

# PROGRAMAÇÃO ANDROID

---

## Criando web services

Universidade Federal de Sergipe  
Departamento de Sistemas de Informação  
Prof. Andrés Menéndez  
[ammenendez@gmail.com](mailto:ammenendez@gmail.com)

# Introdução

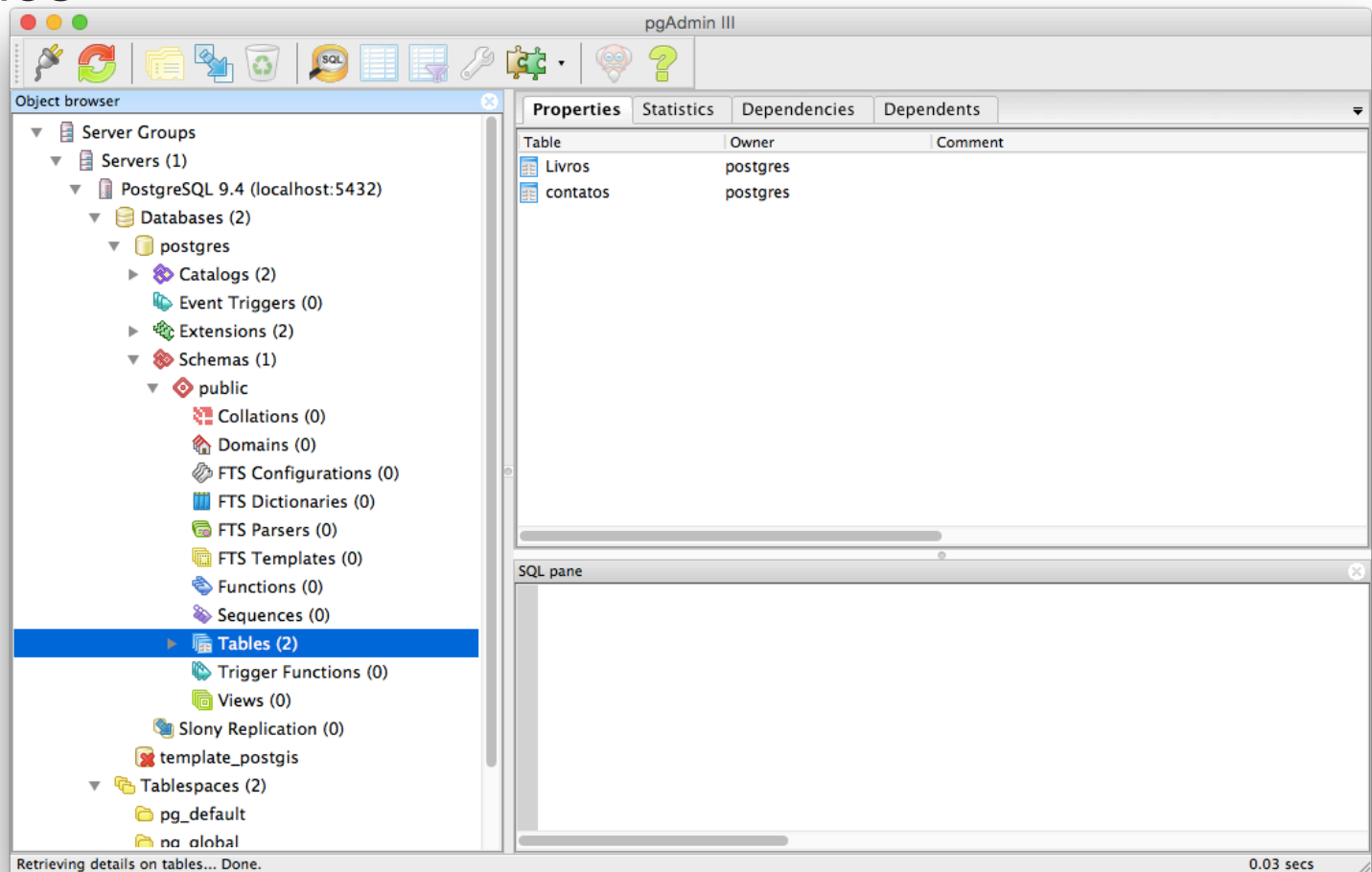
- Vamos aprender como criar web services do tipo RESTFul com a plataforma Java
- Poderia ser em várias outras plataformas, como PHP ou .NET
- As ferramentas que serão utilizadas para este exemplo são:
  - NetBeans 8.0
  - GlassFish Server
- Além disso vamos precisar de um banco de dados, que será o Postgresql

# Instalação do banco de dados

- A primeira atividade é a instalação do banco de dados
- Na instalação temos que ter cuidado com o nome do database, o nome do usuário do banco de dados e a senha que será atribuída

# Instalação do banco de dados

- Caso tudo tenha transcorrido normalmente podemos abrir o programa pgAdmin, que permite manipular o banco de dados



# Tabelas

- Adicione uma tabela de pessoas que terá as colunas: id, nome e fone
- Deixe a coluna id como sendo a chave primária da tabela
- Esse é o comando que cria a tabela

```
CREATE TABLE pessoas(  
    id serial NOT NULL,  
    nome character(20),  
    fone character(10),  
    CONSTRAINT pk_pessoas PRIMARY KEY (id)  
)
```

