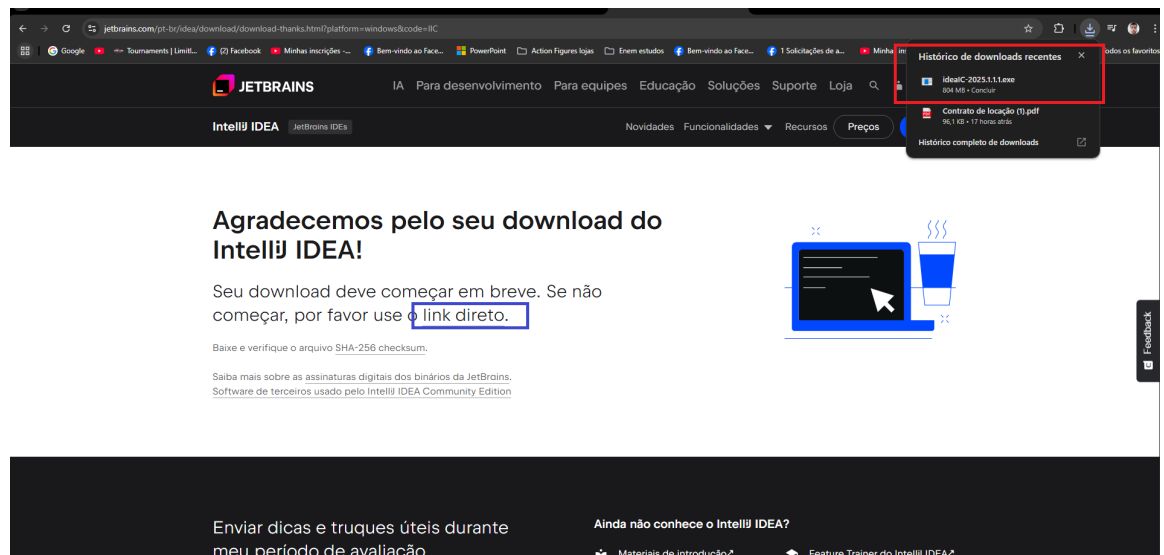


Figura 3 – Caminho de instalação do executável do IntelliJ IDEA

4. Seguir instruções do executável até a abertura da IDE

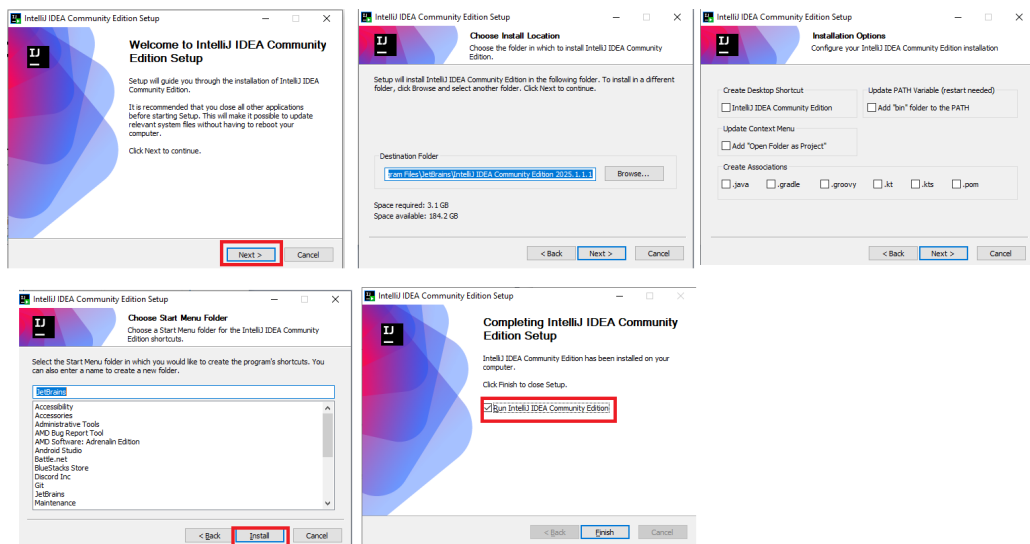


Figura 4 – Tela de execução do IntelliJ após instalação

5. Instalação concluída , essa é a tela que você deve estar enxergando

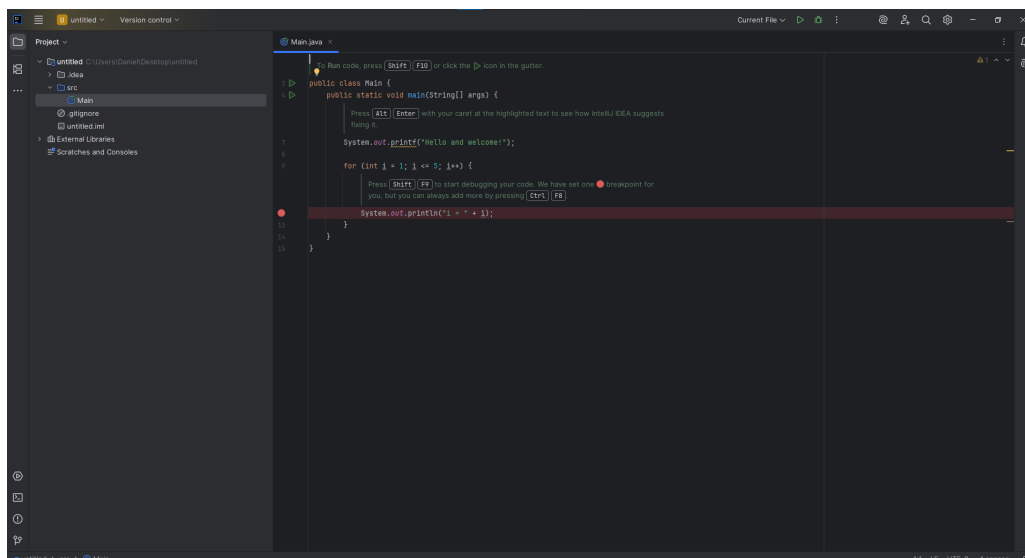


Figura 5

2.2 Instalação do XAMPP e MySQL

1. Acesse: <https://www.apachefriends.org/pt.br/index.html> ou pesquise no Google por “XAMPP Installers and Downloads for Apache Friends”.



