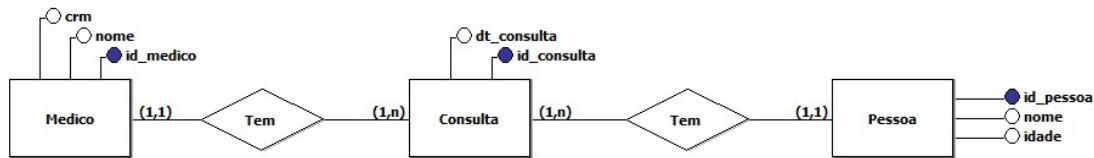


## Projeto da Atividade Online 01

Aluno: Rafael Muzzi da Silva

Tema do Projeto: Cenário Médico

Imagem do DER Conceitual:

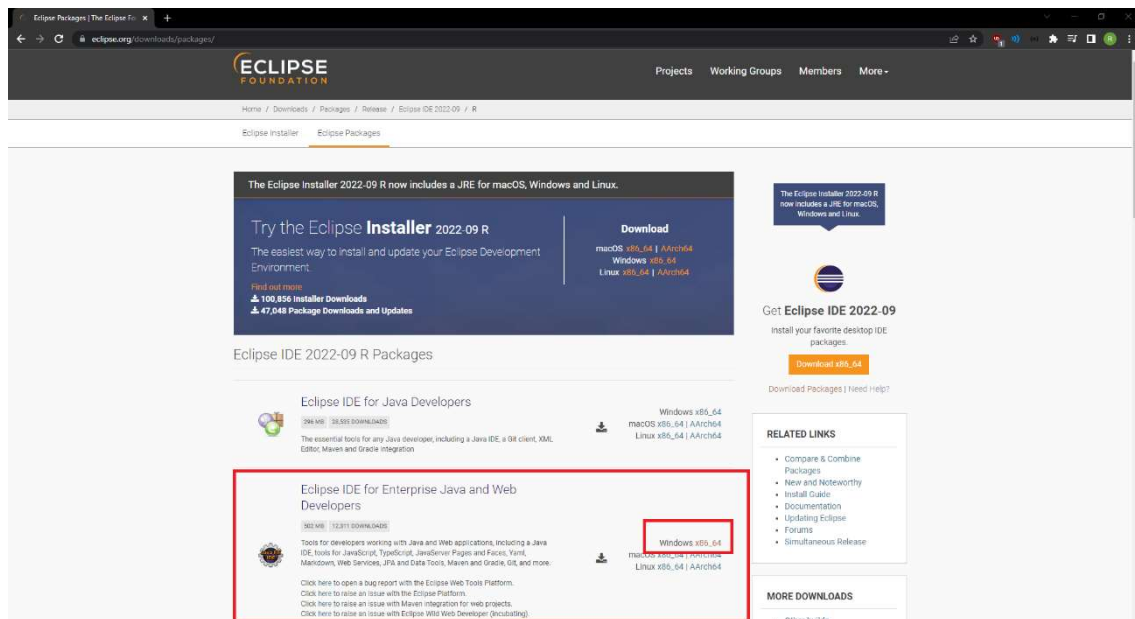


Nesse cenário, é possível guardar informações do paciente (Pessoa), do Médico (Medico) e da consulta do paciente (Consulta).

### Arquivo a ser Baixado:

1º - Baixar Eclipse IDE for Enterprise Java and Web Developers

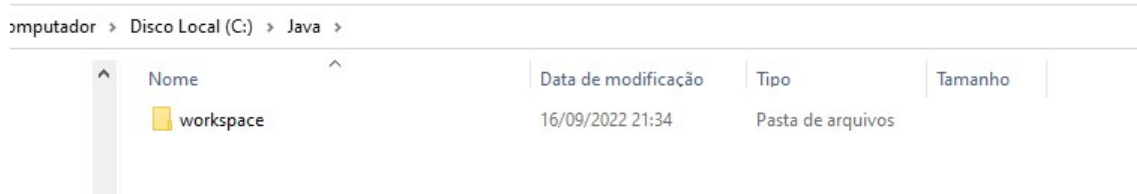
Link: <https://www.eclipse.org/downloads/packages/>



### Configuração do Projeto:

O Eclipse pode ser descompactado em qualquer pasta, por exemplo em: "C:"

-Para configuração do projeto será criada uma pasta do Projeto chamada "Java" e uma sub-pastas, a pasta chamada workspace que conterá o projeto Java.



## Iniciando a IDE Eclipse

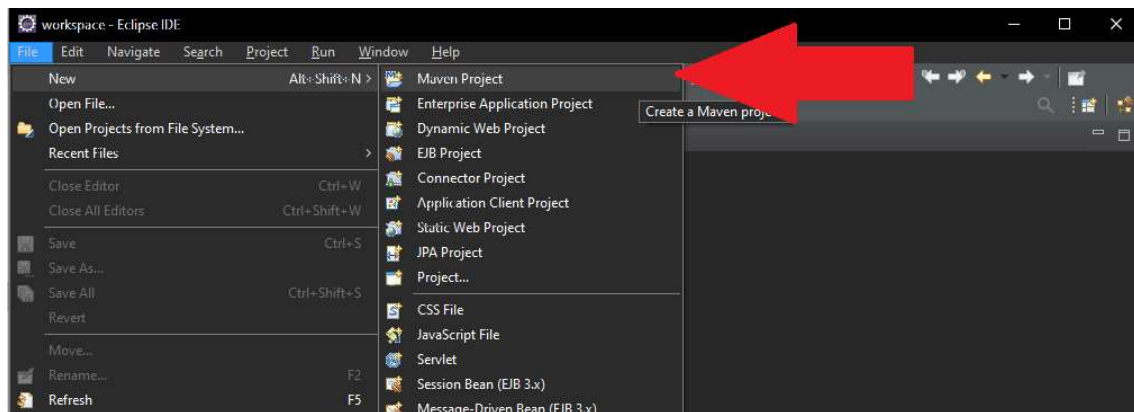
Para iniciar a IDE Eclipse basta executar o arquivo eclipse.exe dentro da pasta do Eclipse.