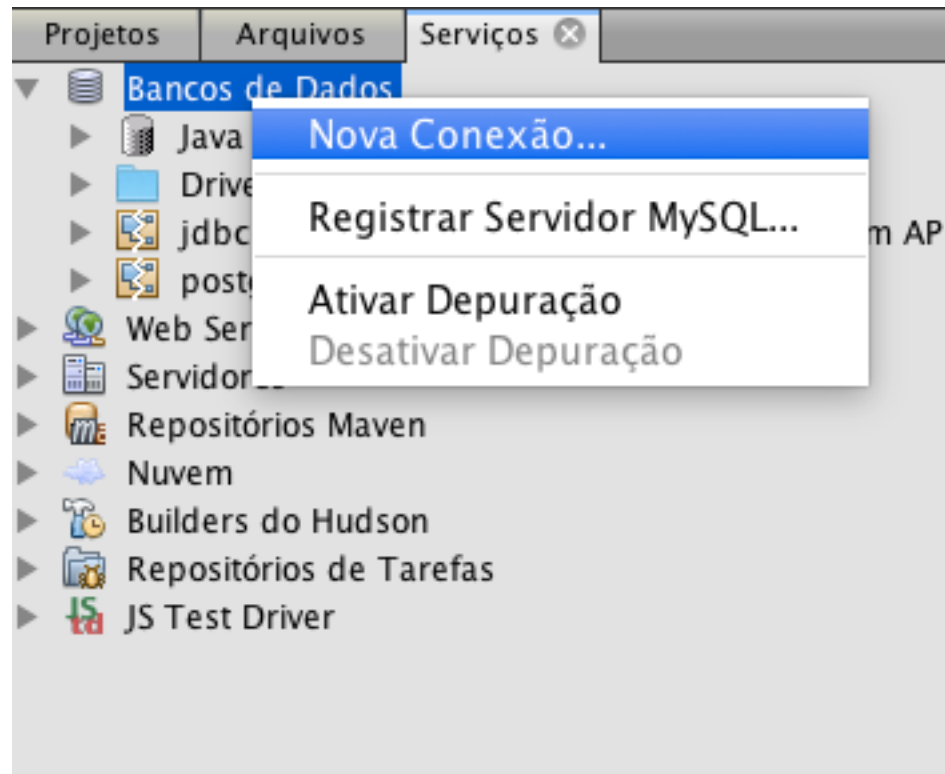
- O tipo serial indica que o campo id será do tipo auto incremento

# Tabelas

- Por enquanto isto é o suficiente para o banco de dados
- Mas tenha em mente que o modelo de dados provavelmente será bem mais complexo do que este exemplo

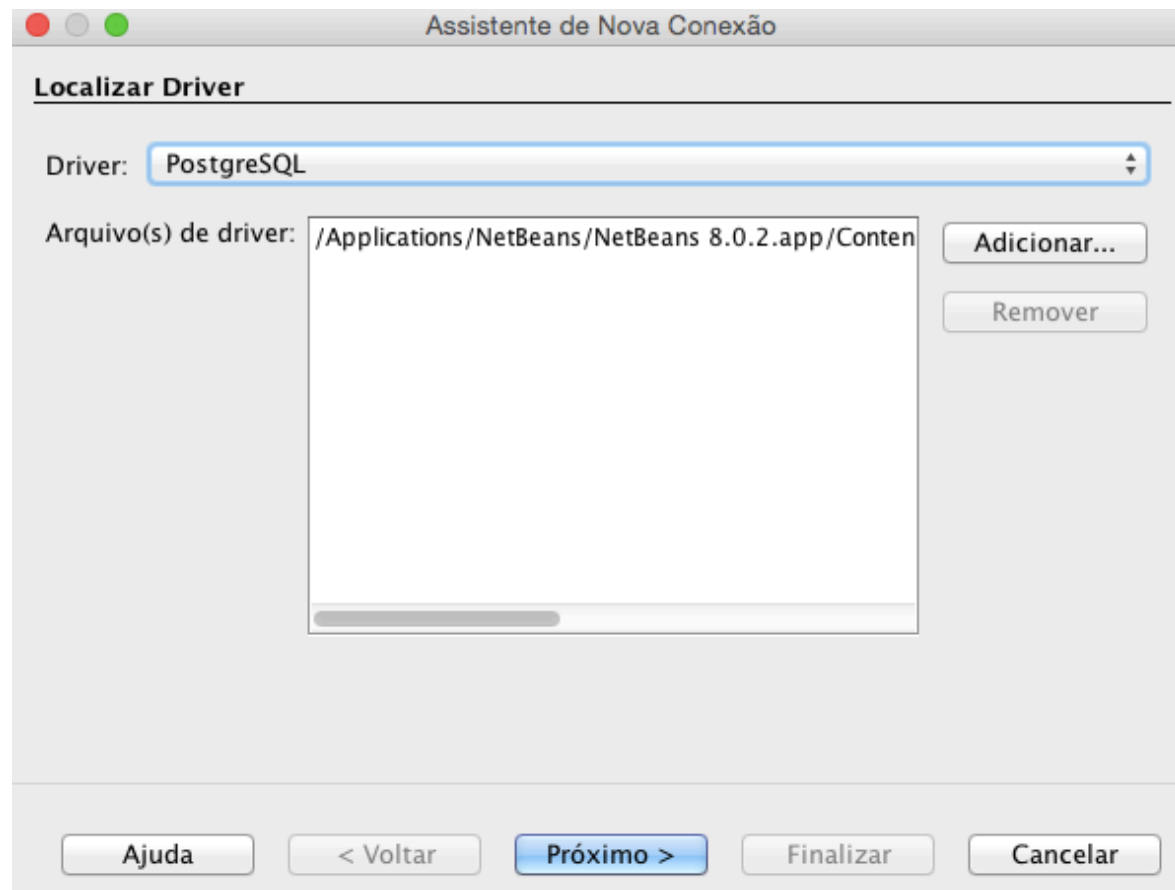
# Criando a conexão no NetBeans

- No NetBeans vai ser necessário criar uma conexão com o banco de dados
- Isto é feito adicionando uma Nova Conexão