Figura 6 – Página oficial do site para download do XAMPP

2. Clique para baixar a versão recomendada para o seu sistema operacional. O instalador começará automaticamente.



Figura 7 – Download do executável do XAMPP

3. Após o download, abra o instalador e siga as etapas padrão de instalação clicando em “Next”.

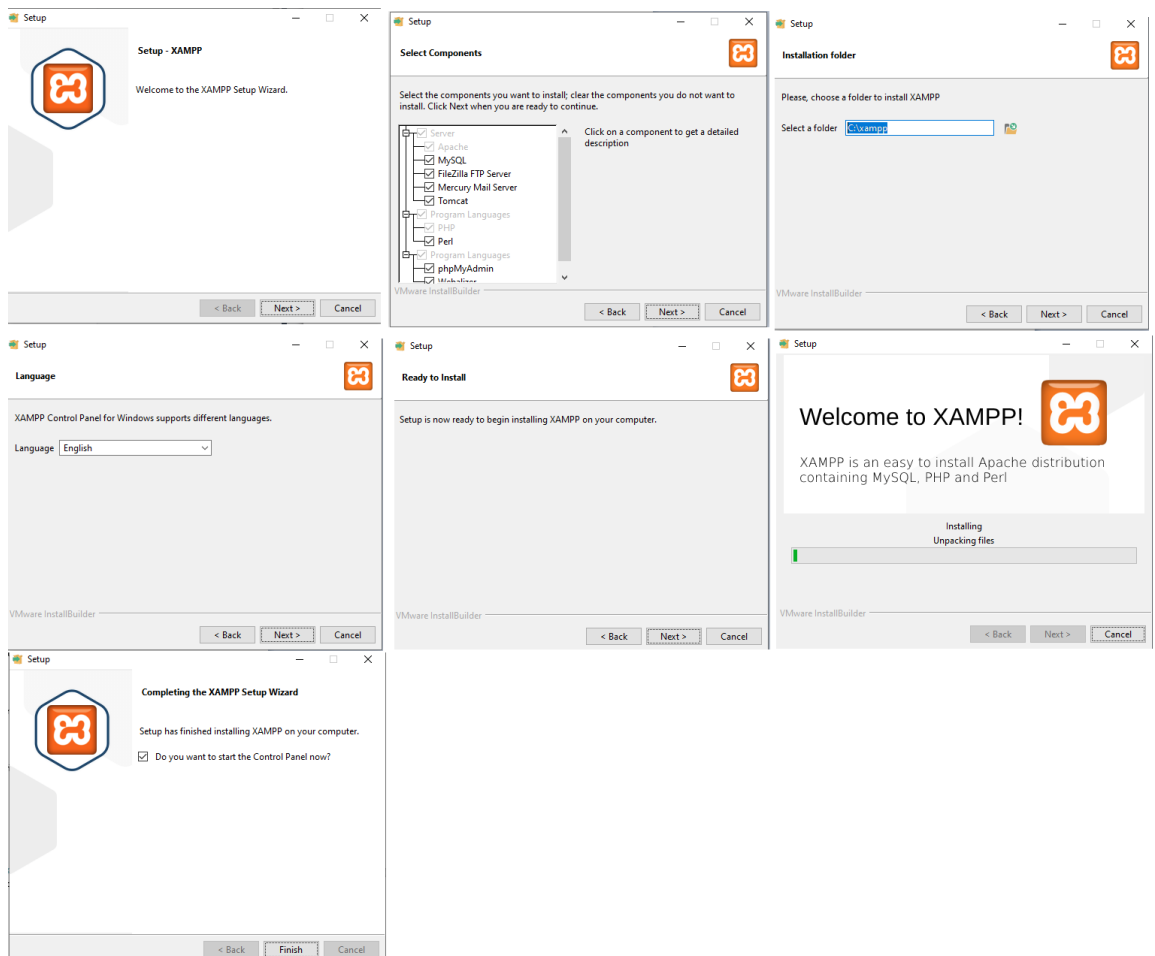


Figura 8 – Caminho de instalação do XAMPP

4. Concluída a instalação, abra o painel de controle do XAMPP. Em seguida, O botão destacado "Start" será usado posteriormente para abrir a porta.