Será necessário indicar um caminho para a workspace de projetos do Eclipse, indicar o seguinte caminho:

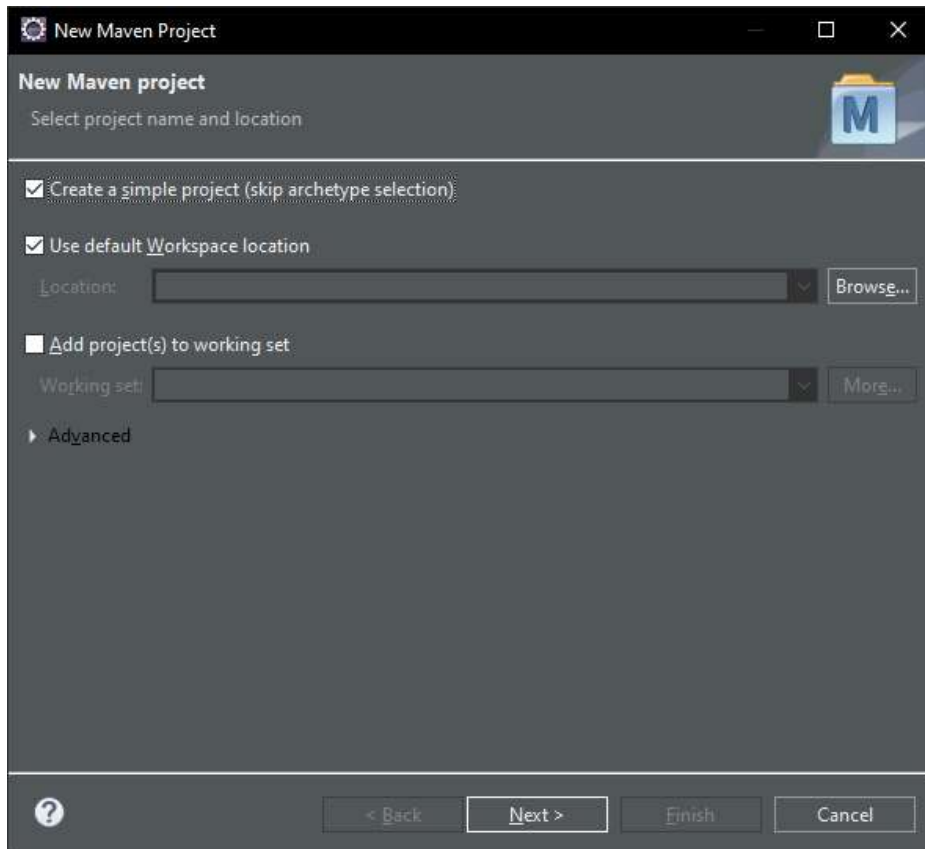
C:/Java/workspace

## Criando o Projeto

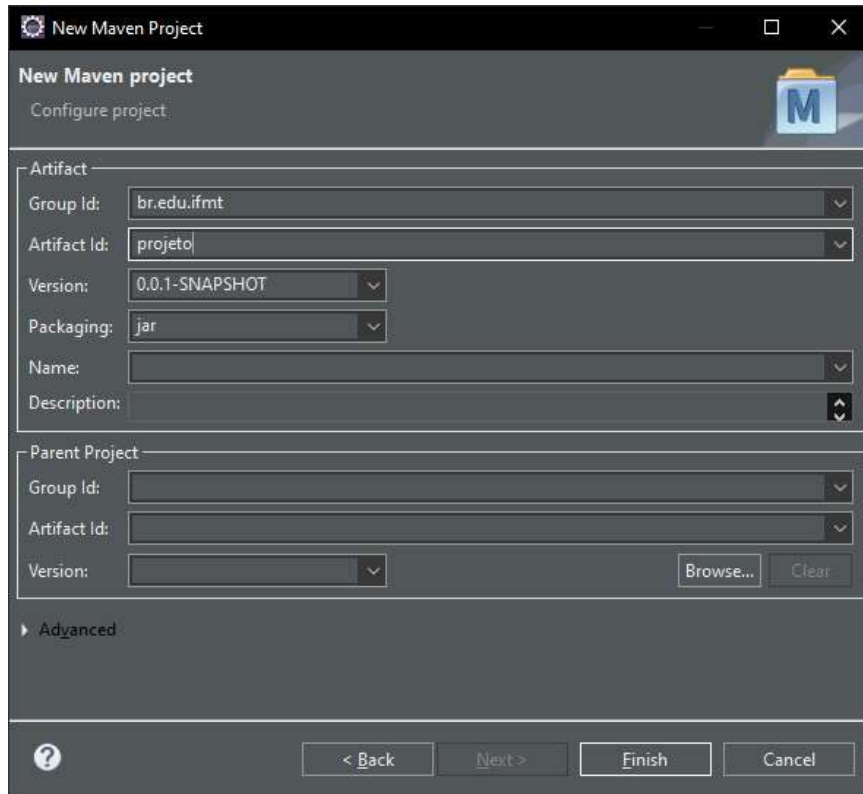
Para criar um Projeto, clique em File -> New -> Maven Project:



Após isso, clique em “Create a simple project (skip archetype selection)” e depois em “Next”:



Em Group Id, coloque “br.edu.ifmt” e em “Artifact Id” coloque “projeto”, depois clique em “Finish” e aguarde o projeto ser criado:



The screenshot shows the 'New Maven Project' dialog box. The 'Artifact' section contains the following fields: Group Id (br.edu.ifmt), Artifact Id (projeto), Version (0.0.1-SNAPSHOT), and Packaging (jar). The 'Parent Project' section is empty. The 'Advanced' section is collapsed. At the bottom, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

#### Adicionando dependências no Pom.xml

As seguintes dependências serão adicionadas, que habilitam o uso da ORM hibernate, do entitymanager e do driver para conexão com o banco de dados:

```
<dependencies>
  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>5.4.12.Final</version>
  </dependency>
  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-entitymanager</artifactId>
    <version>5.4.12.Final</version>
  </dependency>
  <dependency>
    <groupId>org.postgresql</groupId>
    <artifactId>postgresql</artifactId>
    <version>42.5.0</version>
  </dependency>
</dependencies>
```

## Criando a pasta META-INF

Dentro dessa pasta é criada o arquivo persistence.xml que contém configurações de acesso ao banco de dados:



Cujo código é:

```
1<?xml version="1.0" encoding="UTF-8" ?>
2<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
3             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4             xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
5                                 http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd"
6             version="2.1">
7
8     <persistence-unit name="persistenceUnitName" transaction-type="RESOURCE_LOCAL">
9
10         <properties>
11             <property name="javax.persistence.jdbc.driver" value="org.postgresql.Driver" />
12             <property name="javax.persistence.jdbc.url" value="jdbc:postgresql://localhost:5432/cenario_medico" />
13             <property name="javax.persistence.jdbc.user" value="postgres" />
14             <property name="javax.persistence.jdbc.password" value="123456" />
15
16             <property name="hibernate.dialect" value="org.hibernate.dialect.PostgreSQL95Dialect"/>
17             <property name="hibernate.hbm2ddl.auto" value="update" />
18
19             <property name="hibernate.show_sql" value="true" /> <!-- Show SQL in console -->
20             <property name="hibernate.format_sql" value="true" /> <!-- Show SQL formatted -->
21         </properties>
22     </persistence-unit>
23 </persistence>
```

Esse XML possui informações de configuração como o nome da unidade de persistência que é passada ao método de criar o entityManager.

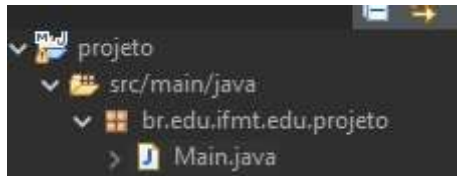
O endereço do banco de dados junto ao nome do banco de dados, nesse caso será o "cenário\_medico".

O nome do usuário do banco de dados e a senha.

Informações para mostrar os códigos no console da aplicação.

### Criando os pacotes do Projeto

Criando o pacote principal “br.edu.ifmt.projeto” que contém a classe Main.java:

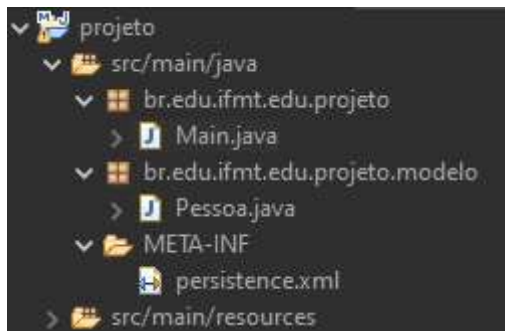


Cujo inicialmente código é:

```
1 package br.edu.ifmt.edu.projeto;
2
3 public class Main {
4
5     public static void main(String[] args) {
6
7     }
8
9 }
10
```

### Criando as entidades do Projeto

Criando o pacote Modelo e ainda criando a classe Pessoa.java que representa um paciente no cenário de Médico:



Cujo código é:

```
1 package br.edu.ifmt.edu.projeto.modelo;
2
3 import javax.persistence.Entity;
4 import javax.persistence.GeneratedValue;
5 import javax.persistence.GenerationType;
6 import javax.persistence.Id;
7 import javax.persistence.Table;
8
9 @Entity
10 @Table(name = "TB_PESSOA")
11 public class Pessoa {
12     @Id
13     @GeneratedValue(strategy = GenerationType.IDENTITY)
14     private Integer id;
15     private String nome;
16     private Integer idade;
17
18     public Pessoa(Integer id, String nome, Integer idade) {
19         this.id = id;
20         this.nome = nome;
21         this.idade = idade;
22     }
23
24     //Getters e Setters
25
26     @Override
27     public String toString() {
28         return "Pessoa [id=" + id + ", nome=" + nome + ", idade=" + idade + "]";
29     }
30 }
31
```

Alterando a classe Main.java para persistir primeiramente dados de um paciente no banco de dados:

Seguinte código:

```
1 package br.edu.ifmt.edu.projeto;
2
3 import javax.persistence.EntityManager;
4 import javax.persistence.EntityManagerFactory;
5 import javax.persistence.Persistence;
6
7 import br.edu.ifmt.edu.projeto.modelo.Pessoa;
8
9 public class Main {
10
11     public static void main(String[] args) {
12         Pessoa p1 = new Pessoa(null, "Rafael", 25);
13
14         EntityManagerFactory emf = Persistence.createEntityManagerFactory("persistenceUnitName");
15         EntityManager em = emf.createEntityManager();
16
17         em.getTransaction().begin();
18         em.persist(p1);
19         em.getTransaction().commit();
20         System.out.println("Cadastrado com sucesso!");
21
22         em.close();
23         emf.close();
24     }
25 }
26
```

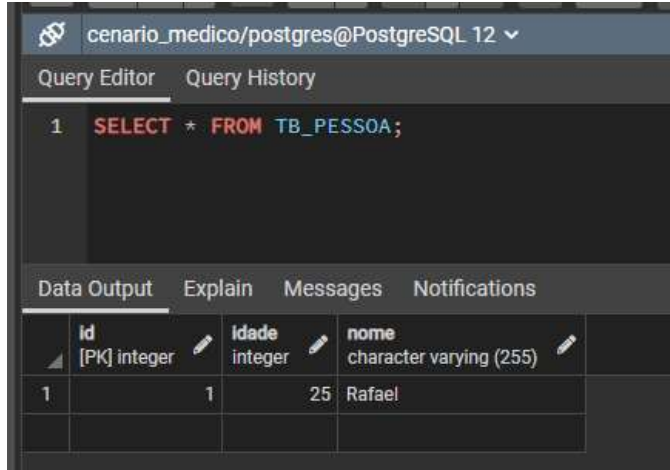
Na linha 14, é passado o nome da persistenceUnit do arquivo persistence.xml. Após isso, é instanciada o entityManager e criado. Na linha 17 é iniciada a transação, persistido a criação de p1 e logo após, fazendo o commit da transação. Nas linhas 22 e 23 são feitas as chamadas de fechamento do EntityManager.