# Criando uma conexão no NetBeans

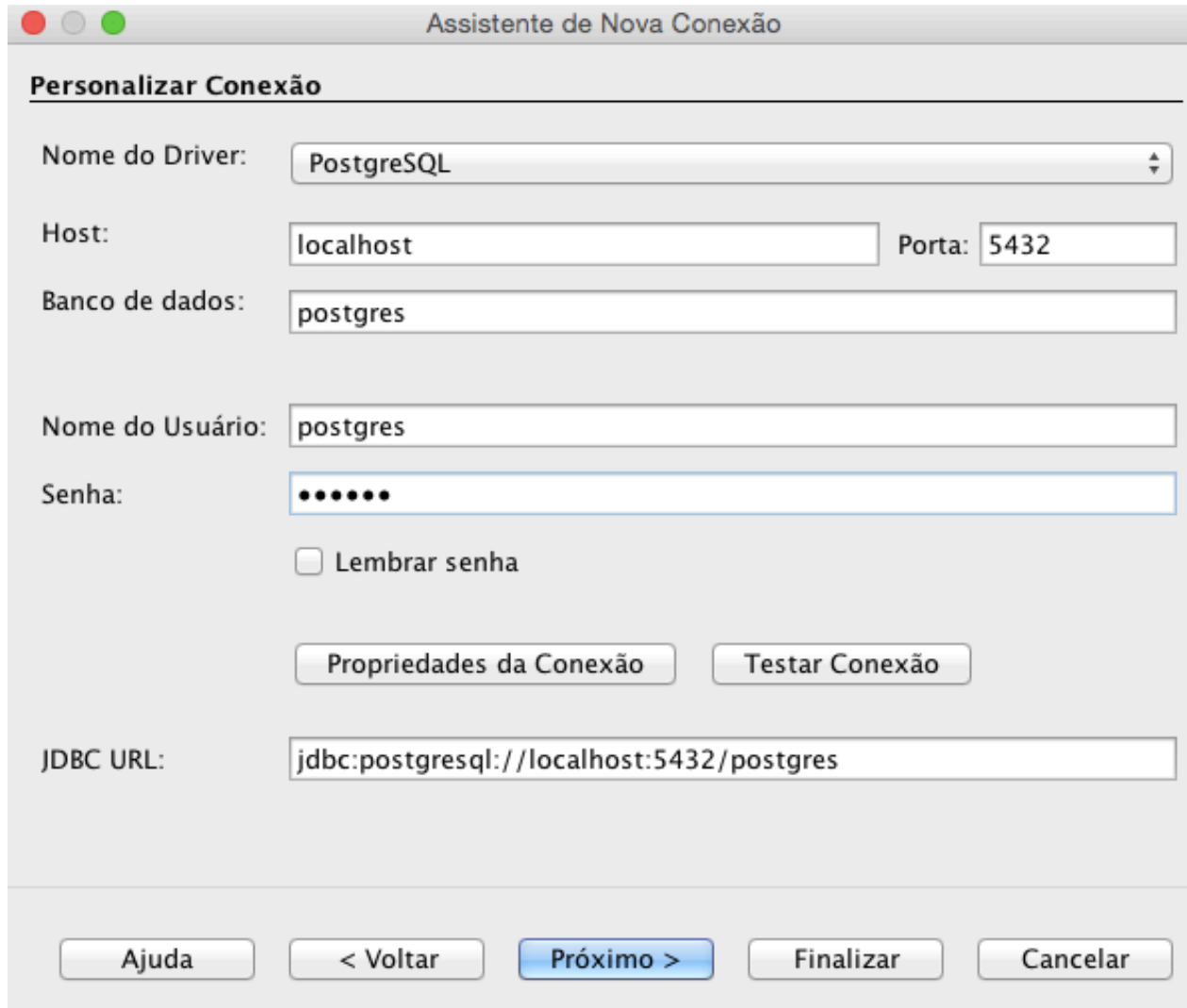
- Agora precisa adicionar o driver do bando de dados





# Criando uma conexão no NetBeans

- Precisa ter atenção neste passo!