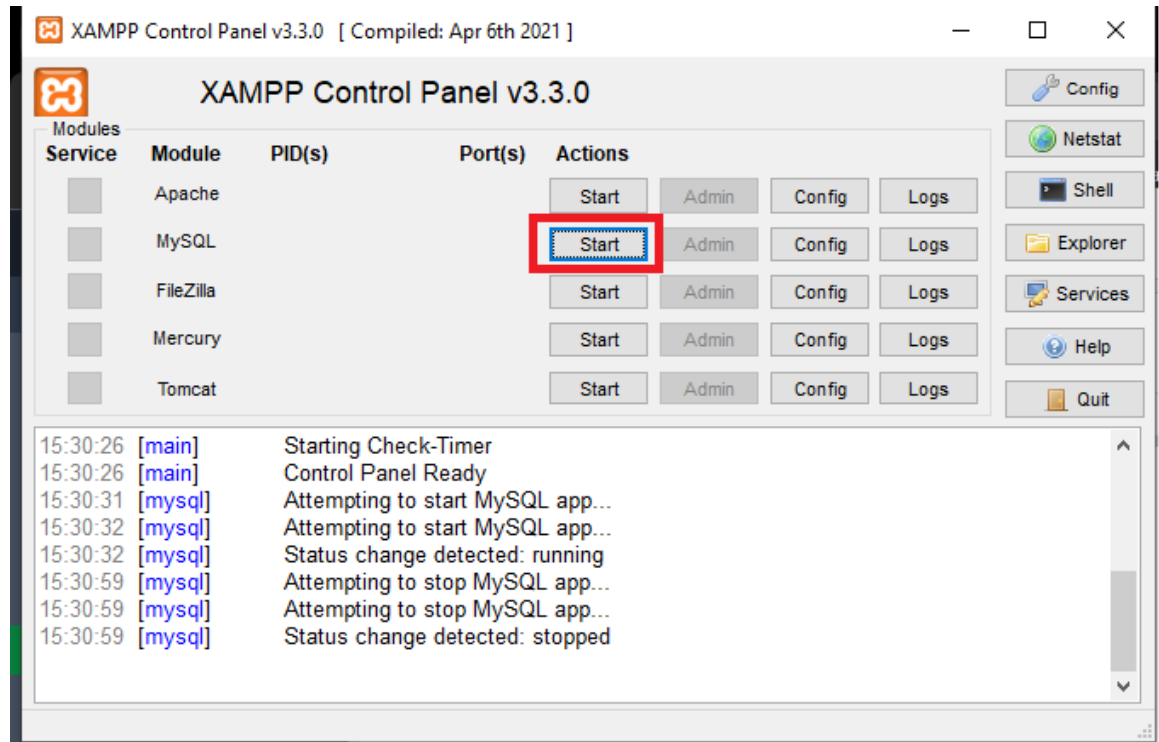


Figura 9 – XAMPP em execução com Apache e MySQL ativos

2.3 Instalação do Git

1. Acesse o site oficial: <https://git-scm.com/download/win> ou pesquise no Google por "Download Git Windows".

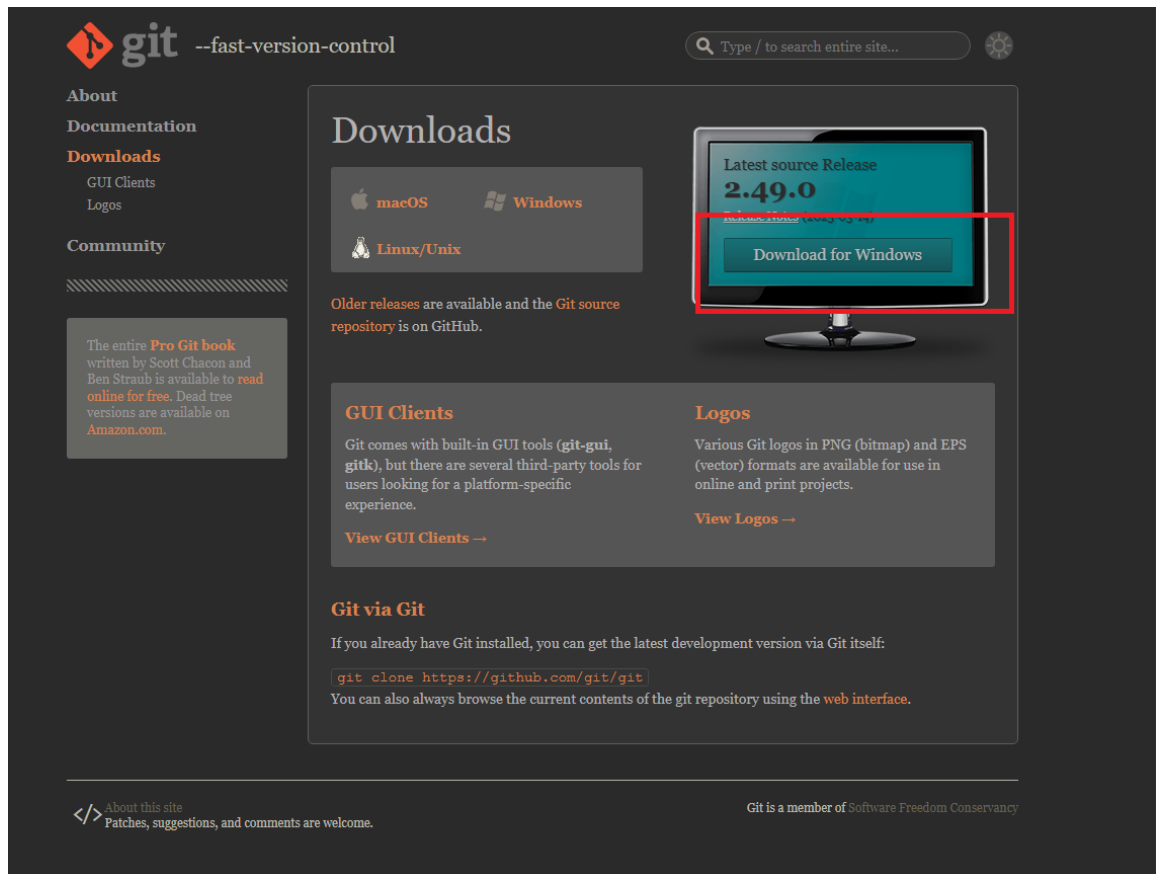


Figura 10 – Página oficial para download do Git

2. Após o download, localize o executável e execute-o para iniciar a instalação.

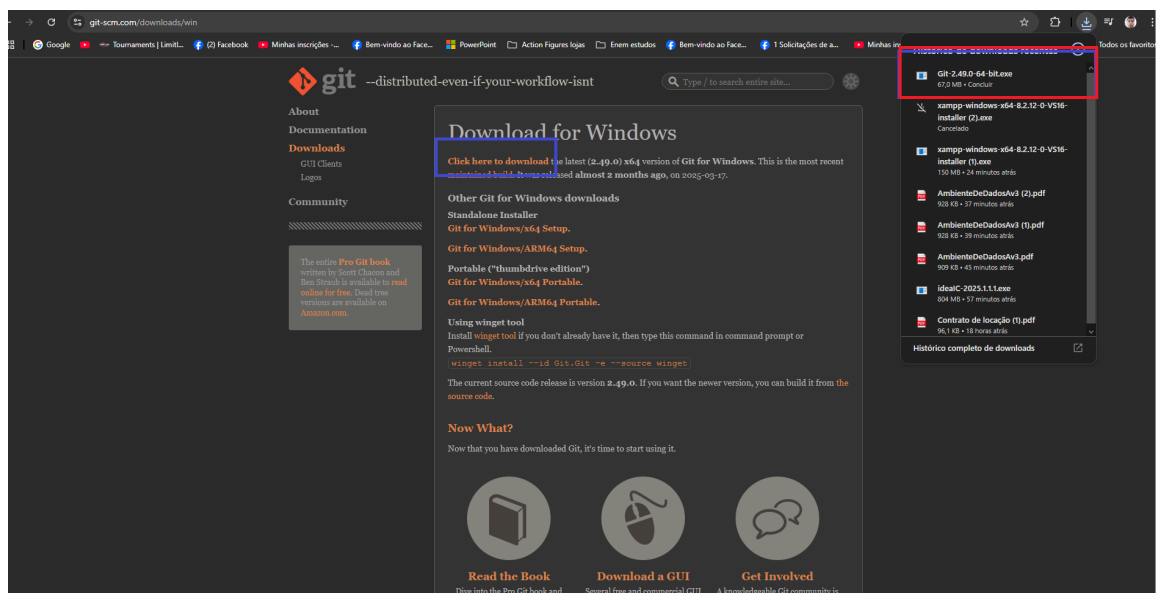


Figura 11 – Executável do Git baixado

3. Siga as etapas do instalador. As opções padrão são suficientes na maioria dos casos, mas certifique-se de deixar marcada a opção que adiciona o Git ao PATH do sistema.

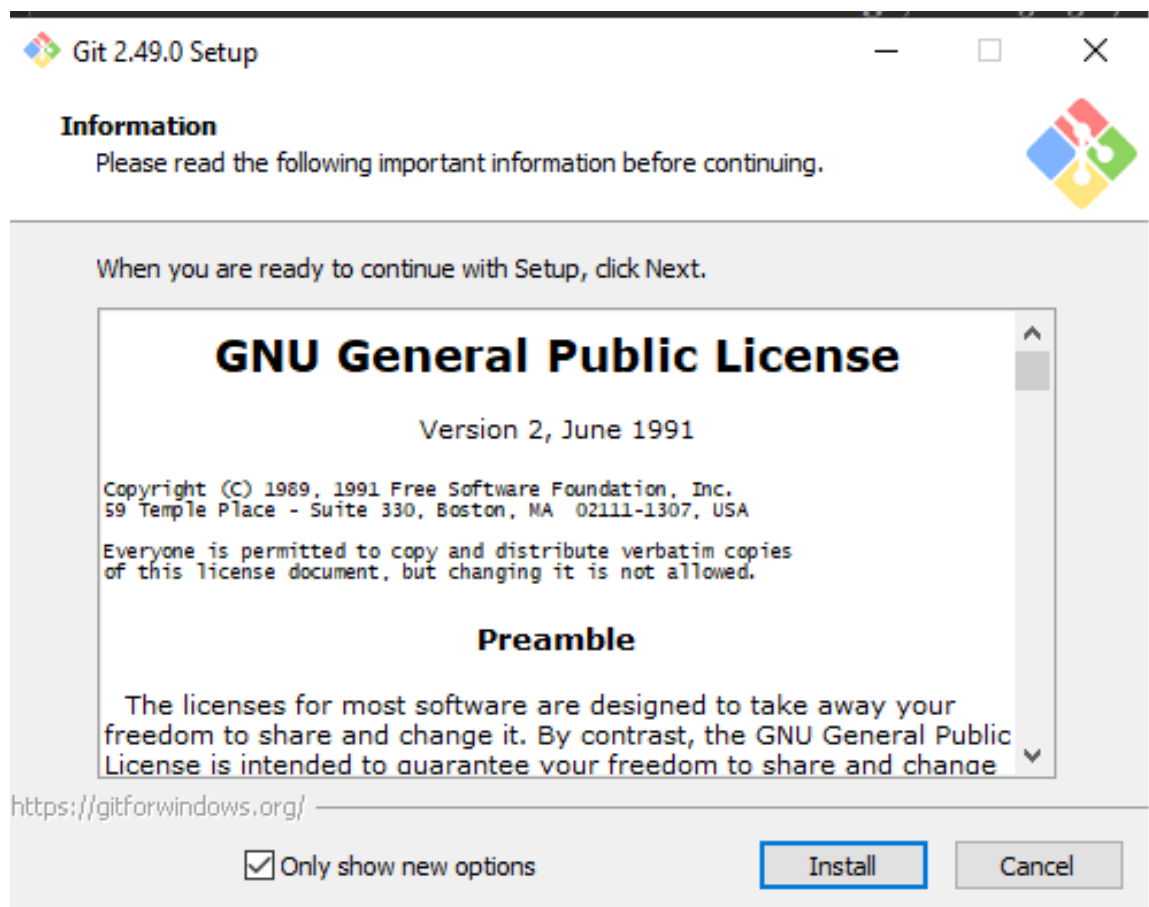


Figura 12 – Processo de instalação do Git

4. Finalize a instalação. Após isso, o Git pode ser acessado tanto pelo Prompt de Comando quanto pelo Git Bash.

Obs.: Para verificar se o Git foi instalado corretamente, digite o comando `git --version` no terminal.