## Verificando no Banco de Dados

Para ver os dados de p1 no banco de dados, é usada a seguinte consulta:

```
SELECT * FROM TB_PESSOA;
```

Resultado da consulta:



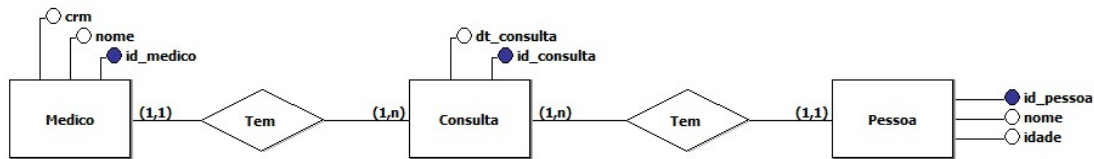
The screenshot shows a PostgreSQL Query Editor window with the title 'cenario\_medico/postgres@PostgreSQL 12'. The 'Query Editor' tab is active, displaying the query: `1 SELECT * FROM TB_PESSOA;`. Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is selected, showing a table with the following data:

	id [PK] integer	idade integer	nome character varying (255)
1	1	25	Rafael

Portanto, os dados estão sendo persistidos.

## Criando as entidades para o Cenário Médico do Projeto

Relembrando o cenário médico que possui as seguintes tabelas:



Como a tabela Pessoa que representa os pacientes já foi criada, a **entidade medico** será a próxima:

```
1 package br.edu.ifmt.edu.projeto.modelo;
2
3 import javax.persistence.Entity;
4 import javax.persistence.GeneratedValue;
5 import javax.persistence.GenerationType;
6 import javax.persistence.Id;
7 import javax.persistence.Table;
8
9 @Entity
10 @Table(name = "TB_MEDICO")
11 public class Medico {
12     @Id
13     @GeneratedValue(strategy = GenerationType.IDENTITY)
14     private Integer id;
15     private String nome;
16     private Integer crm;
17
18     public Medico(Integer id, String nome, Integer crm) {
19         this.id = id;
20         this.nome = nome;
21         this.crm = crm;
22     }
23
24     //Getters e Setters
25
26     @Override
27     public String toString() {
28         return "Medico [id=" + id + ", nome=" + nome + ", crm=" + crm + "];";
29     }
30 }
31
```

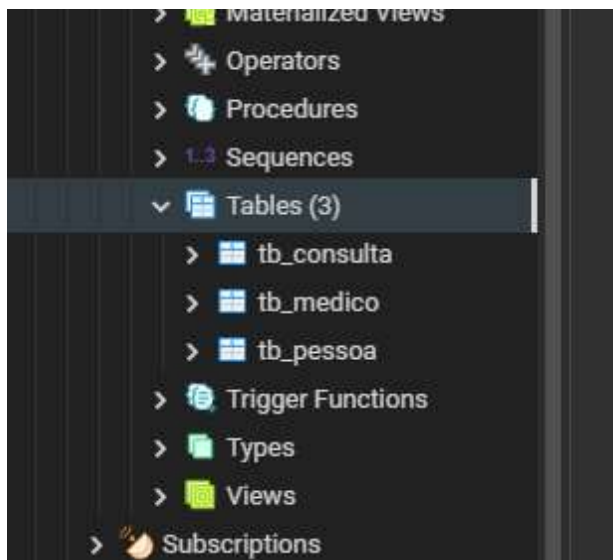


## Criando a entidade consulta

Como a entidade consulta relaciona com outras tabelas, é necessário mapear as chaves estrangeiras dentro da mesma, usando a anotação @ManyToOne.

```
1 package br.edu.ifmt.edu.projeto.modelo;
2
3 import java.time.LocalDate;
4 import javax.persistence.Entity;
5 import javax.persistence.GeneratedValue;
6 import javax.persistence.GenerationType;
7 import javax.persistence.Id;
8 import javax.persistence.ManyToOne;
9 import javax.persistence.Table;
10
11 @Entity
12 @Table(name = "TB_CONSULTA")
13 public class Consulta {
14
15     @Id
16     @GeneratedValue(strategy = GenerationType.IDENTITY)
17     private Integer id;
18     private LocalDate dtConsulta;
19     @ManyToOne
20     private Pessoa paciente;
21     @ManyToOne
22     private Medico medico;
23
24     public Consulta(Integer id, LocalDate dtConsulta, Pessoa paciente, Medico medico) {
25         this.id = id;
26         this.dtConsulta = dtConsulta;
27         this.paciente = paciente;
28         this.medico = medico;
29     }
30     //Getters e Setters
31     @Override
32     public String toString() {
33         return "Consulta [id=" + id + ", dtConsulta=" + dtConsulta + ", paciente=" + paciente + ", medico=" + medico
34             + "]\n";
35     }
36 }
37
```

Após criar as três classes, roda-se a aplicação para o Hibernate criar as tabelas no banco de dados:





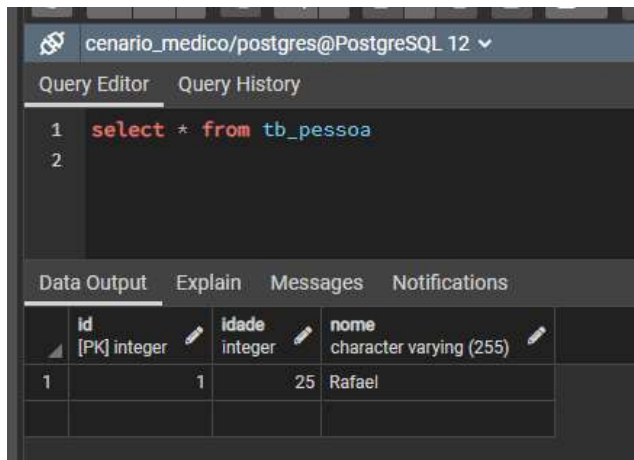
## Povoando as tabelas do Banco de Dados

Para povoar as tabelas, serão instanciados novos objetos na classe Main.java, no construtor dos objetos o atributo id é definido como "null", pois a criação do id do registro é delegado ao banco de dados:

```
public class Main {  
    public static void main(String[] args) {  
        Pessoa p1 = new Pessoa(null, "Rafael", 25);  
        Medico m1 = new Medico(null, "Paulo", 4125);  
        Consulta c1 = new Consulta(null, LocalDate.now(), p1, m1);  
  
        EntityManagerFactory emf = Persistence.createEntityManagerFactory("persistenceUnitName");  
        EntityManager em = emf.createEntityManager();  
  
        em.getTransaction().begin();  
        em.persist(p1);  
        em.persist(m1);  
        em.persist(c1);  
        em.getTransaction().commit();  
        System.out.println("Cadastrado com sucesso!");  
  
        em.close();  
        emf.close();  
    }  
}
```

