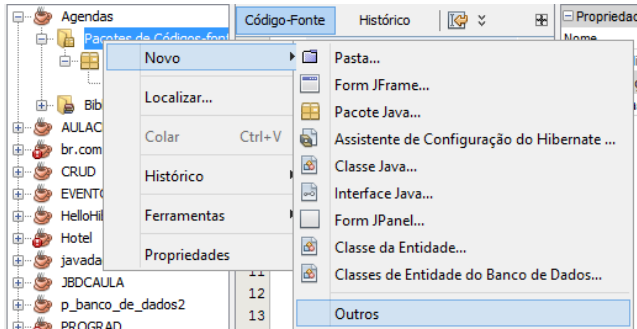
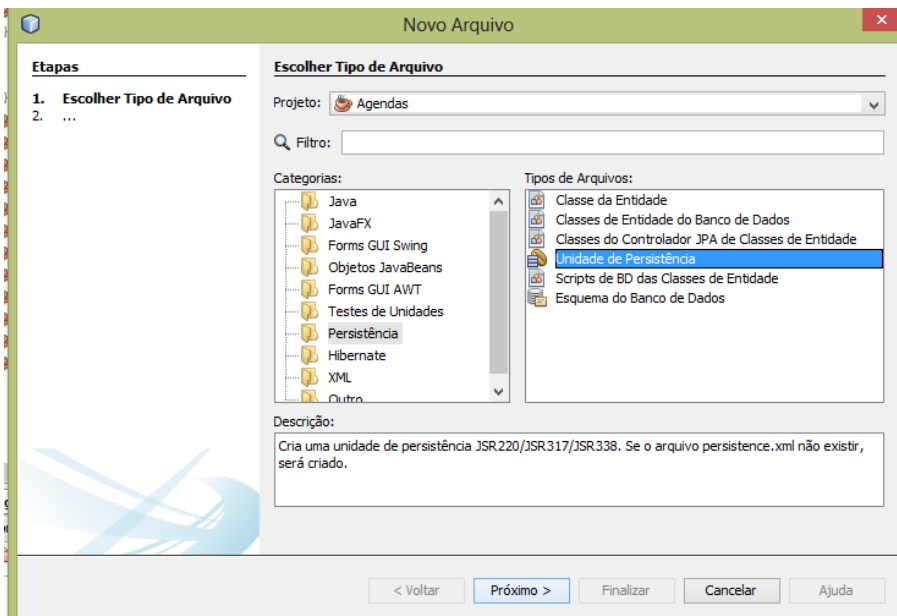


Introdução a JPA – Java Persistence API

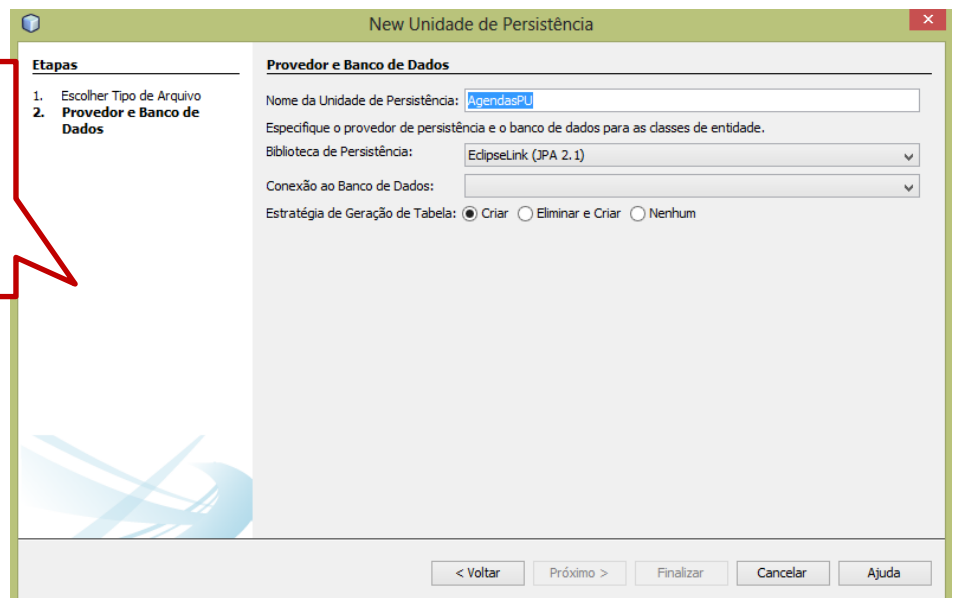
- 1- Criar o projeto (Arquivo / novo projeto / Aplicação Java)
- 2- Criar Pacote agenda
- 3- Criar a unidade de Persistência. (Botão direito / outros)

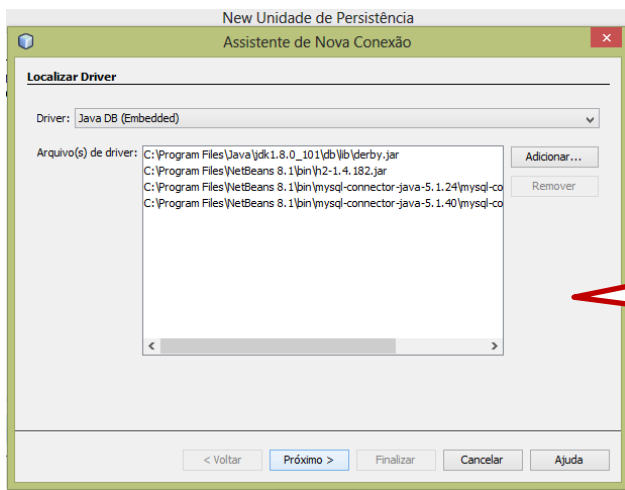


Selecionar Persistência / Unidade de Persistência , onde iremos criar o nosso arquivo persistencia. XML