3 Clonado e executando o projeto

3.1 Passo a passo para executar o projeto

1. Certifique-se de que o servidor MySQL está ligado no XAMPP port 3306.

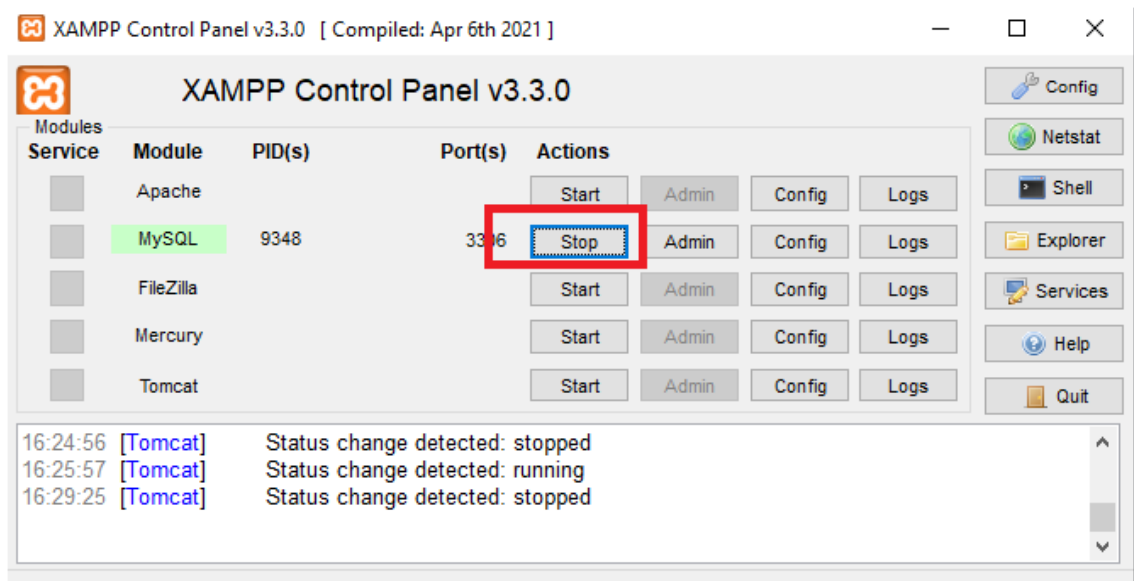


Figura 13 – MySQL ativado no painel do XAMPP

2. Clone o projeto via Git Bash no diretório desejado:

```
git clone https://github.com/seu-usuario/seu-projeto.git
```

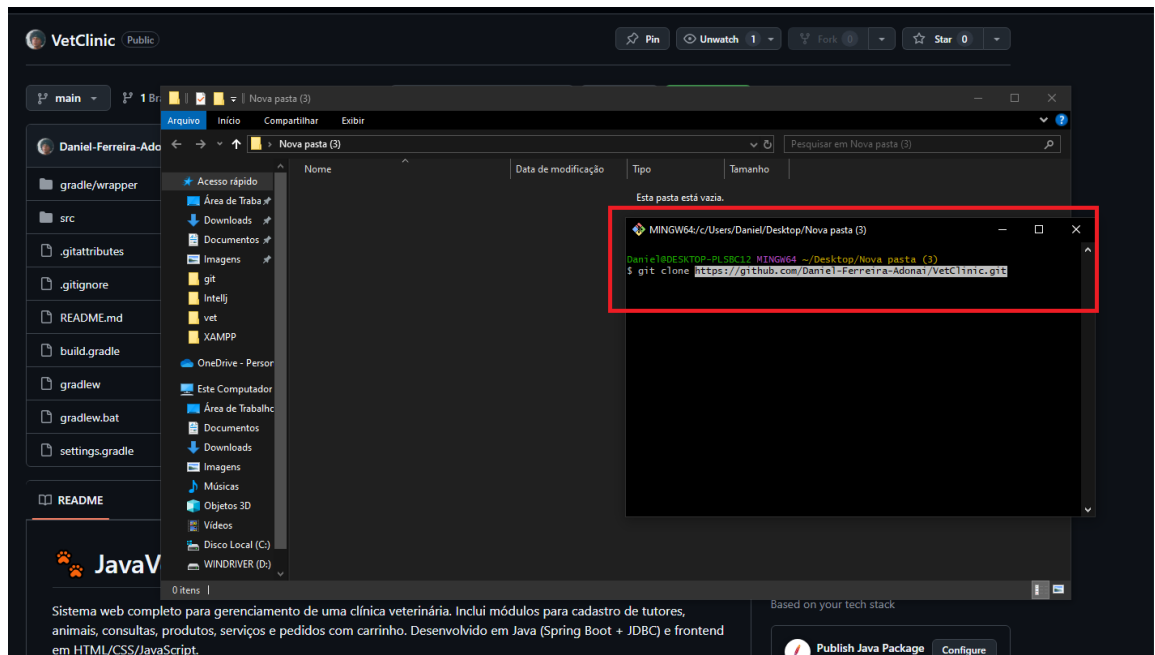


Figura 14 – Clonando o projeto via terminal (Git Bash)

3. Alternativamente, você pode clonar o projeto diretamente pelo IntelliJ IDEA.

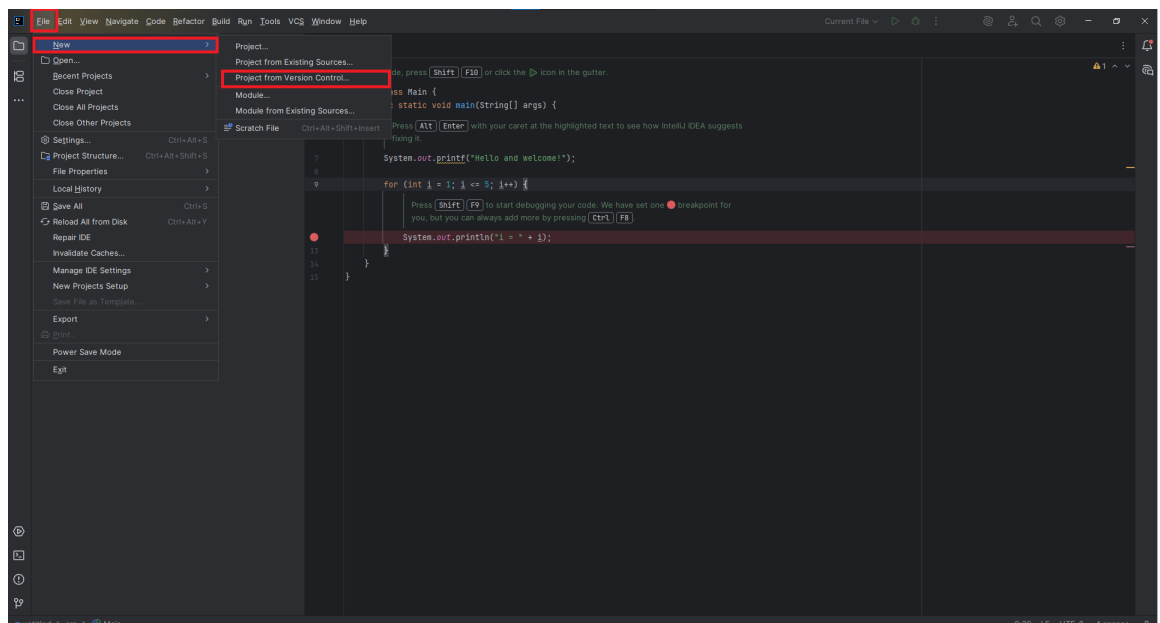


Figura 15 – Clonando via IntelliJ – parte 1 (informar URL do repositório)

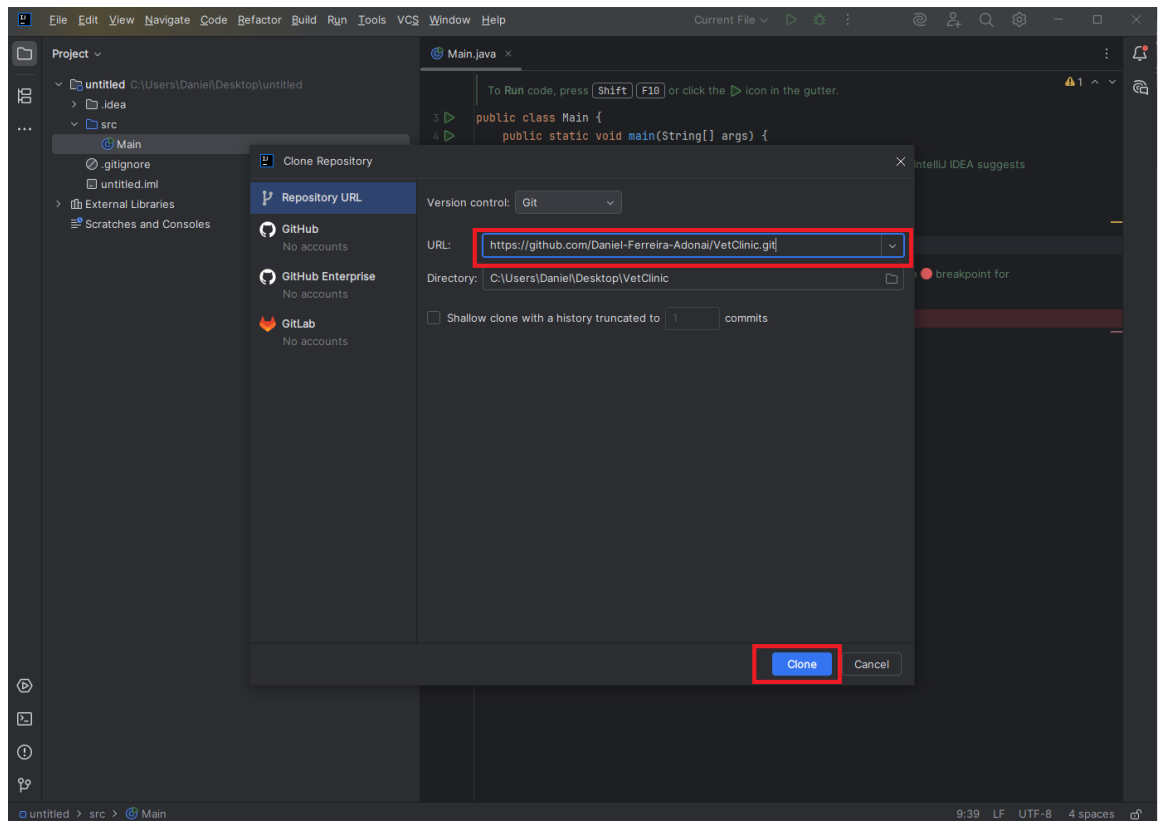


Figura 16 – Clonando via IntelliJ – parte 2 (confirmando diretório)

- Após a clonagem, ao abrir o projeto no IntelliJ, o sistema de build Gradle será executado automaticamente. Esse processo pode demorar alguns minutos na primeira vez, pois depende da conexão com a internet para baixar dependências.