Verificando no Banco de Dados os dados salvados:

Na tabela TB\_PESSOA:



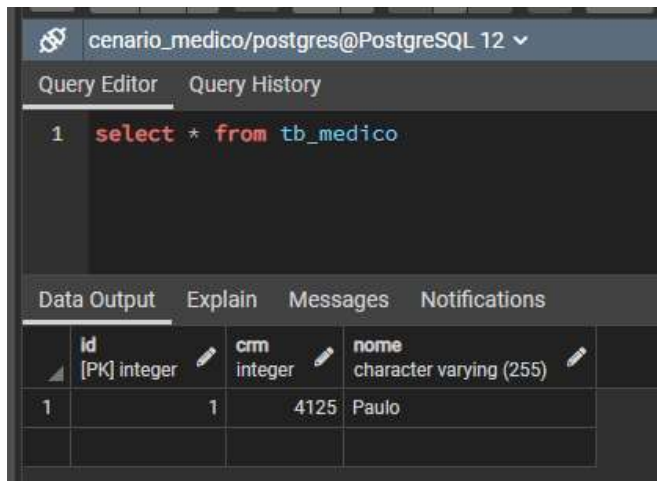
The screenshot shows a PostgreSQL query editor with the following query in the Query Editor tab:

```
1 select * from tb_pessoa  
2
```

The Data Output tab displays the results of the query:

	id [PK] integer	idade integer	nome character varying (255)
1	1	25	Rafael

Na tabela TB\_MEDICO:



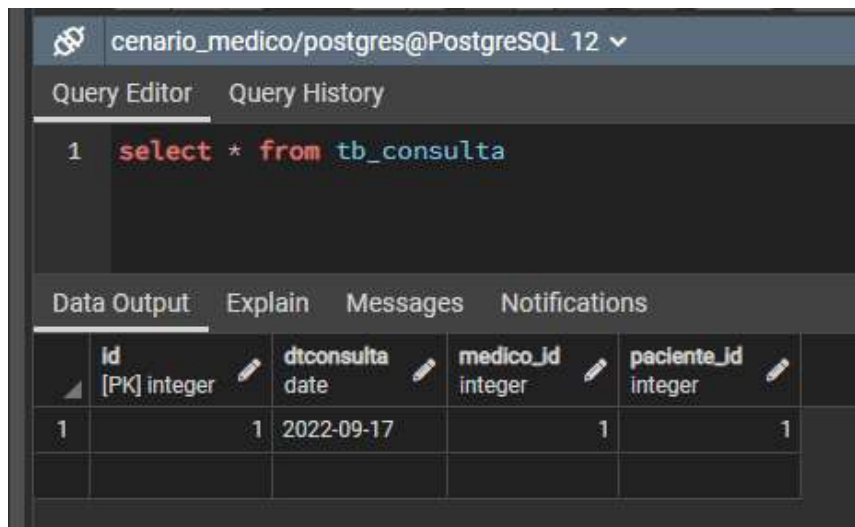
The screenshot shows a PostgreSQL query editor with the following query in the Query Editor tab:

```
1 select * from tb_medico
```

The Data Output tab displays the results of the query:

	id [PK] integer	crm integer	nome character varying (255)
1	1	4125	Paulo

E na tabela TB\_CONSULTA:

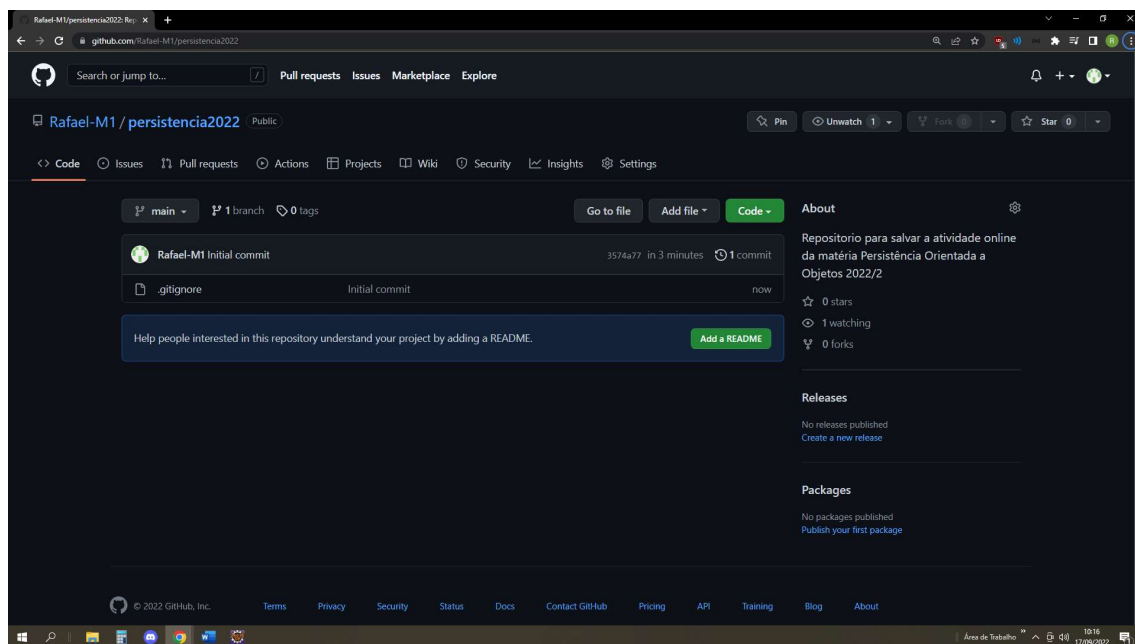


The screenshot shows a PostgreSQL query editor interface. At the top, the connection is 'cenario\_medico/postgres@PostgreSQL 12'. The 'Query Editor' tab is active, displaying the SQL query: `1 select * from tb_consulta`. Below the query editor, the 'Data Output' tab is selected, showing a table with 5 columns: 'Id [PK] integer', 'dtconsulta date', 'medico\_Id integer', and 'paciente\_Id integer'. The first row of data is visible, with values: '1', '2022-09-17', '1', and '1'.