Assistente de Nova Conexão

**Personalizar Conexão**

Nome do Driver: PostgreSQL

Host: localhost Porta: 5432

Banco de dados: postgres

Nome do Usuário: postgres

Senha: ••••••

☐ Lembrar senha

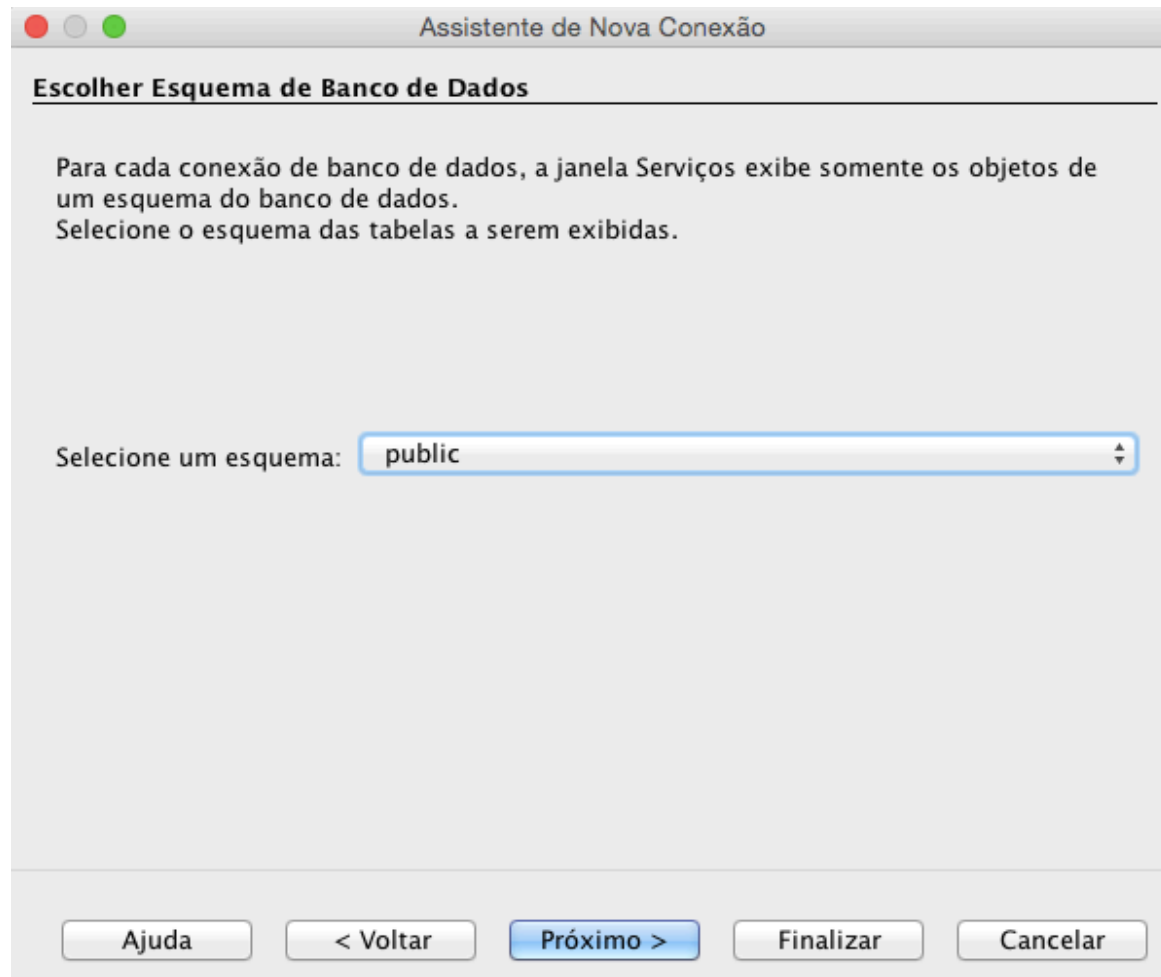
Propriedades da Conexão Testar Conexão

JDBC URL: jdbc:postgresql://localhost:5432/postgres

Ajuda < Voltar **Próximo >** Finalizar Cancelar

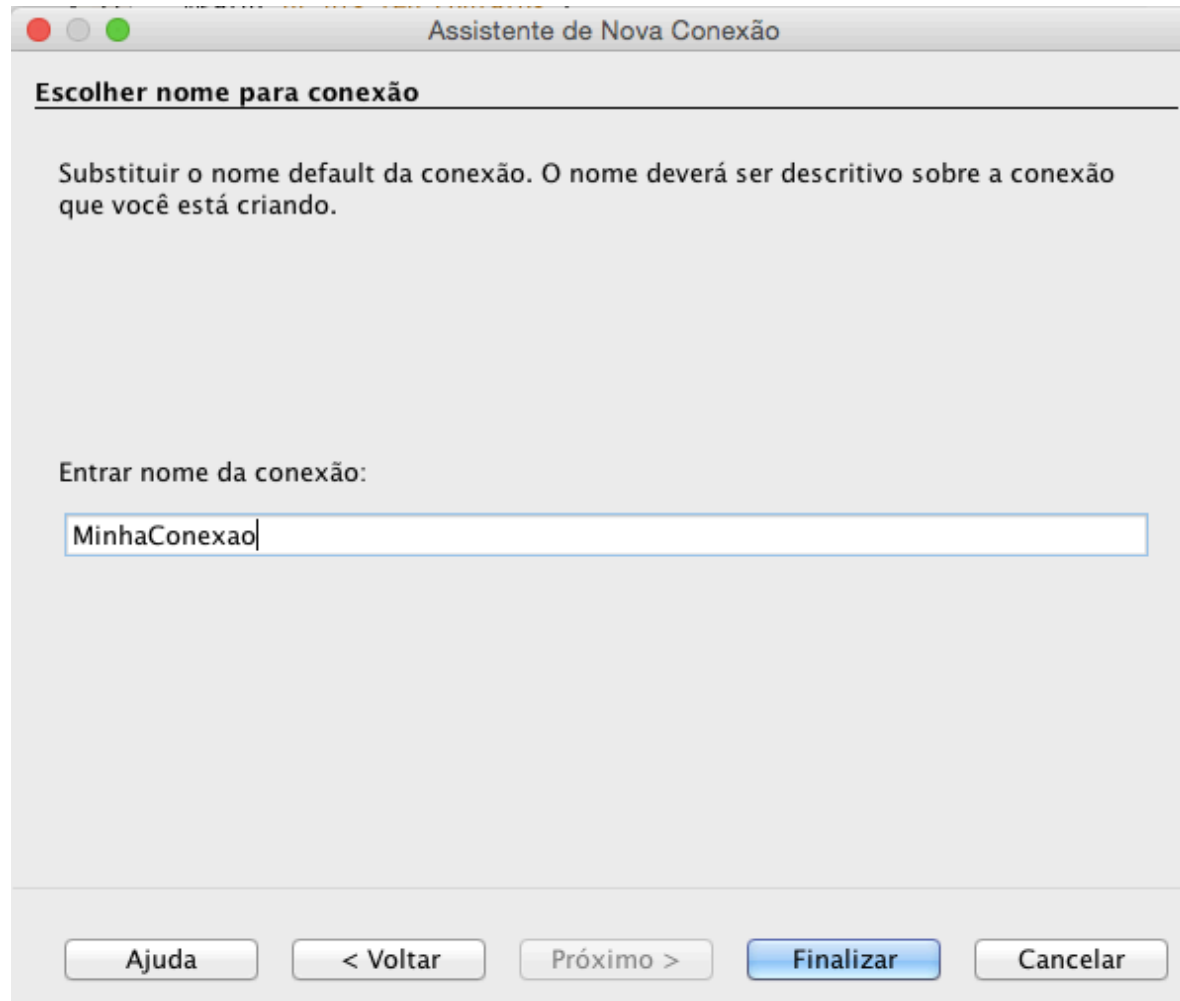
# Criando uma conexão no NetBeans

- Depois precisa escolher qual o esquema do banco de dados que será utilizado



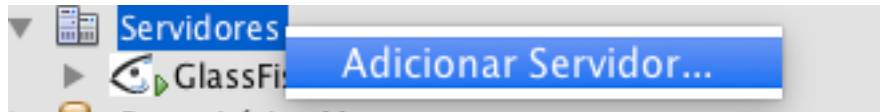
# Criando uma conexão no NetBeans

- Finalmente deve-se nomear a conexão que está sendo criada



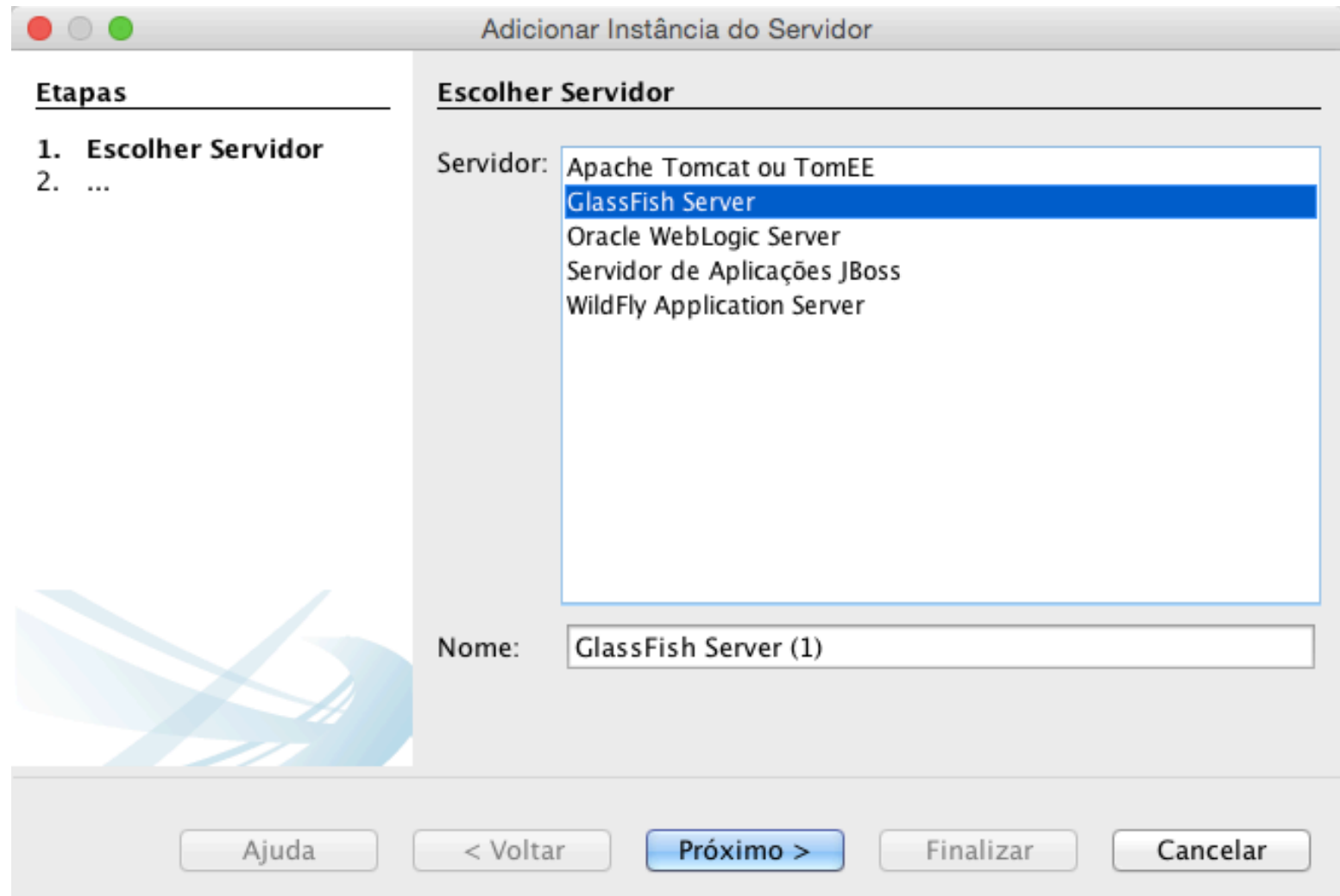
# Servidor web

- O servidor web que será utilizado neste exemplo é o GlassFish
- Temos que adicionar o GlassFish aos servidores, caso não tenha sido adicionado



# Servidor web

- Escolha o servidor



# Servidor web

- Indique onde está localizado o servidor GlassFish

**Adicionar Instância do Servidor**

**Etapas**

1. Escolher Servidor
2. **Localização do Servidor**
3. Nome/Localização do Domínio

**Localização do Servidor**

Localização de Instalação:

☒ Domínio Local ☐ Domínio Remoto

☐ Eu li e aceito o contrato de licença... (clique)

**i** domain1 já está registrado para esta instalação.  
Escolha outro domínio na próxima página.

# Servidor web

- Finalmente dê um nome ao domínio

**Adicionar Instância do Servidor**

**Etapas**

1. Escolher Servidor
2. Localização do Servidor
3. **Nome/Localização do Domínio**

**Localização do Domínio**

Domínio:

Host:

Porta DAS:  Porta HTTP:  ☐ Utilizar portas default (des:)

Destino:

Nome do Usuário:

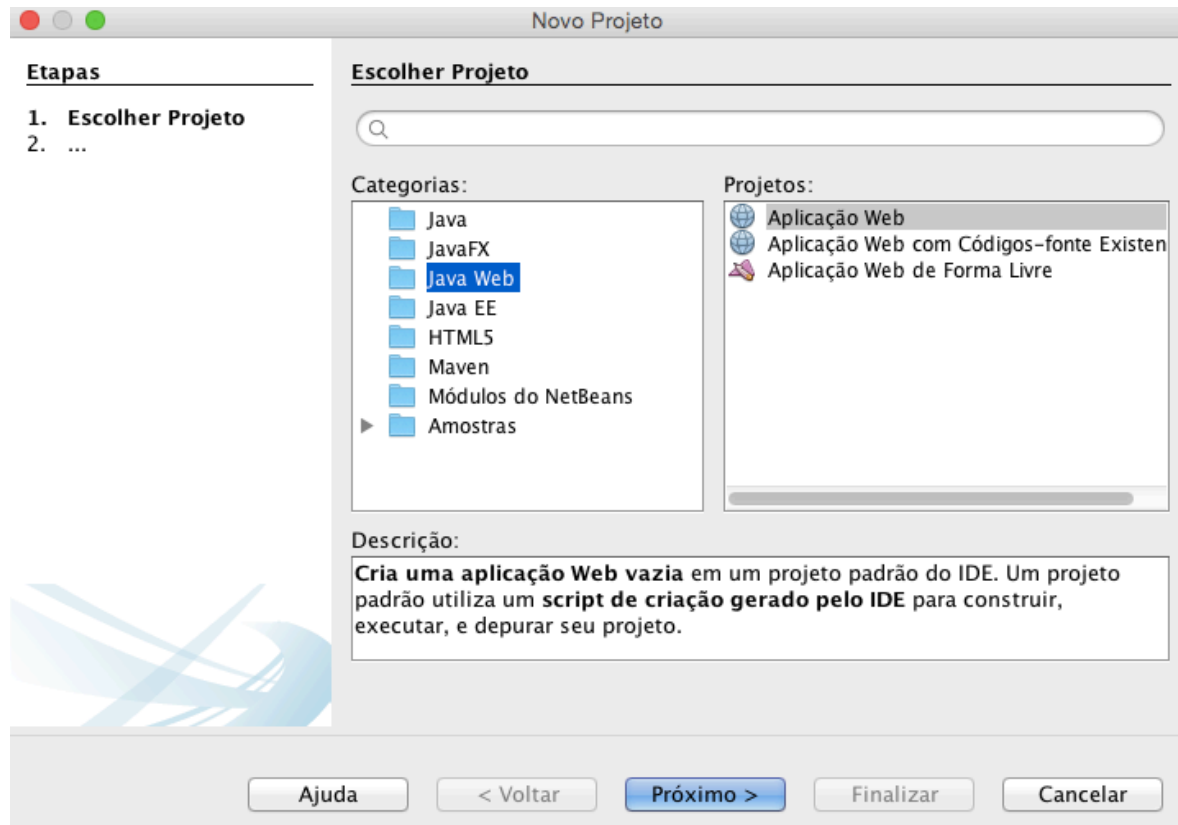
Senha:

Criar novo domínio incorporado: dominio

Ajuda < Voltar Próximo > **Finalizar** Cancelar

# Criação dos web services

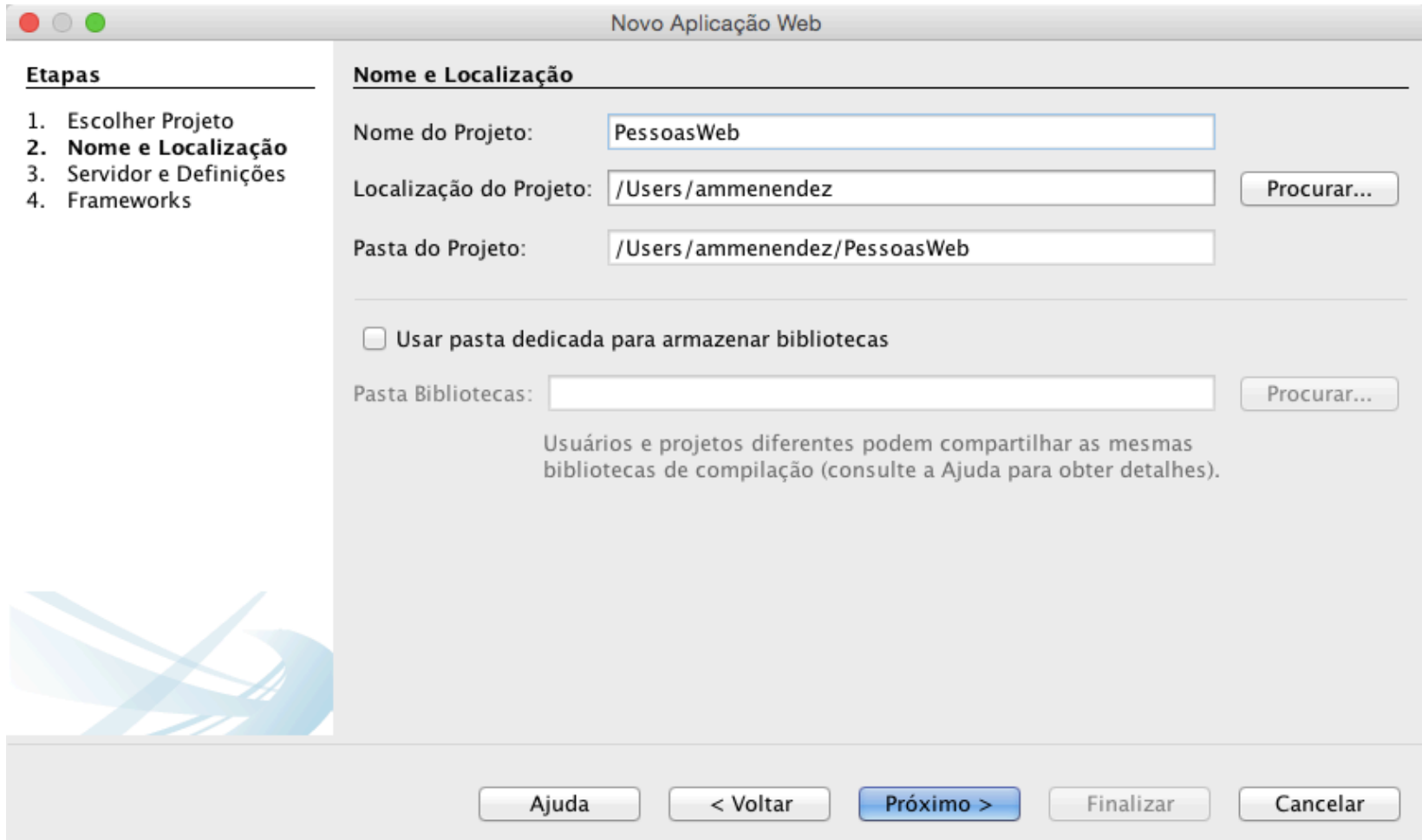
- Vamos agora criar os web services que permitirão acessar essa tabela
- Inicialmente deve ser criado um projeto web





# Criação dos web services

- Devemos dar o nome da aplicação web



Novo Aplicação Web

**Etapas**

1. Escolher Projeto
2. **Nome e Localização**
3. Servidor e Definições
4. Frameworks

**Nome e Localização**

Nome do Projeto:

Localização do Projeto:

Pasta do Projeto:

☐ Usar pasta dedicada para armazenar bibliotecas

Pasta Bibliotecas:

Usuários e projetos diferentes podem compartilhar as mesmas bibliotecas de compilação (consulte a Ajuda para obter detalhes).