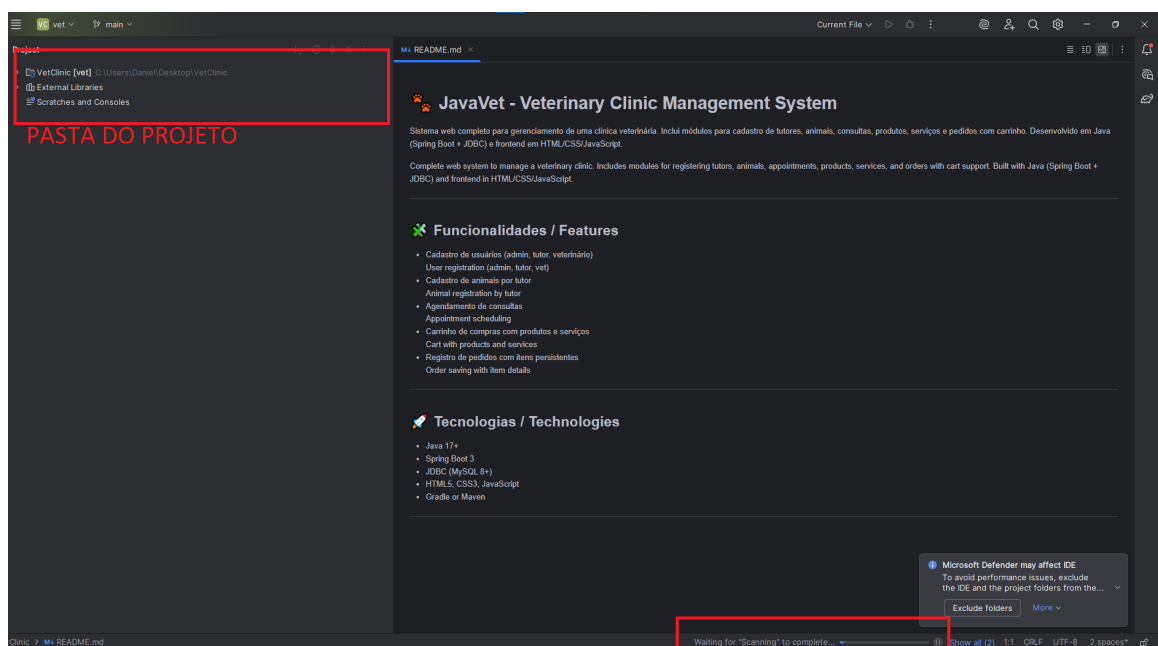


Figura 17 – Projeto clonado e Gradle executando a primeira compilação

3.2 Execução do Sistema Completo

1. Navegue até o diretório ‘src/main/java/com/example/vet/Backend/dao/’ e execute o arquivo ‘DatabaseInitializer.java’. Esse processo cria o banco de dados e todas as tabelas automaticamente. **IMPORTANTE:** esse processo também **insere três usuários de teste** no sistema com as seguintes credenciais:

- **tutor@teste.com** – senha: **tutor123**
- **vet@teste.com** – senha: **vet123**
- **admin@teste.com** – senha: **admin123**

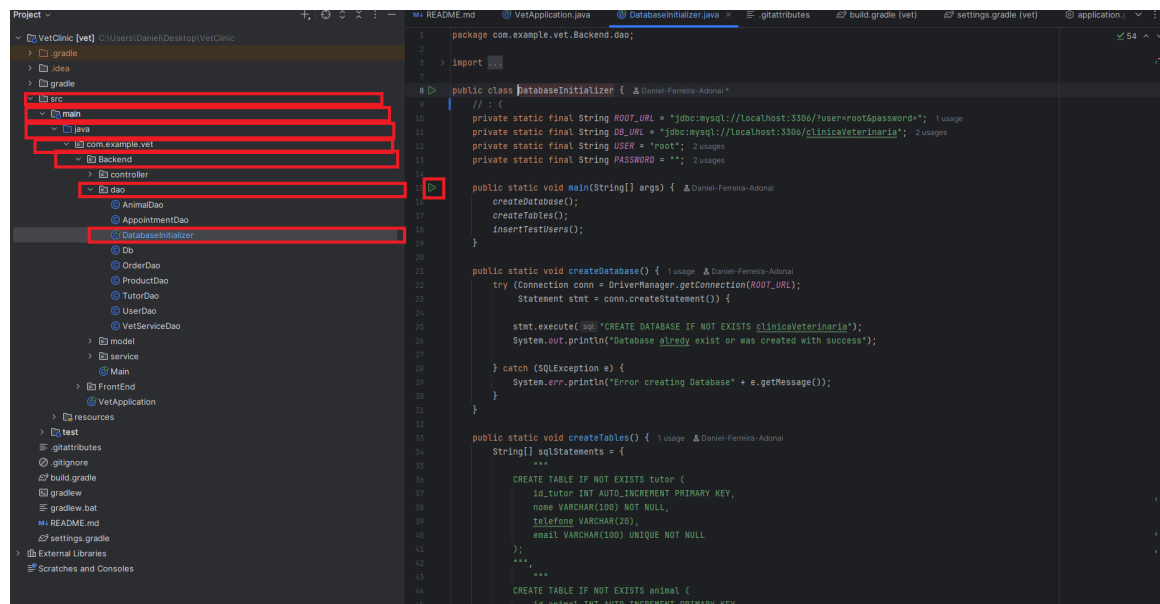


Figura 18 – Acesso ao arquivo DatabaseInitializer para rodar a criação das tabelas

2. Em seguida, execute a classe principal ‘VetApplication’, localizada em ‘com.example.vet’, para iniciar o backend com Spring Boot.

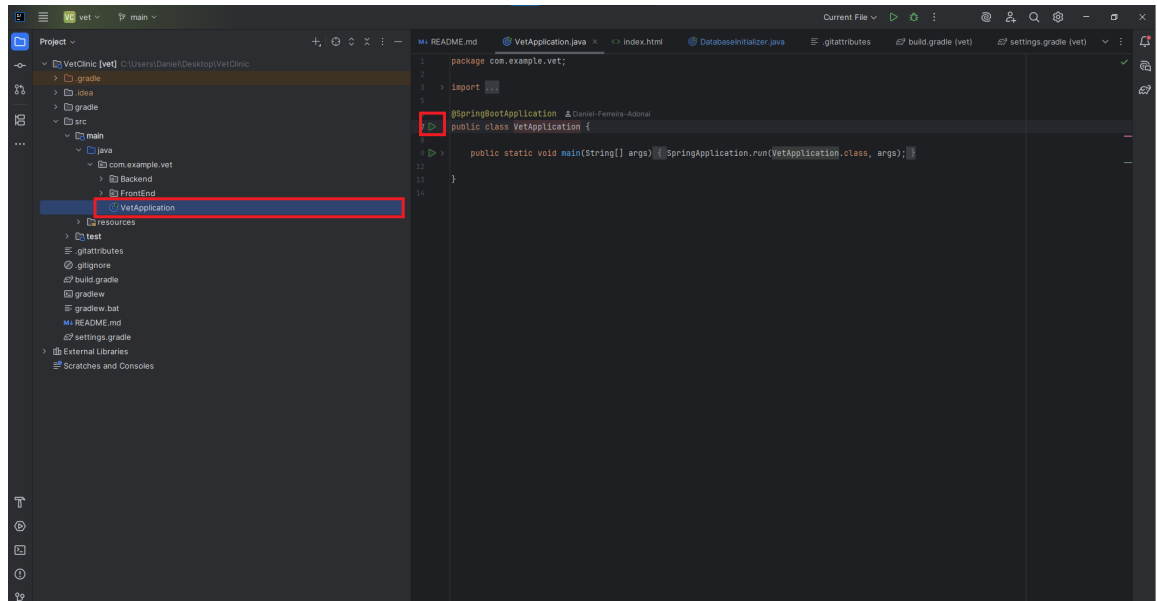


Figura 19 – Inicializando o backend com Spring Boot

- Após a execução, o terminal mostrará a mensagem indicando que o servidor foi iniciado com sucesso na porta '8080'. É essencial **manter essa janela aberta** durante o uso do sistema.