	Id [PK] integer	dtconsulta date	medico_Id integer	paciente_Id integer
1	1	2022-09-17	1	1

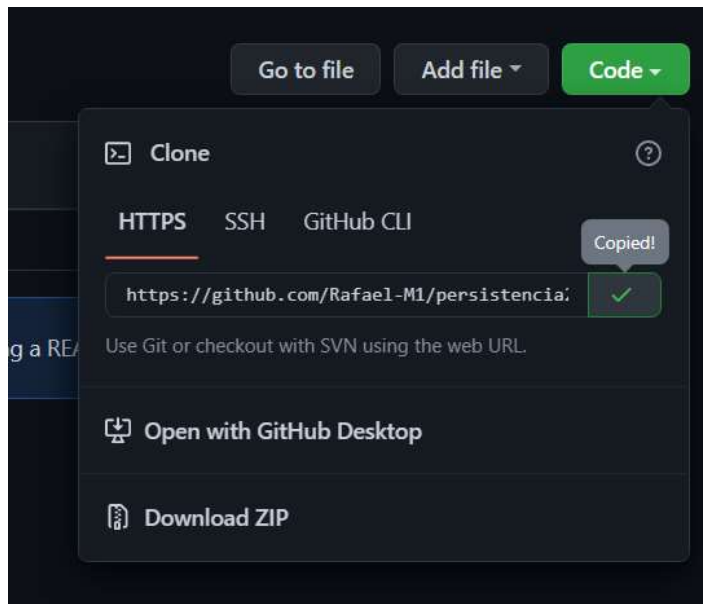
### Salvando o Projeto no Github

Após terminado o projeto, é necessário salvar o projeto no GitHub, primeiramente criando um repositório com o nome persistencia2022:

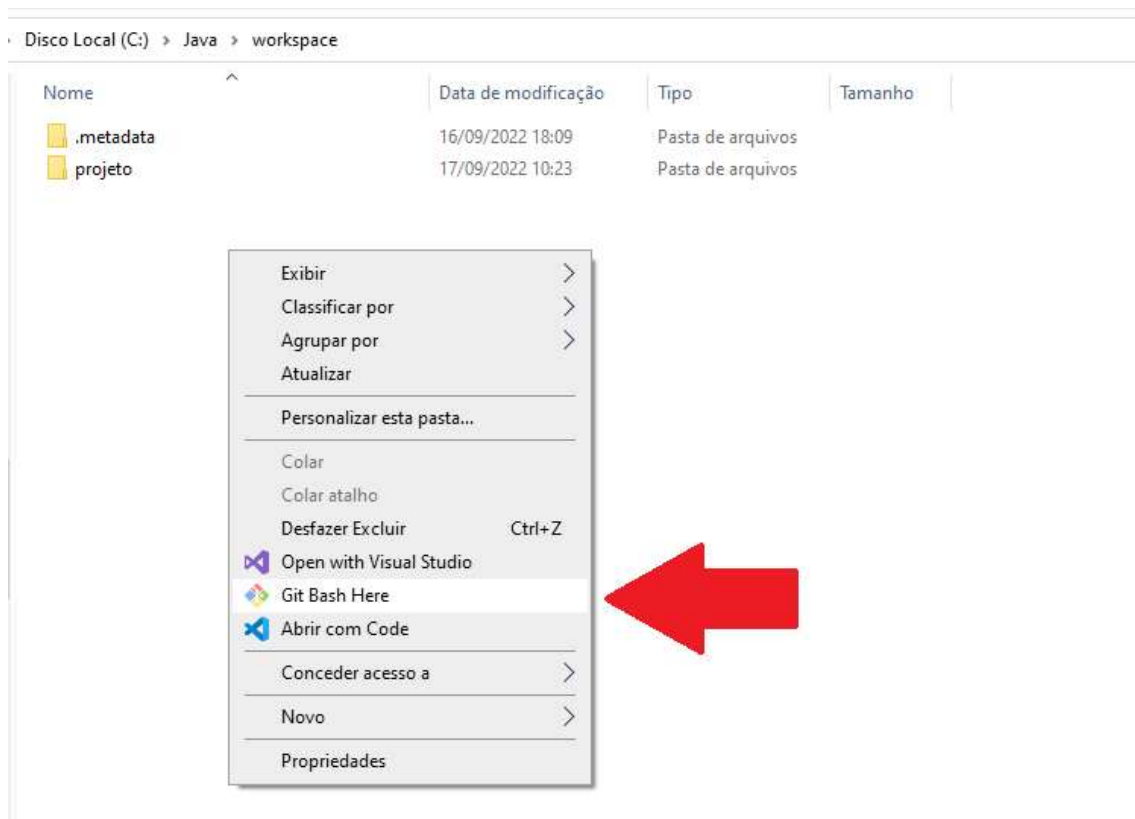


Na aba Code, copie o link do repositório, nesse caso:

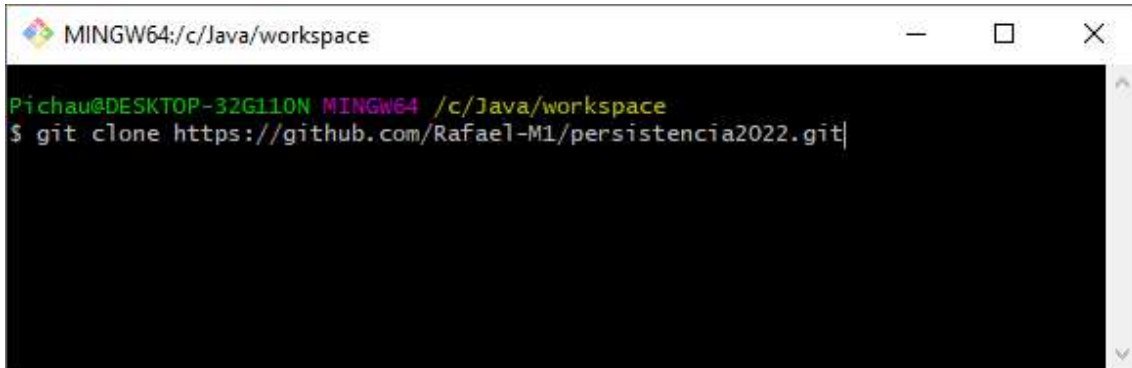
<https://github.com/Rafael-M1/persistencia2022.git>