# Criação dos web services

- Depois escolhe o servidor e a versão do Java

**Novo Aplicação Web**

**Etapas**

1. Escolher Projeto
2. Nome e Localização
- 3. Servidor e Definições**
4. Frameworks

**Servidor e Definições**

Adicionar à aplicação corporativa: <Nenhum(a)>

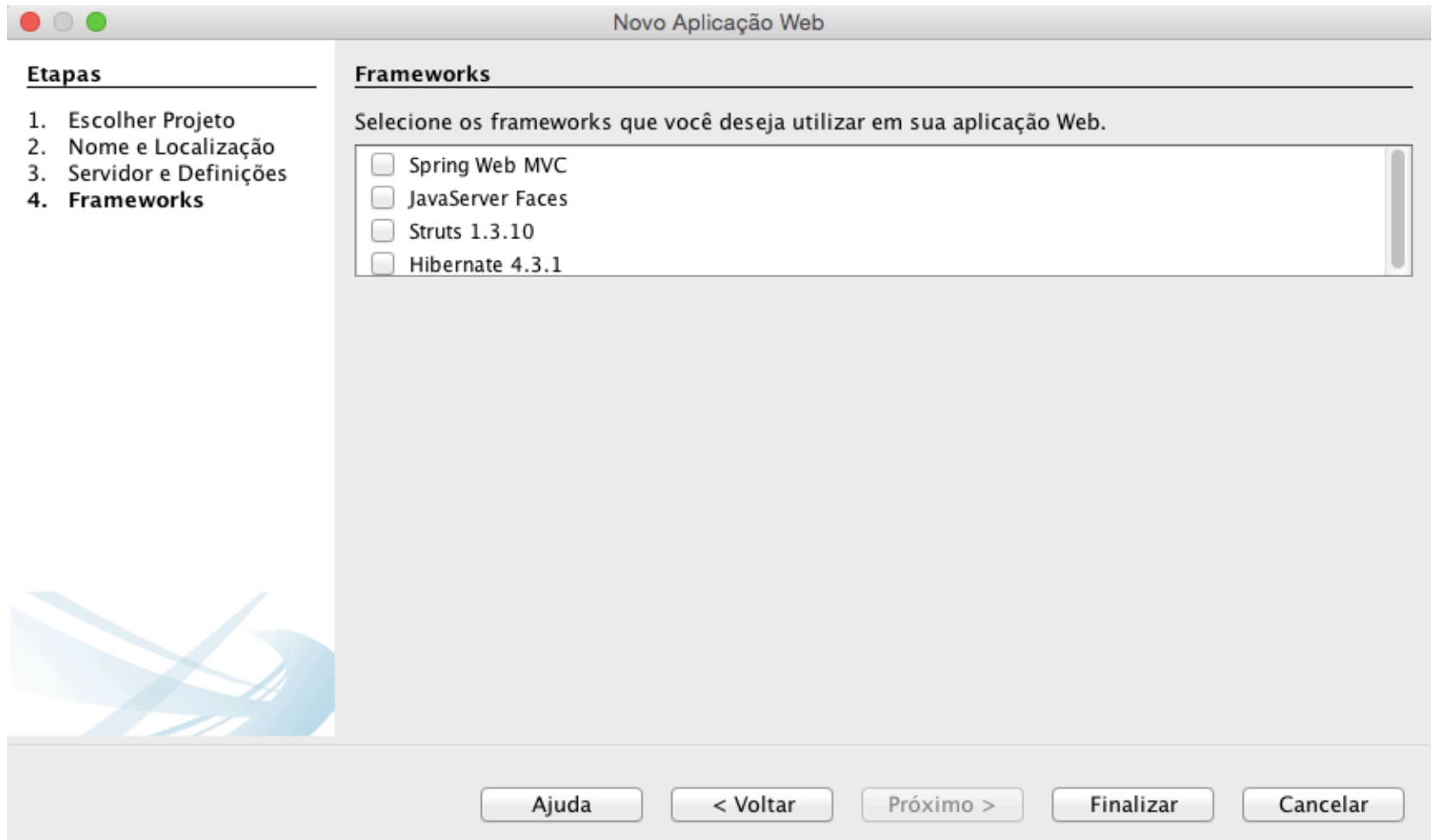
Servidor: GlassFish Server

Versão do Java EE: Java EE 7 Web

Caminho do Contexto: /PessoasWeb

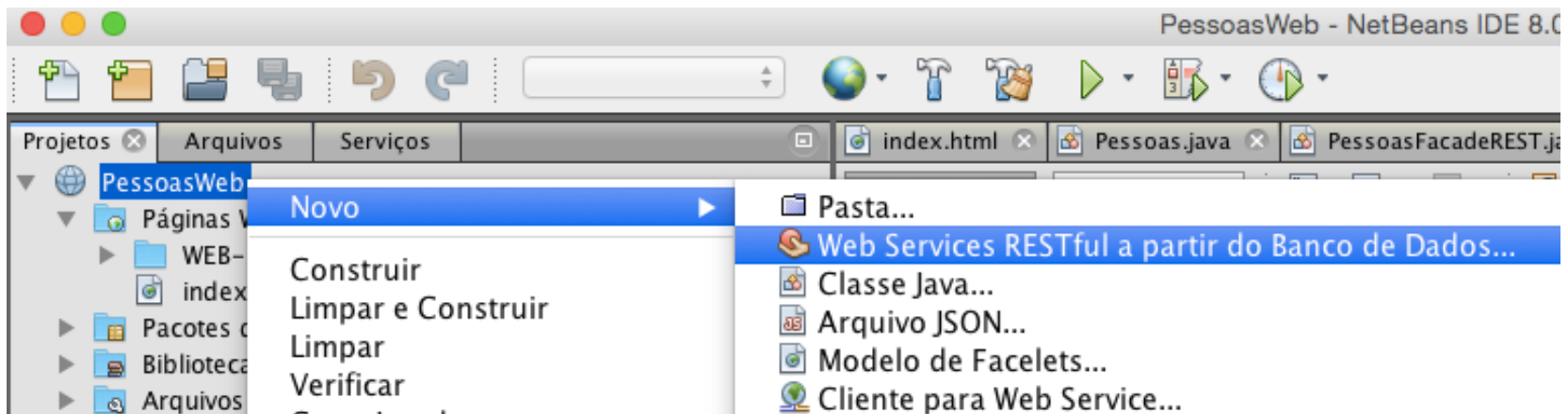
# Criação dos web services

- Não precisamos adicionar nenhum framework no passo final



# Criação de web services

- O projeto web deve ter sido criado e agora é hora de adicionar os web services