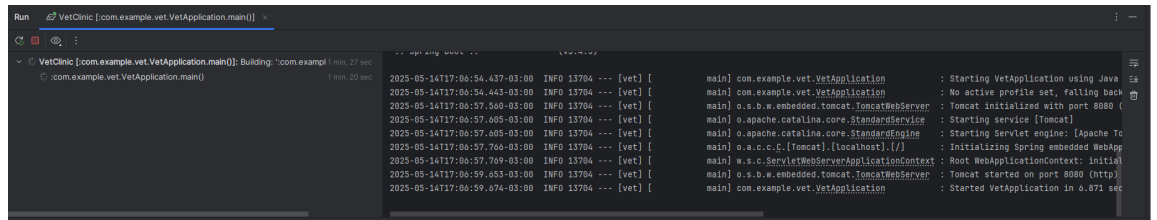


Figura 20 – Terminal confirmando que o backend está rodando

- Navegue até a pasta 'frontend' e abra o arquivo 'index.html' com um navegador da sua preferência (Google Chrome ou Firefox, por exemplo).

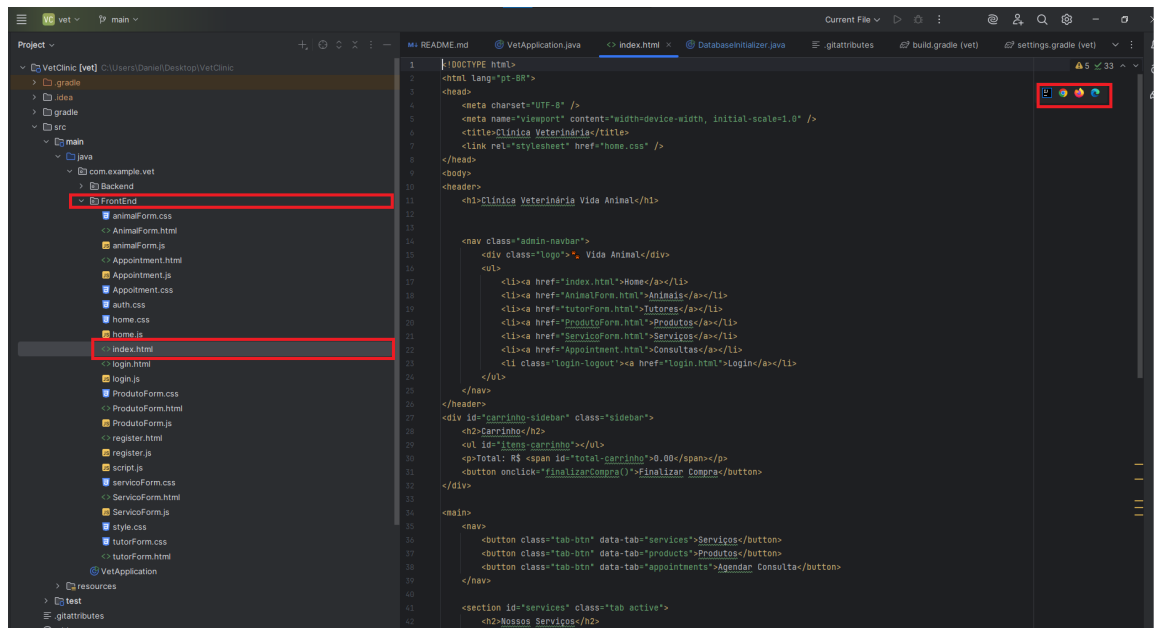


Figura 21 – Abertura do sistema via arquivo index.html

5. A tela inicial do sistema será exibida conforme imagem abaixo:

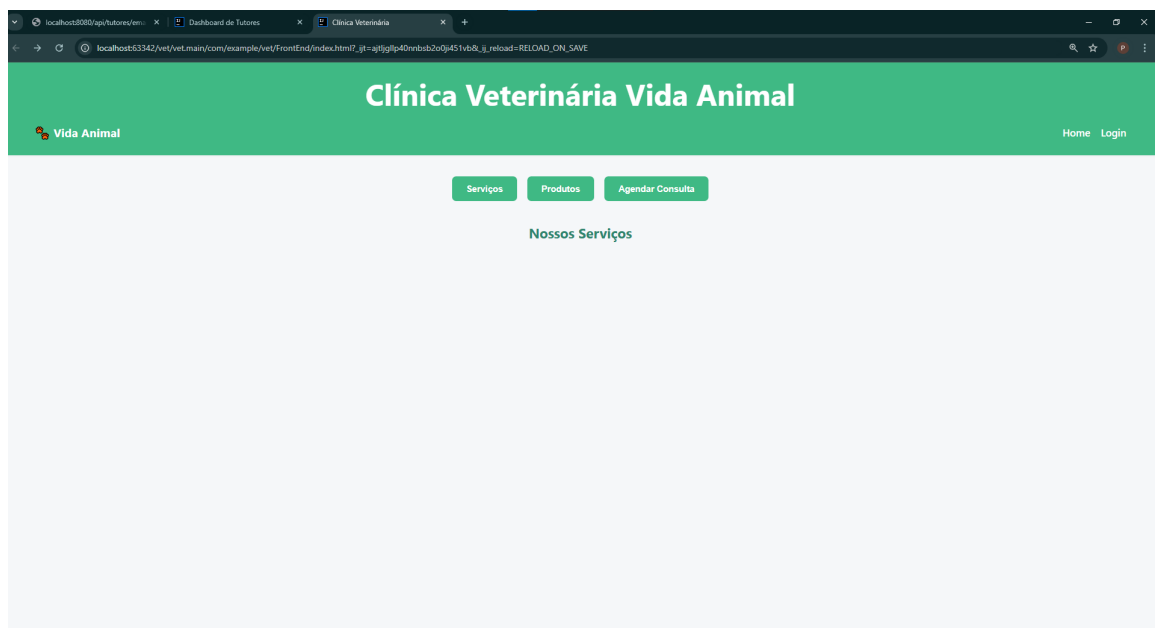


Figura 22 – Tela inicial do frontend da clínica veterinária

3.3 Operações CRUD via Interface do Frontend

1. Na tela inicial do sistema, é possível visualizar o botão **“Login”**, localizado no canto superior direito da interface.

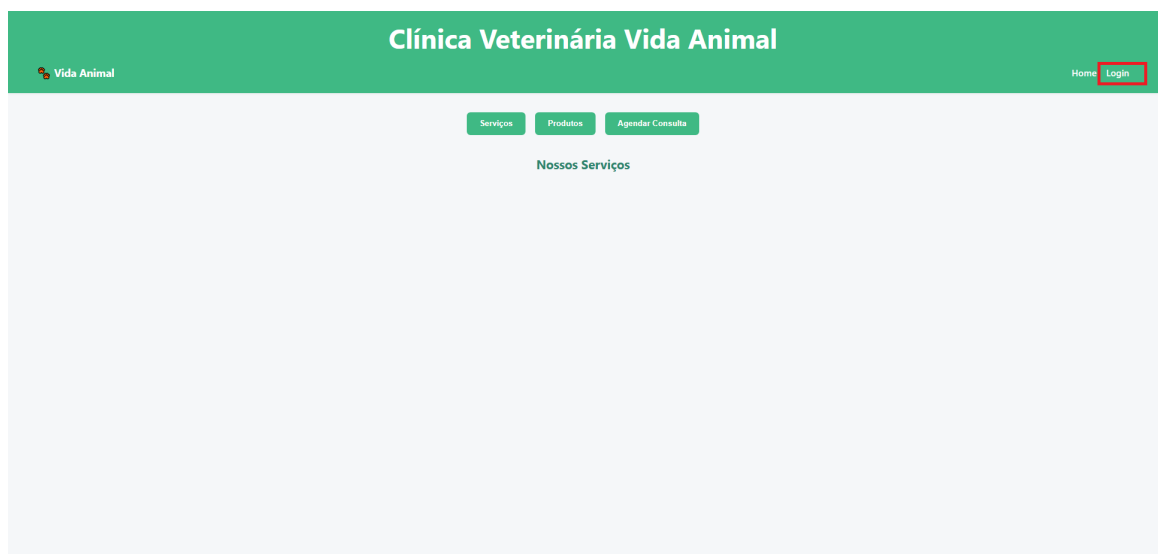


Figura 23 – Tela inicial com o botão de login em destaque

2. A seguir, será exibida a tela de login. Utilize o usuário de teste **tutor@teste.com** com a senha **tutor123** para autenticar como tutor.