Na pasta onde está o projeto: “C:\Java\workspace” abra o terminal do Git Bash com o botão direito:

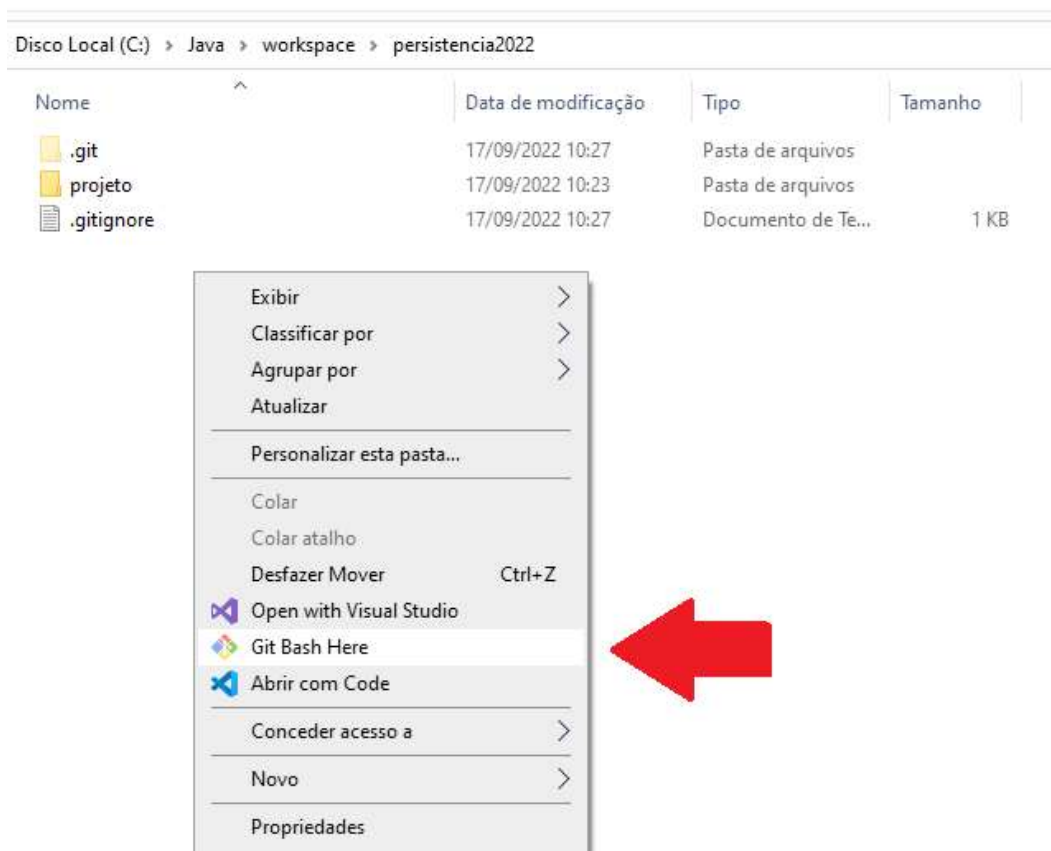


No terminal, digite:

A screenshot of a terminal window titled 'MINGW64:/c/Java/workspace'. The prompt is 'Pichau@DESKTOP-32G110N MINGW64 /c/Java/workspace'. The command entered is '\$ git clone https://github.com/Rafael-M1/persistencia2022.git'.

```
MINGW64:/c/Java/workspace
Pichau@DESKTOP-32G110N MINGW64 /c/Java/workspace
$ git clone https://github.com/Rafael-M1/persistencia2022.git
```

Arraste a pasta projeto para dentro de persistencia2022 e abra o terminal dentro da pasta persistência2022:



No terminal, digite “git status” e depois “git add .” :

```
MINGW64:/c/Java/workspace/persistencia2022

Pichau@DESKTOP-32G110N MINGW64 /c/Java/workspace/persistencia2022 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    projeto/

nothing added to commit but untracked files present (use "git add" to track)

Pichau@DESKTOP-32G110N MINGW64 /c/Java/workspace/persistencia2022 (main)
$ git add .

Pichau@DESKTOP-32G110N MINGW64 /c/Java/workspace/persistencia2022 (main)
$ |
```

Depois digite git commit -m “Projeto criado” :

```
MINGW64:/c/Java/workspace/persistencia2022

Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    projeto/

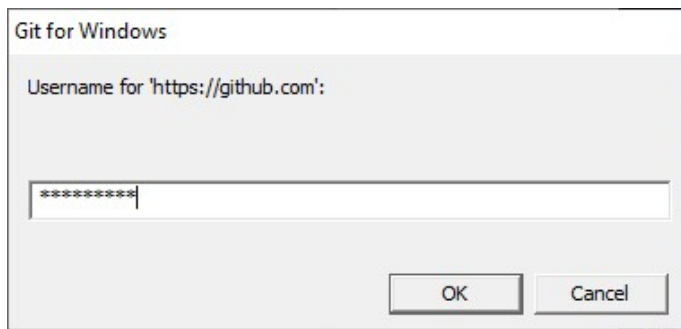
nothing added to commit but untracked files present (use "git add" to track)

Pichau@DESKTOP-32G110N MINGW64 /c/Java/workspace/persistencia2022 (main)
$ git add .

Pichau@DESKTOP-32G110N MINGW64 /c/Java/workspace/persistencia2022 (main)
$ git commit -m "Projeto criado"
[main 0695fd2] Projeto criado
14 files changed, 370 insertions(+)
create mode 100644 projeto/.classpath
create mode 100644 projeto/.project
create mode 100644 projeto/.settings/org.eclipse.jdt.core.preferences
create mode 100644 projeto/.settings/org.eclipse.m2e.core.preferences
create mode 100644 projeto/pom.xml
create mode 100644 projeto/src/main/java/META-INF/persistence.xml
create mode 100644 projeto/src/main/java/br/edu/ifmt/edu/projeto/Main.java
create mode 100644 projeto/src/main/java/br/edu/ifmt/edu/projeto/modelo/Consulta.java
create mode 100644 projeto/src/main/java/br/edu/ifmt/edu/projeto/modelo/Medico.java
create mode 100644 projeto/src/main/java/br/edu/ifmt/edu/projeto/modelo/Pessoa.java
create mode 100644 projeto/target/classes/META-INF/MANIFEST.MF
create mode 100644 projeto/target/classes/META-INF/maven/br.edu.ifmt.projeto/pom.properties
create mode 100644 projeto/target/classes/META-INF/maven/br.edu.ifmt.projeto/pom.xml
create mode 100644 projeto/target/classes/META-INF/persistence.xml

Pichau@DESKTOP-32G110N MINGW64 /c/Java/workspace/persistencia2022 (main)
$ |
```