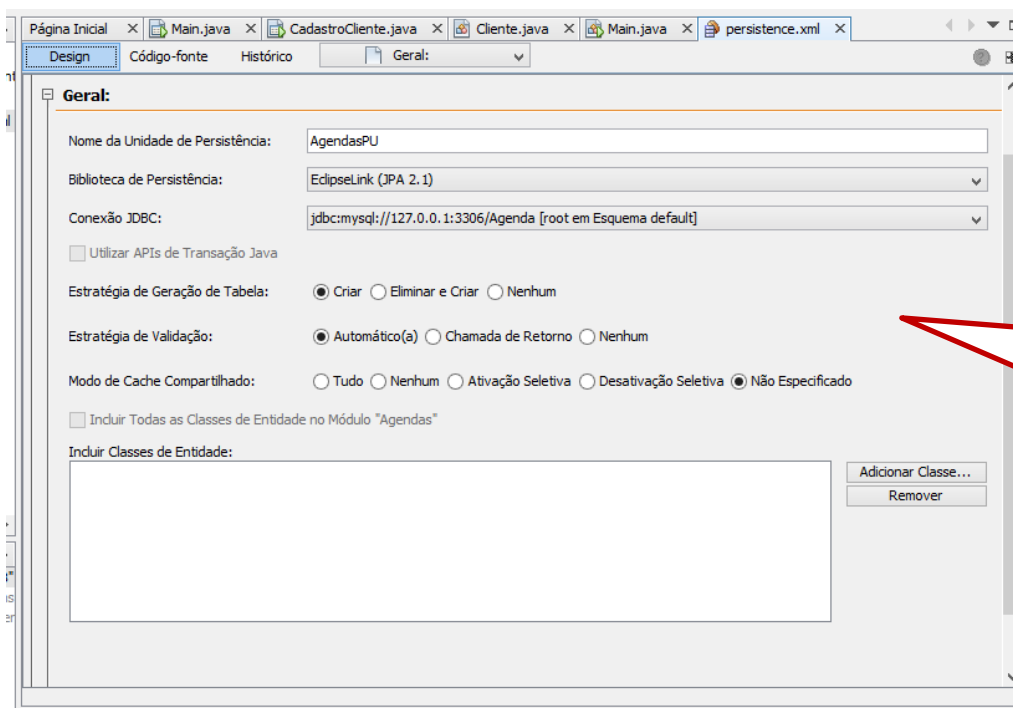
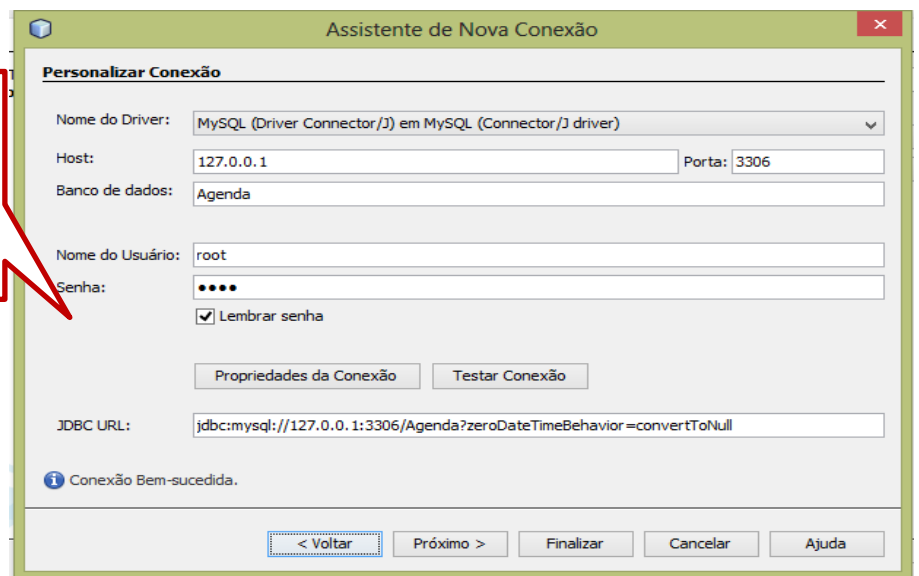
Onde sugere o nome AgendaPU e a biblioteca de Persistência EclipseLink. (Poderia ser qualquer outra... Hibernate,





Em conexão de Banco de Dados, Criar selecionar Nova Conexão, na próxima tela selecionar o drive Mysql.