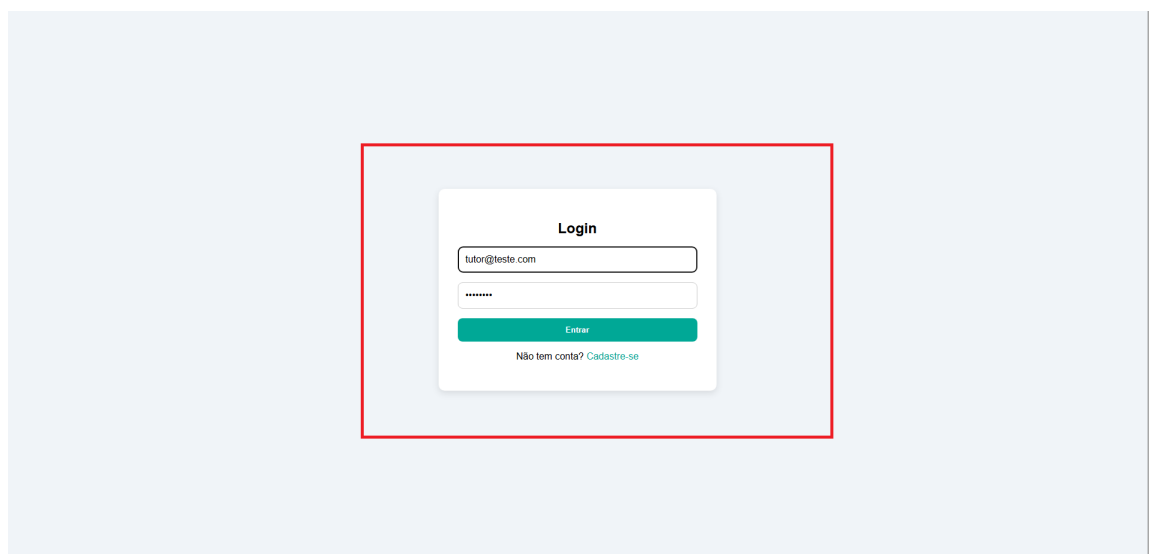


Figura 24 – Tela de login do sistema

3. Após o login, será exibida a tela do CRUD (Create, Read, Update, Delete), com as funcionalidades numeradas em caixas vermelhas para melhor identificação:
 - 1. Adicionar (Create)
 - 2. Listar (Read)
 - 3. Editar (Update)
 - 4. Deletar (Delete)

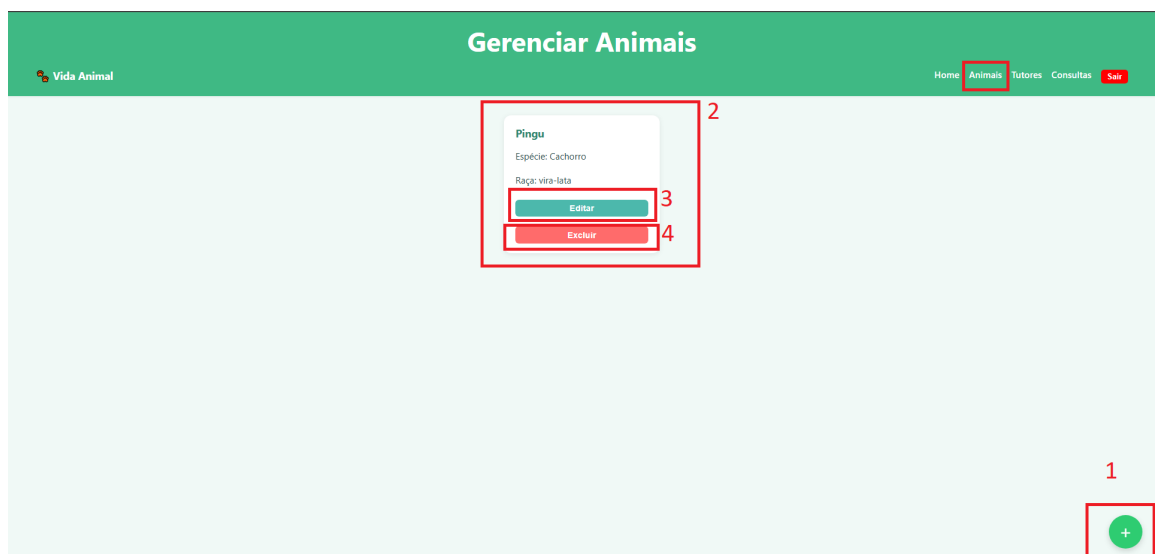


Figura 25 – Tela CRUD com funções destacadas em vermelho

Atenção: apenas usuários do tipo **TUTOR** conseguem cadastrar novos animais no sistema. Caso um usuário **ADMIN** deseje cadastrar um animal, ele deverá na caixa de adição vincular o e-mail de um tutor existente ao animal desejado. Isso garante o relacionamento correto entre tutor e animal conforme a estrutura do banco de dados.

3.4 Execução de Operações CRUD via Comandos no Backend

Além da interface gráfica no frontend, o sistema também permite a execução de operações CRUD diretamente no backend, por meio do código Java. Para isso, é possível acessar o arquivo 'Main.java', localizado em:

`src/main/java/com/example/vet/Backend/Main.java`

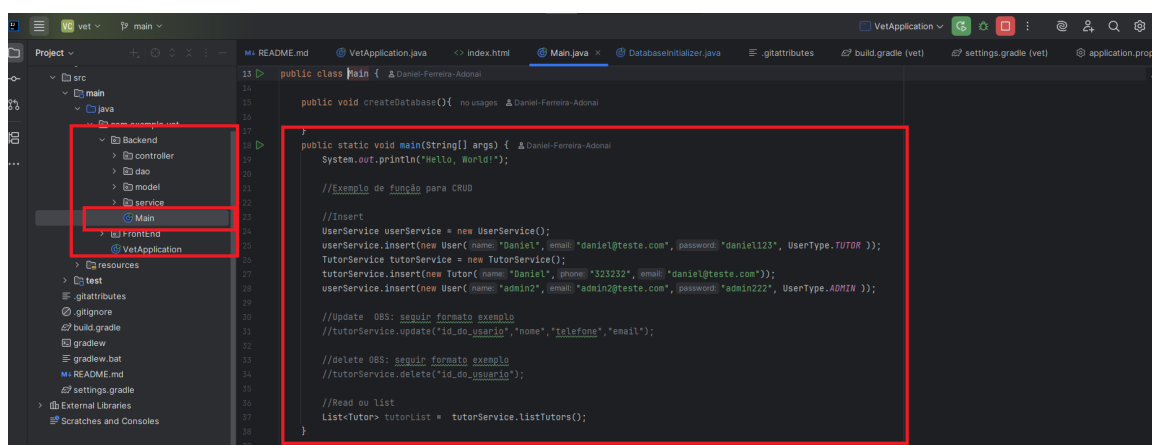


Figura 26 – Caminho até o arquivo Main com exemplos de comandos CRUD

Esse arquivo contém comandos de exemplo utilizando o serviço ‘TutorService’, com métodos representando as principais operações de CRUD:

- **insert** – Insere um novo tutor no banco de dados.
- **list** – Lista todos os tutores cadastrados.
- **Update** – Atualiza os dados de um tutor existente.
- **delete** – Remove um tutor do sistema pelo ID.