Depois “git push” e o terminal irá pedir as credenciais para fazer o push:



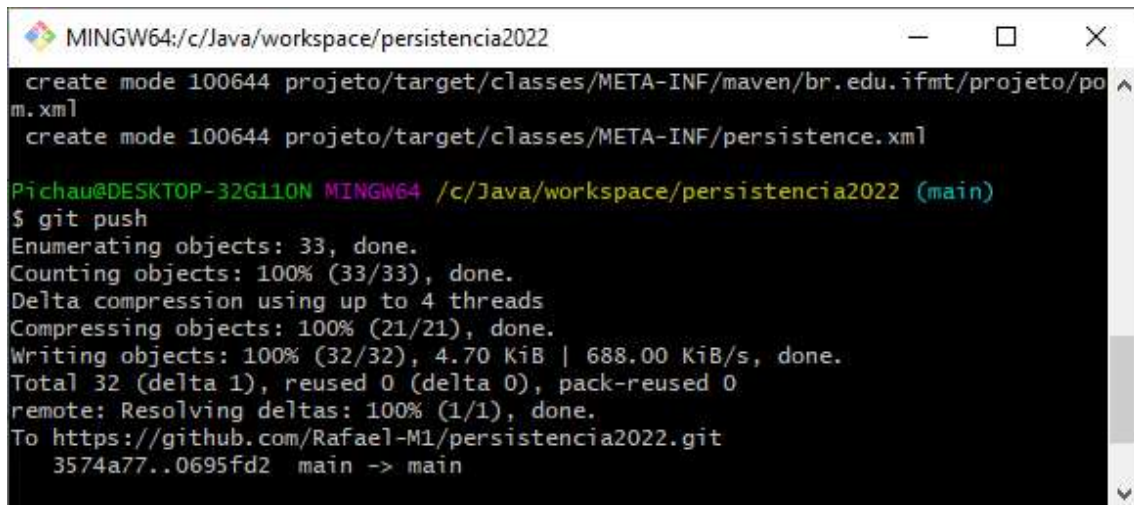
A dialog box titled "Git for Windows". It contains the text "Username for 'https://github.com':". Below this text is a text input field containing several asterisks. At the bottom right of the dialog are two buttons: "OK" and "Cancel".

E depois:



A dialog box titled "Git for Windows". It contains the text "Password for 'https://rafael-m1@github.com':". Below this text is a text input field containing several asterisks. At the bottom right of the dialog are two buttons: "OK" and "Cancel".

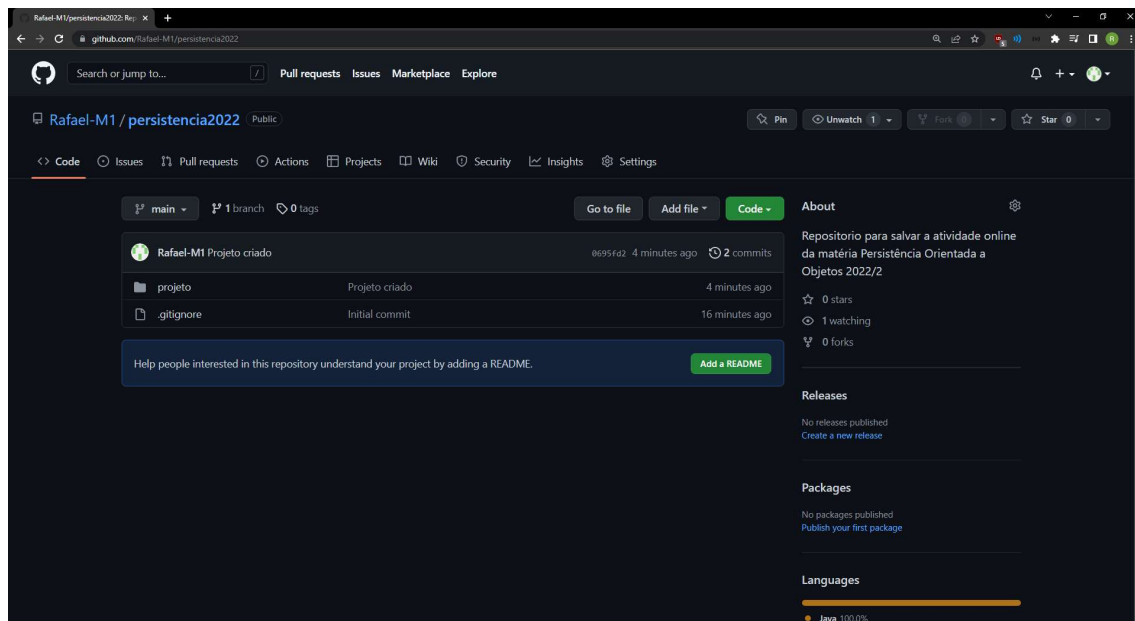
No console, caso o processo ocorra corretamente irá aparecer a seguinte mensagem:



```
MINGW64:/c/Java/workspace/persistencia2022
create mode 100644 projeto/target/classes/META-INF/maven/br.edu.ifmt/projeto/po
m.xml
create mode 100644 projeto/target/classes/META-INF/persistence.xml
Pichau@DESKTOP-32G110N MINGW64 /c/Java/workspace/persistencia2022 (main)
$ git push
Enumerating objects: 33, done.
Counting objects: 100% (33/33), done.
Delta compression using up to 4 threads
Compressing objects: 100% (21/21), done.
Writing objects: 100% (32/32), 4.70 KiB | 688.00 KiB/s, done.
Total 32 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Rafael-M1/persistencia2022.git
3574a77..0695fd2  main -> main
```



Por fim, verificando no repositório do github o projeto foi adicionado corretamente:



Link do repositório:

<https://github.com/Rafael-M1/persistencia2022>