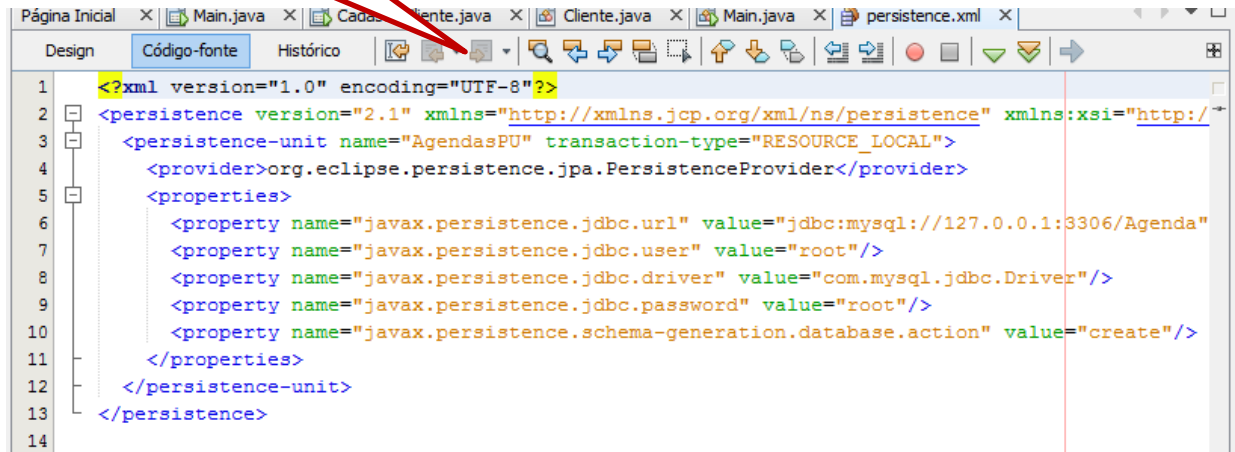
Setar: O host, o banco que criamos (Agenda), usuário e senha e testar conexão. Botão Próximo e vamos manter a opção criar e Finalizar..



Será exibido a tela de design onde será possível alterar a estratégias.

Arquivo **persistência.XML**

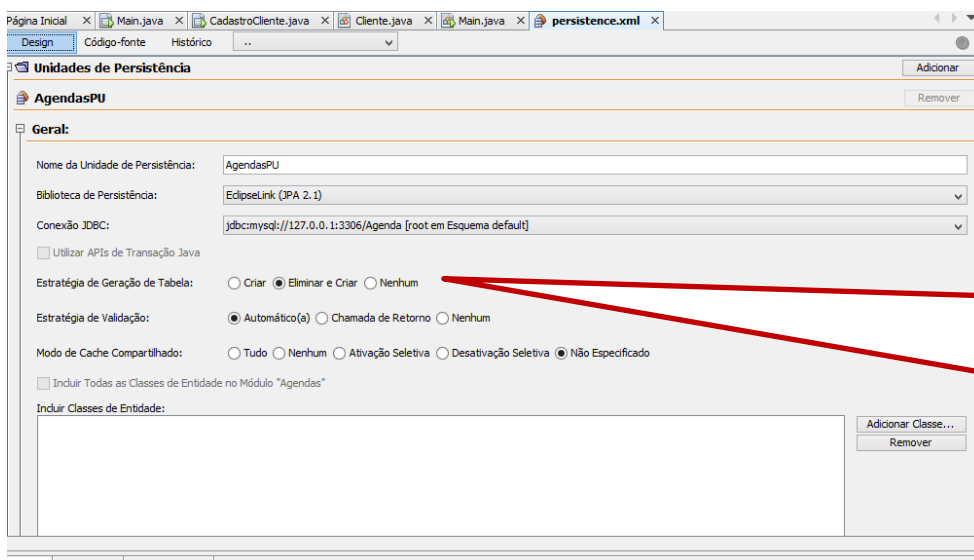
Onde mostra as propriedades de conexão com o banco.