Essa abordagem é útil para realizar testes manuais, validar a lógica do backend ou interagir diretamente com o banco sem precisar da interface gráfica.

3.5 Testando as Rotas com o Insomnia

O sistema também pode ser testado utilizando ferramentas de requisição HTTP como o **Insomnia**, que permite enviar requisições GET, POST, PUT e DELETE para a API do backend de forma simples e visual.

Passo 1: Abra o Insomnia ou outro cliente REST (como Postman).

Passo 2: Crie uma nova requisição do tipo GET.

Passo 3: Insira a seguinte URL para listar todos os tutores cadastrados no sistema:

`http://localhost:8080/api/tutores`

Passo 4: Clique em “Send” para executar a requisição. Se o backend estiver rodando corretamente, você verá a resposta em formato JSON, contendo a lista de tutores.

Outros exemplos que podem ser testados via Insomnia:

- POST em `/api/tutores` – Para inserir um novo tutor.
- PUT em `/api/tutores` – Para atualizar os dados de um tutor.
- DELETE em `/api/tutores/{id}` – Para excluir um tutor pelo ID.
- GET em `/api/tutores/email/{email}` – Para buscar um tutor por e-mail.

OBS essa rota também podem ser testadas diretamente pelo navegador em caso de requisições do tipo GET, como:

`http://localhost:8080/api/tutores`

4 Diagramas do Sistema

Para garantir uma visão estruturada e clara da arquitetura do projeto, foram elaborados dois tipos de diagramas fundamentais: o **Diagrama Entidade-Relacionamento (DER)** e o **Diagrama de Classes UML**. Ambos ajudam na compreensão do funcionamento interno do sistema e na organização de seus componentes.

Diagrama Entidade-Relacionamento (DER)

O diagrama DER representa as entidades do banco de dados relacional, seus atributos e os relacionamentos entre elas. É a base para a criação do banco de dados relacional ‘clinicaVeterinaria’.

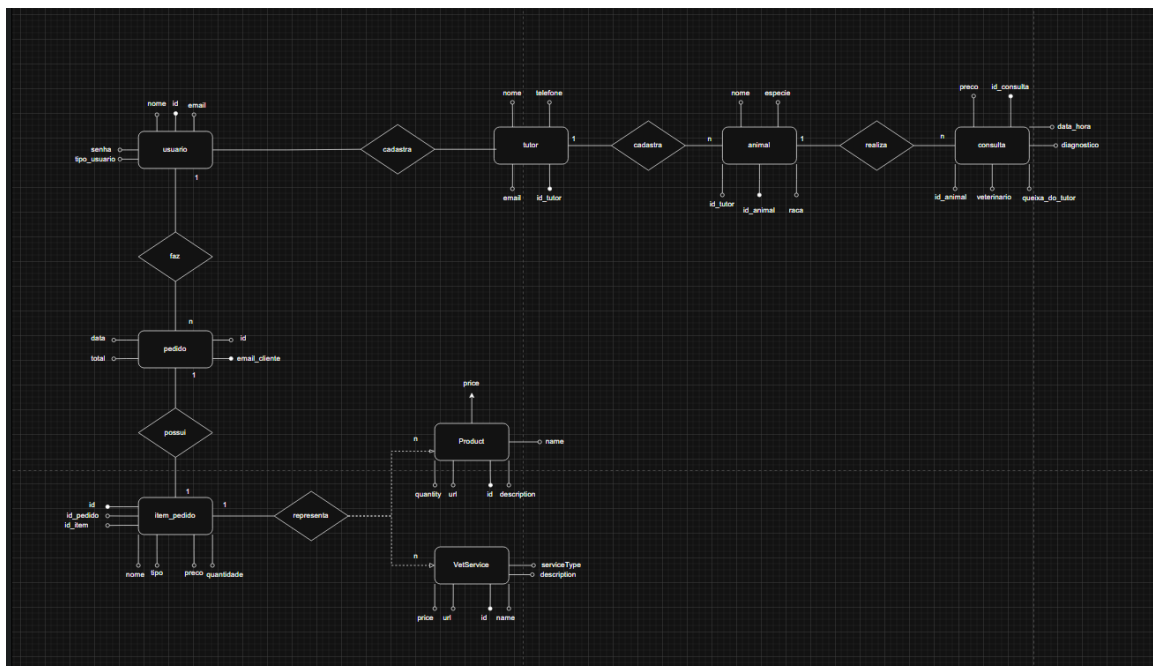


Figura 27 – Diagrama Entidade-Relacionamento do sistema

Diagrama de Classes UML

O diagrama UML apresenta a estrutura das classes da aplicação, incluindo atributos e relacionamentos (como associação e herança). Ele foi desenvolvido com base nos princípios da orientação a objetos e utilizado como referência para a implementação no backend com Java.

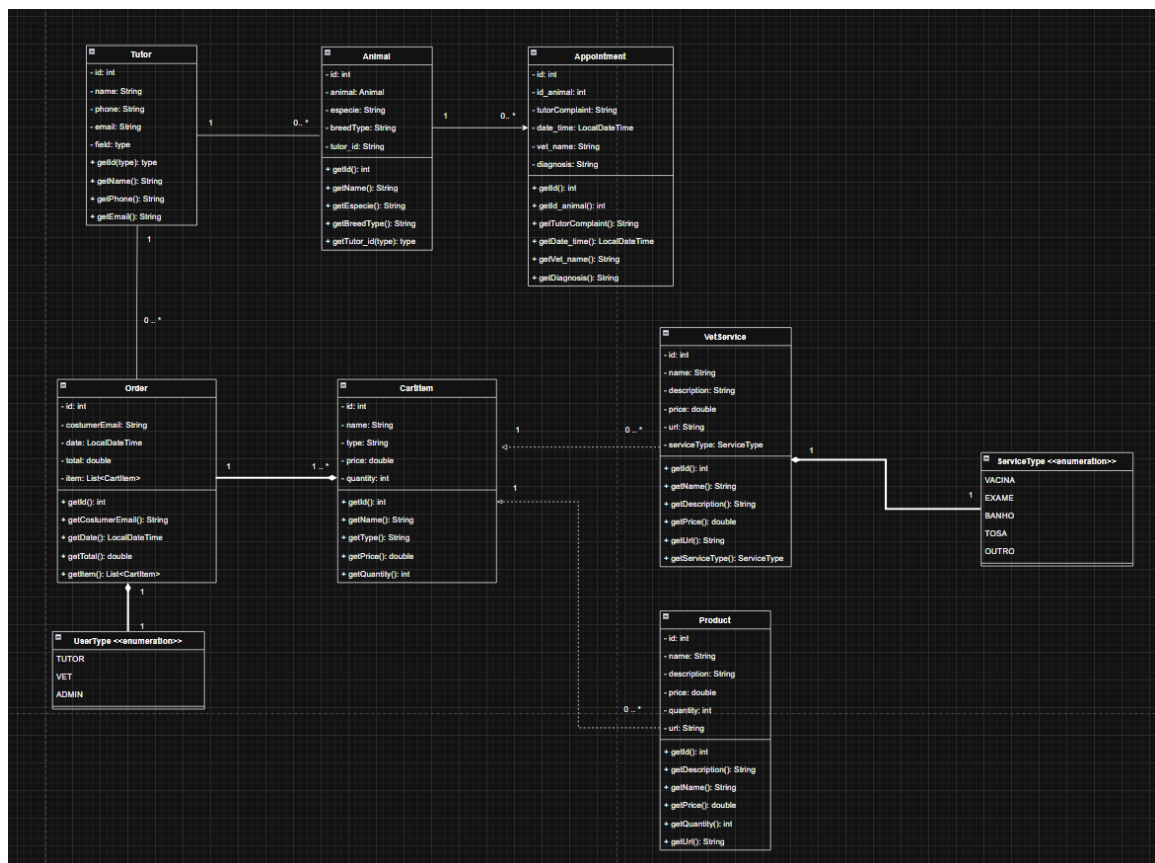


Figura 28 – Diagrama de Classes UML da aplicação

5 Considerações Finais

O projeto finalizado, qualquer duvida, email:d.calpi100@edu.unifor.br ou 85 987751591

.

Referências

- <https://www.jetbrains.com/idea/>
- <https://www.apachefriends.org/>
- <https://spring.io/projects/spring-boot>