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://
3 <persistence-unit name="AgendasPU" transaction-type="RESOURCE_LOCAL">
4 <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
5 <properties>
6 <property name="javax.persistence.jdbc.url" value="jdbc:mysql://127.0.0.1:3306/Agenda"
7 <property name="javax.persistence.jdbc.user" value="root"/>
8 <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"/>
9 <property name="javax.persistence.jdbc.password" value="root"/>
10 <property name="javax.persistence.schema-generation.database.action" value="create"/>
11 </properties>
12 </persistence-unit>
13 </persistence>
14
```

Exemplo: Ao alterar a estratégia de geração das tabelas para Eliminar e Criar (drop e create table)

Onde se a tabela existir excluir e cria novamente com o banco.



Unidades de Persistência

AgendasPU

Geral:

Nome da Unidade de Persistência: AgendasPU

Biblioteca de Persistência: EclipseLink (JPA 2.1)

Conexão JDBC: jdbc:mysql://127.0.0.1:3306/Agenda [root em Esquema default]

☐ Utilizar APIs de Transação Java

Estratégia de Geração de Tabela: ☐ Criar ☒ Eliminar e Criar ☐ Nenhum

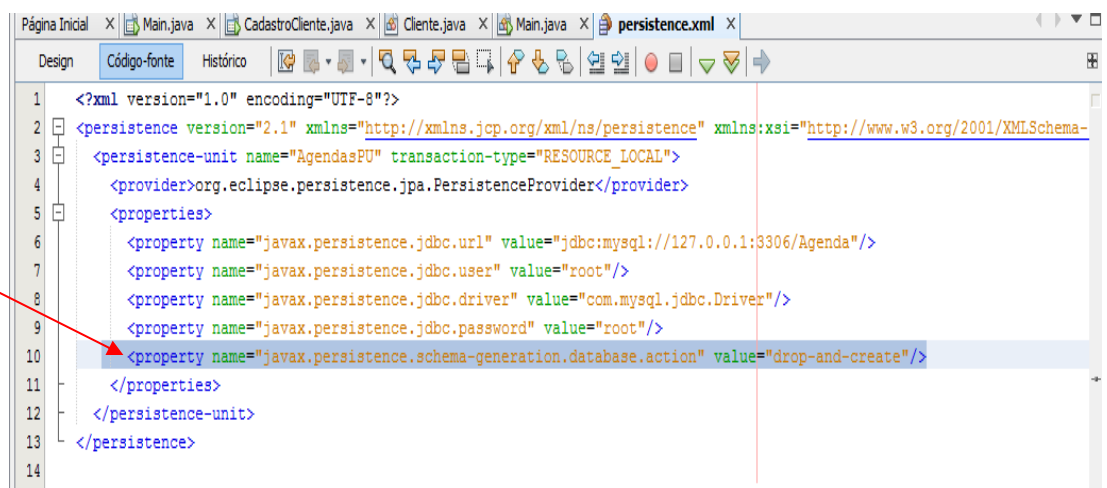
Estratégia de Validação: ☒ Automático(a) ☐ Chamada de Retorno ☐ Nenhum

Modo de Cache Compartilhado: ☐ Tudo ☐ Nenhum ☐ Ativação Seletiva ☐ Desativação Seletiva ☒ Não Especificado

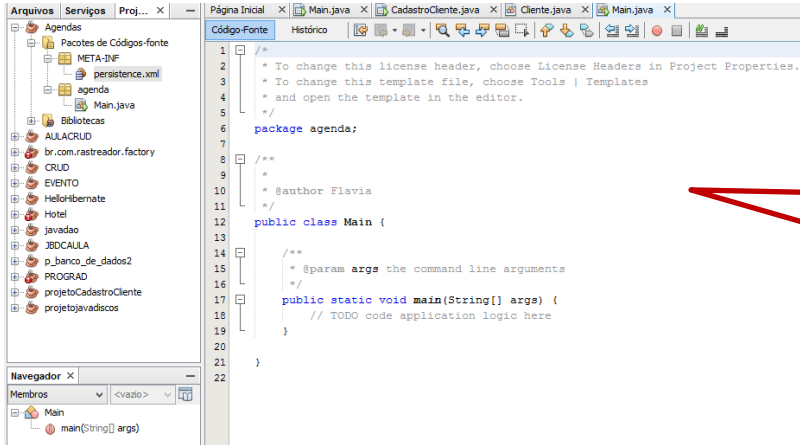
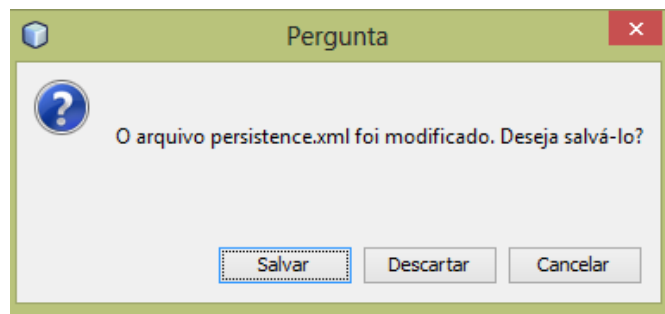
☐ Incluir Todas as Classes de Entidade no Módulo "Agendas"

Incluir Classes de Entidade:

Adicionar Classe...
Remover



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
3 <persistence-unit name="AgendasPU" transaction-type="RESOURCE_LOCAL">
4 <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
5 <properties>
6 <property name="javax.persistence.jdbc.url" value="jdbc:mysql://127.0.0.1:3306/Agenda"/>
7 <property name="javax.persistence.jdbc.user" value="root"/>
8 <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"/>
9 <property name="javax.persistence.jdbc.password" value="root"/>
10 <property name="javax.persistence.schema-generation.database.action" value="drop-and-create"/>
11 </properties>
12 </persistence-unit>
13 </persistence>
14
```

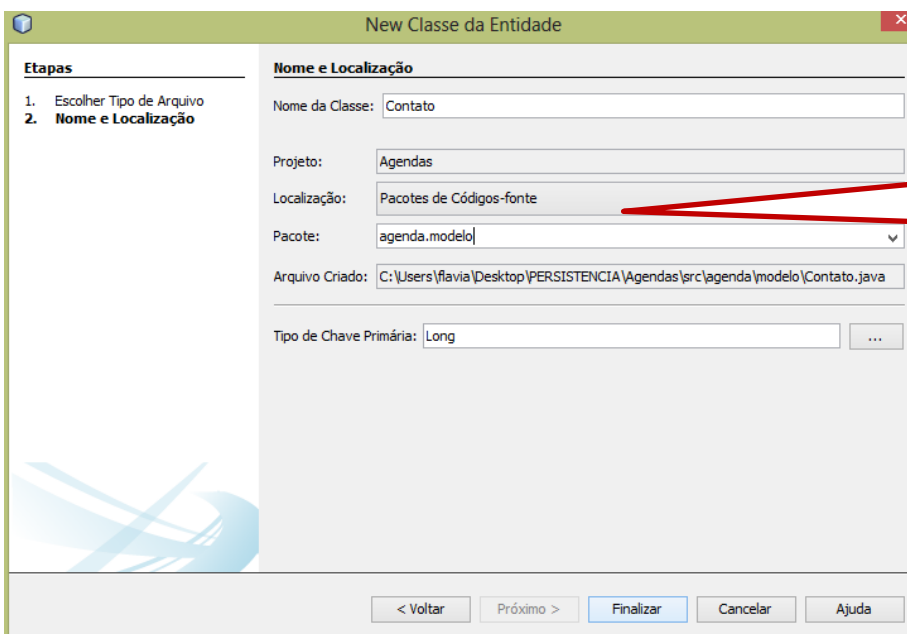
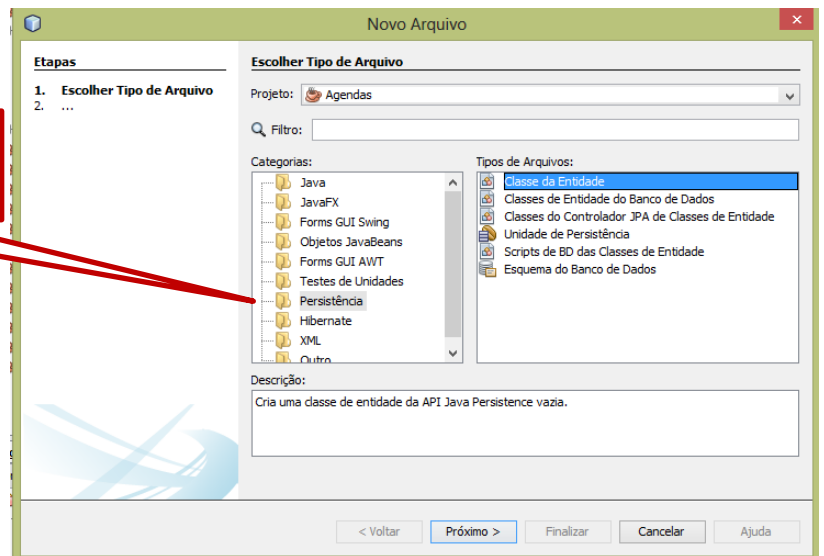


Na classe Main.java

Vamos Criar a entidade.

Botão direito / Outros / Persistência /
Classe de Entidade

Selecionar Persistência / Classe de
Persistência.



Criar a classe
Contato no pacote
agenda.modelo

Iremos usar o padrão de anotações `@Entity` , na classe `Contato`, onde também iremos os nossos métodos Getters e Setters.

@Entity

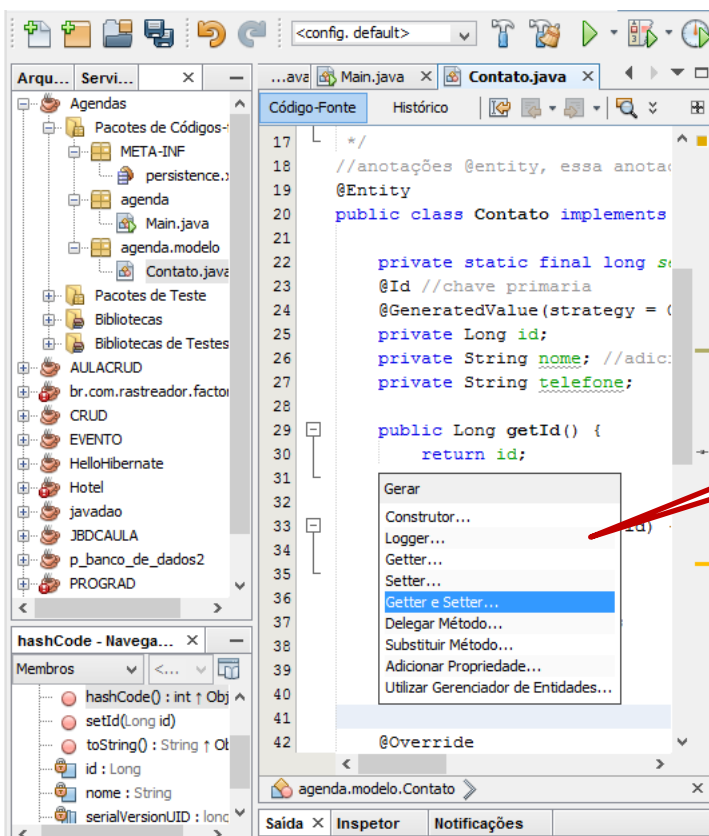
```
package agenda.modelo;

import java.io.Serializable;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

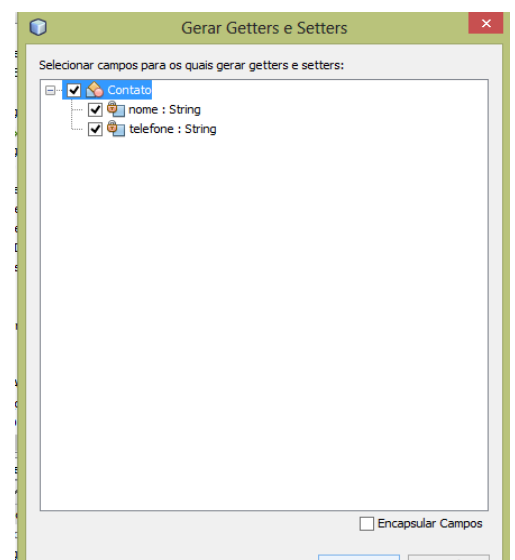
//anotações @entity, essa anotações que vai dizer q esta classe é uma entidade
@Entity
public class Contato implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id //chave primaria
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nome; //adicioanr os atributos
    private String telefone;

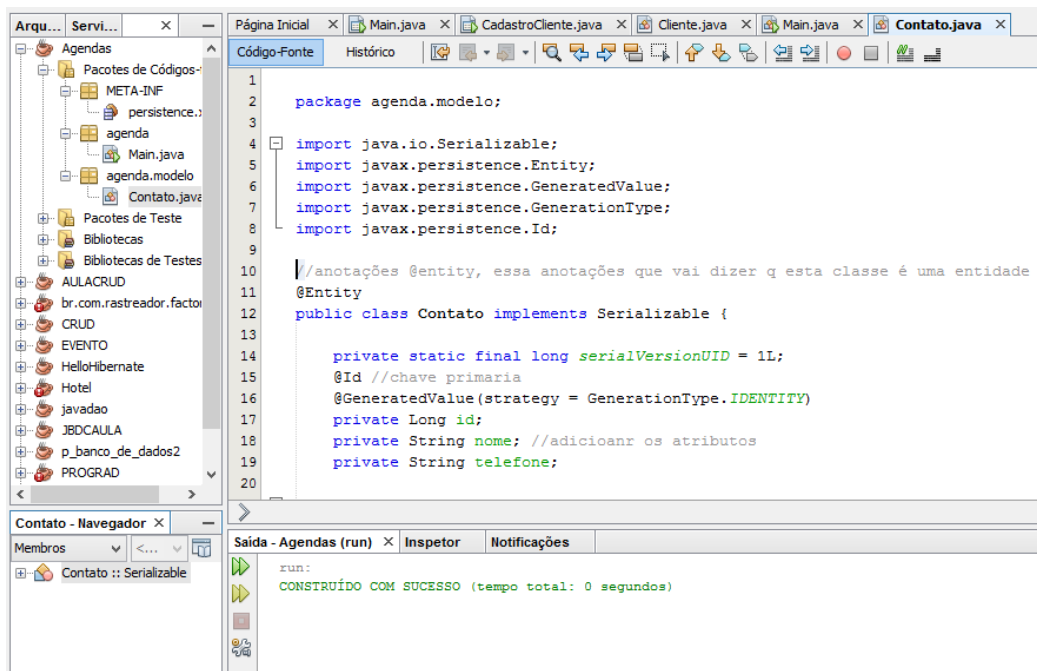
    public Long getId() {
        return id;
    }
}
```



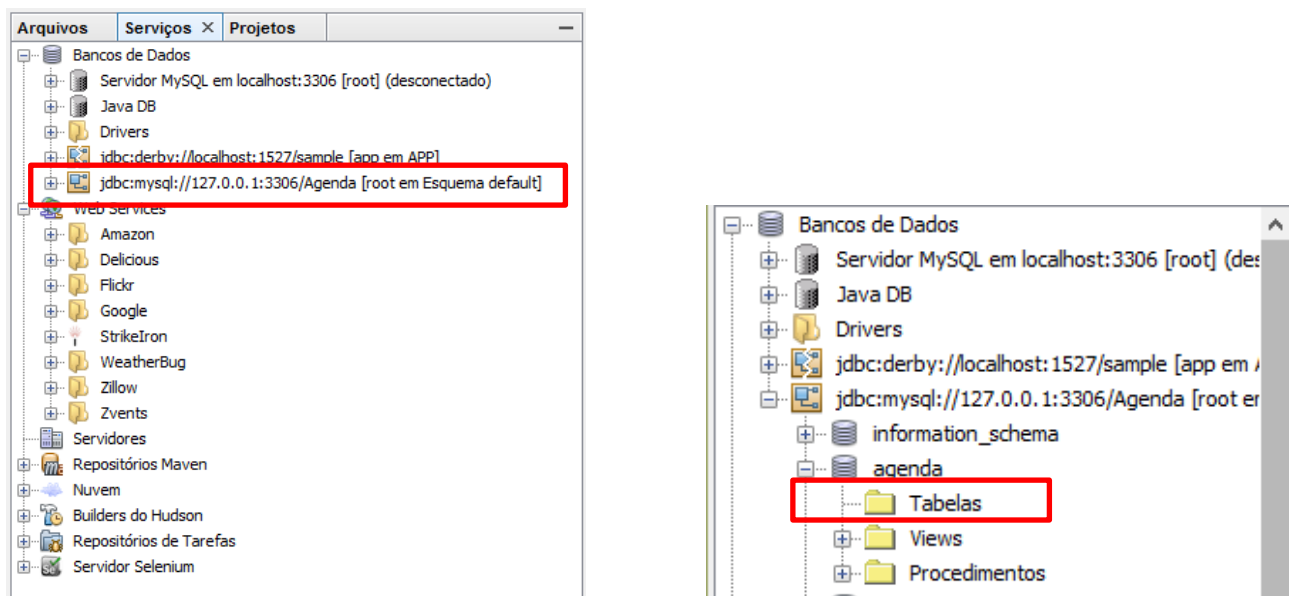
Adicionar os atributos Botão direito / Inserir código / Gerar getters e setters



Agora que temos a nossa Entidade Criada vamos executar.

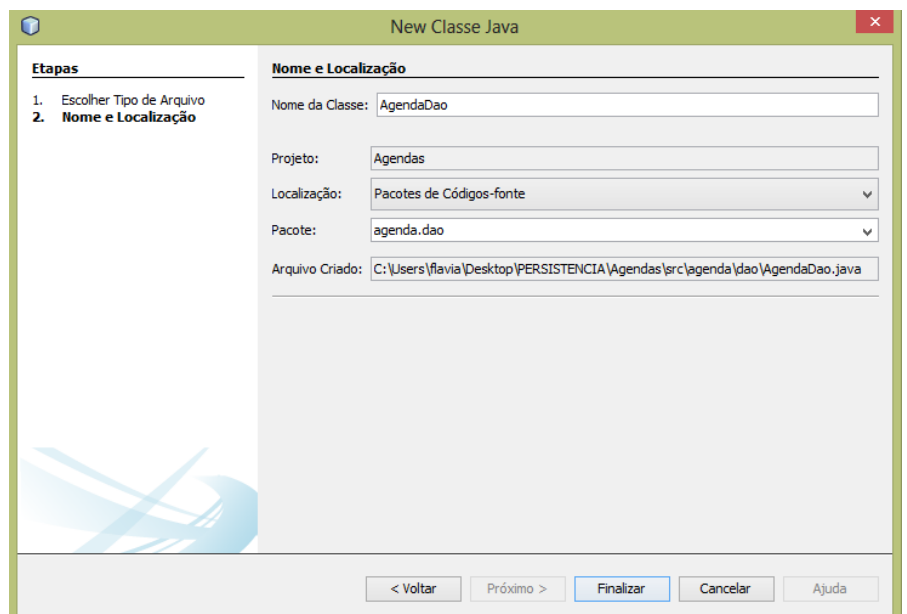
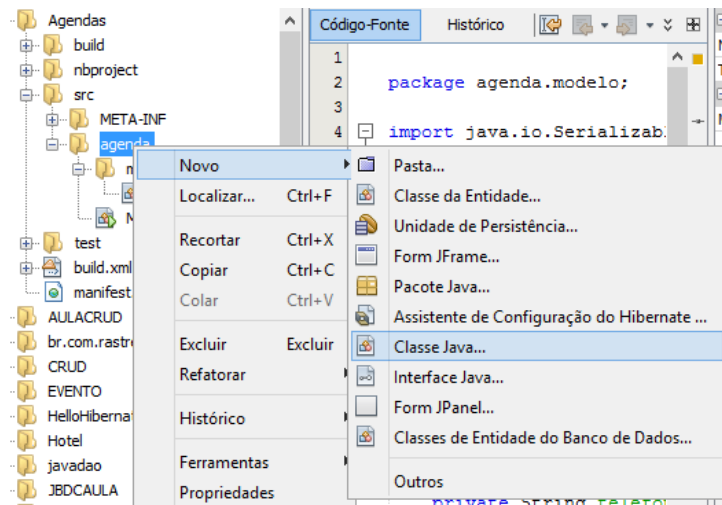


Verificamos que na Guia Serviços nas Conexões com o banco que a nossa tabela não foi criada.

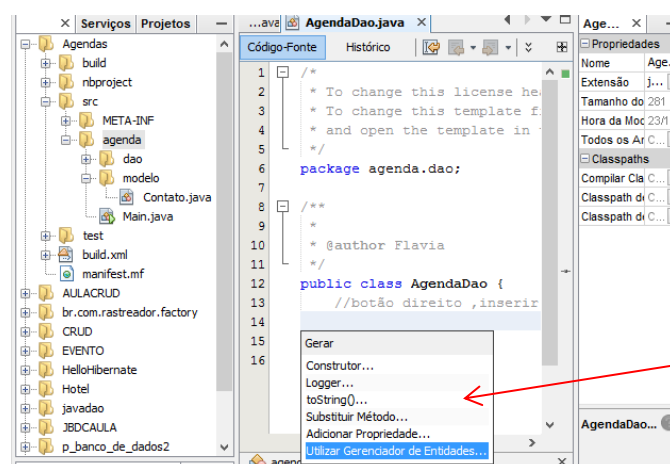


Criamos os arquivos: Persistence.xml. As classes Main e Contato

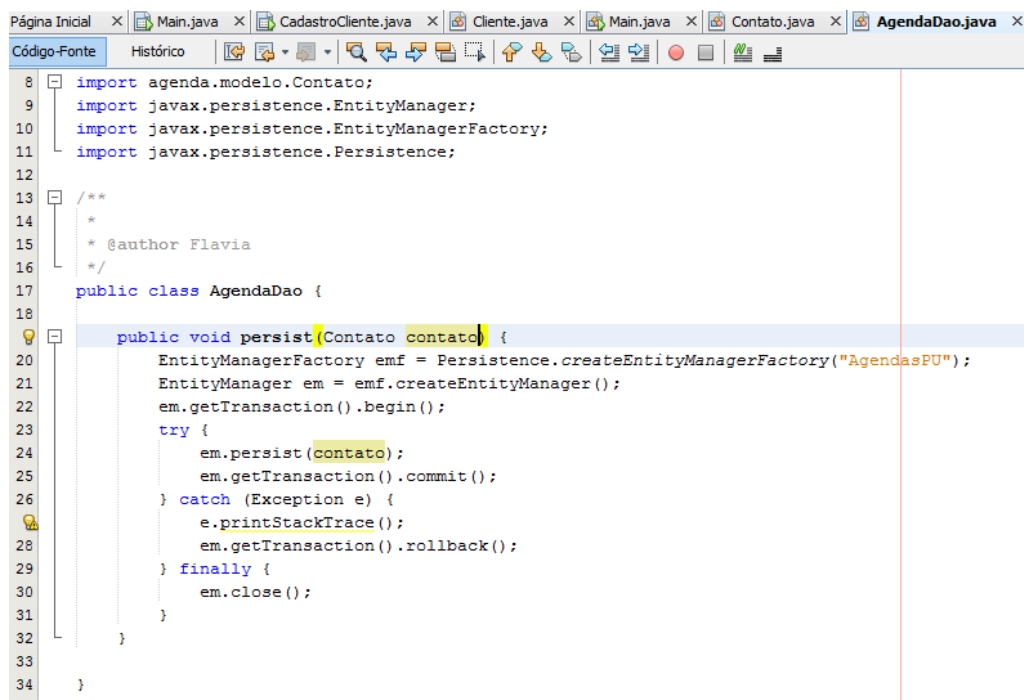
Vamos criar a classe AgendaDao



Na classe, botão direito, Inserir código, utilizar Gerenciador de Unidades



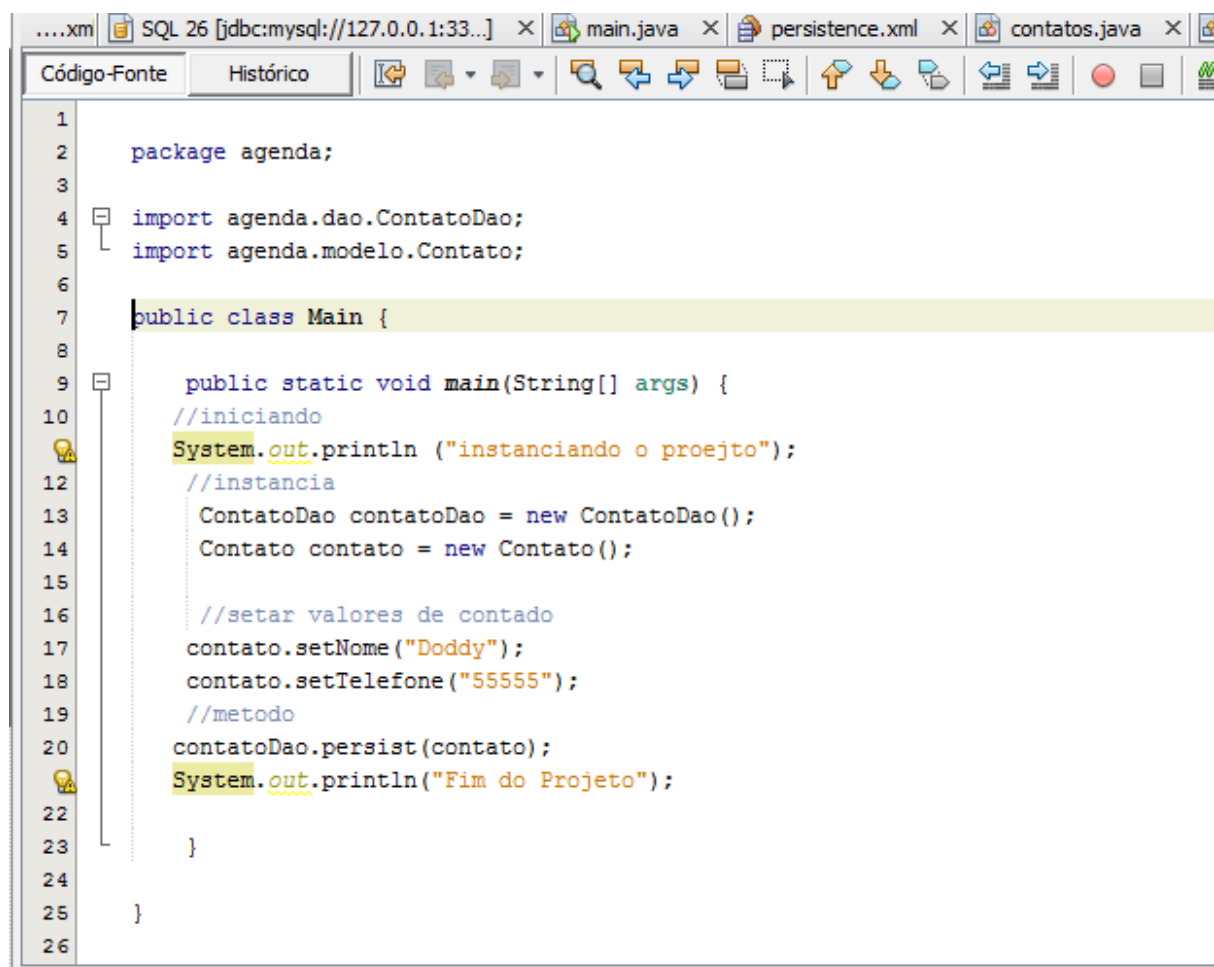
Onde vamos salvar o contato, no banco



```
8 import agenda.modelo.Contato;
9 import javax.persistence.EntityManager;
10 import javax.persistence.EntityManagerFactory;
11 import javax.persistence.Persistence;
12
13 /**
14  *
15  * @author Flavia
16  */
17 public class AgendaDao {
18
19     public void persist(Contato contato) {
20         EntityManagerFactory emf = Persistence.createEntityManagerFactory("AgendasPU");
21         EntityManager em = emf.createEntityManager();
22         em.getTransaction().begin();
23         try {
24             em.persist(contato);
25             em.getTransaction().commit();
26         } catch (Exception e) {
27             e.printStackTrace();
28             em.getTransaction().rollback();
29         } finally {
30             em.close();
31         }
32     }
33 }
34 }
```

Testando...

Na Classe Main, Vamos Iniciar / Instanciar o objeto / Setar os valores e Gerar um resultado



```
1 package agenda;
2
3
4 import agenda.dao.ContatoDao;
5 import agenda.modelo.Contato;
6
7 public class Main {
8
9     public static void main(String[] args) {
10         //iniciando
11         System.out.println("instanciando o proejto");
12         //instancia
13         ContatoDao contatoDao = new ContatoDao();
14         Contato contato = new Contato();
15
16         //setar valores de contato
17         contato.setNome("Doddy");
18         contato.setTelefone("55555");
19         //metodo
20         contatoDao.persist(contato);
21         System.out.println("Fim do Projeto");
22
23     }
24
25 }
26 }
```


Na Guia Serviços, Vamos consultar o nosso Banco de dados, botão direito /Exibir Dados