- Para isso devemos clicar com o botão direito sobre o projeto e depois Novo – Web Services RESTful a partir do Banco de Dados...



# Criação de web services

- Devemos escolher a fonte de dados do postgres que foi configurada anteriormente

New Web Services RESTful a partir do Banco de Dados

**Etapas**

1. Escolher Tipo de Arquivo
2. **Tabela de Banco de Dados**
3. Classes de Entidade
4. Classes Geradas

**Tabela de Banco de Dados**

Fonte de Dados:


Tabelas Disponíveis:

- Livros
- contatos
- pessoas

Qualquer Um

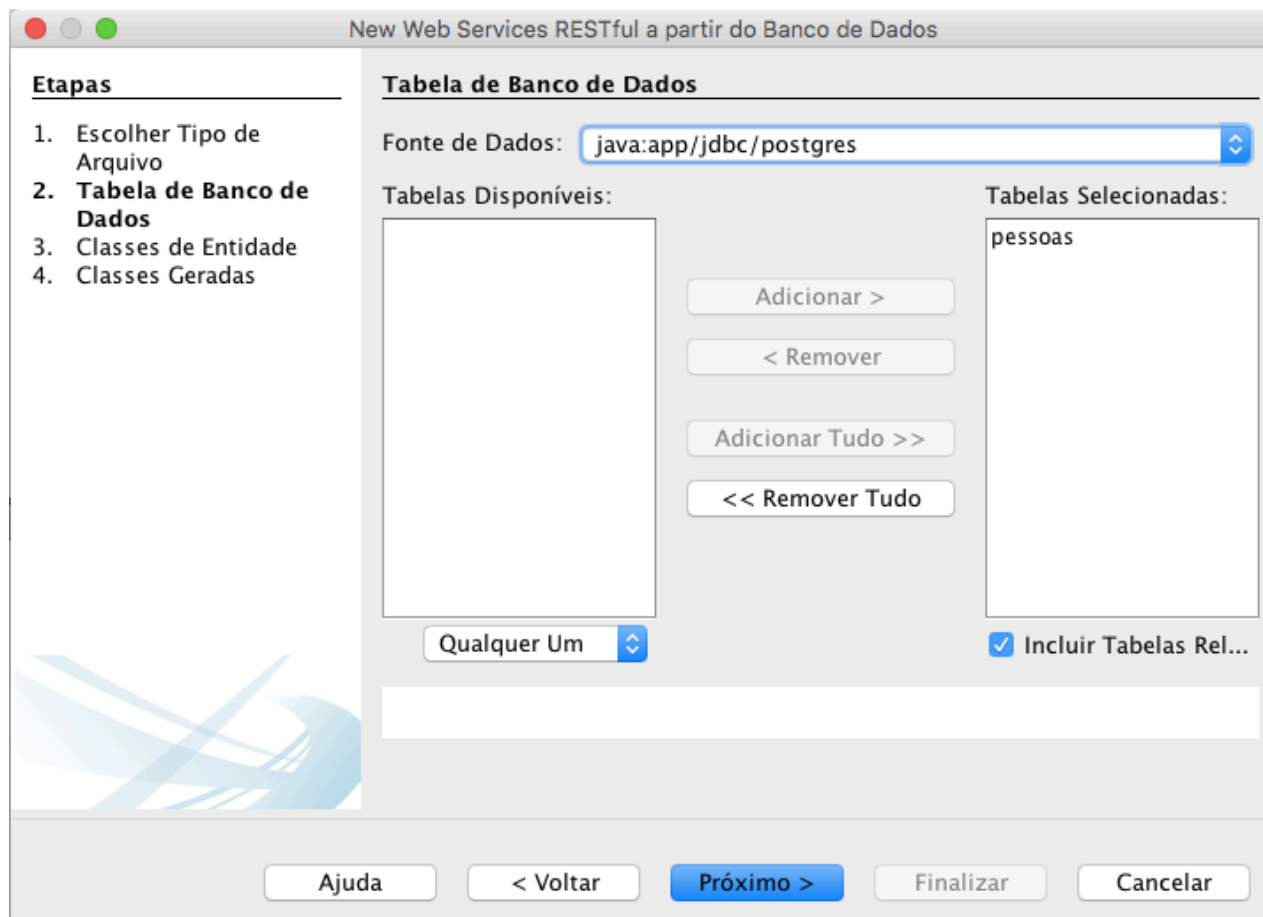
Tabelas Selecionadas:

☒ Incluir Tabelas Rel...

 **Selecione ao menos uma tabela.**

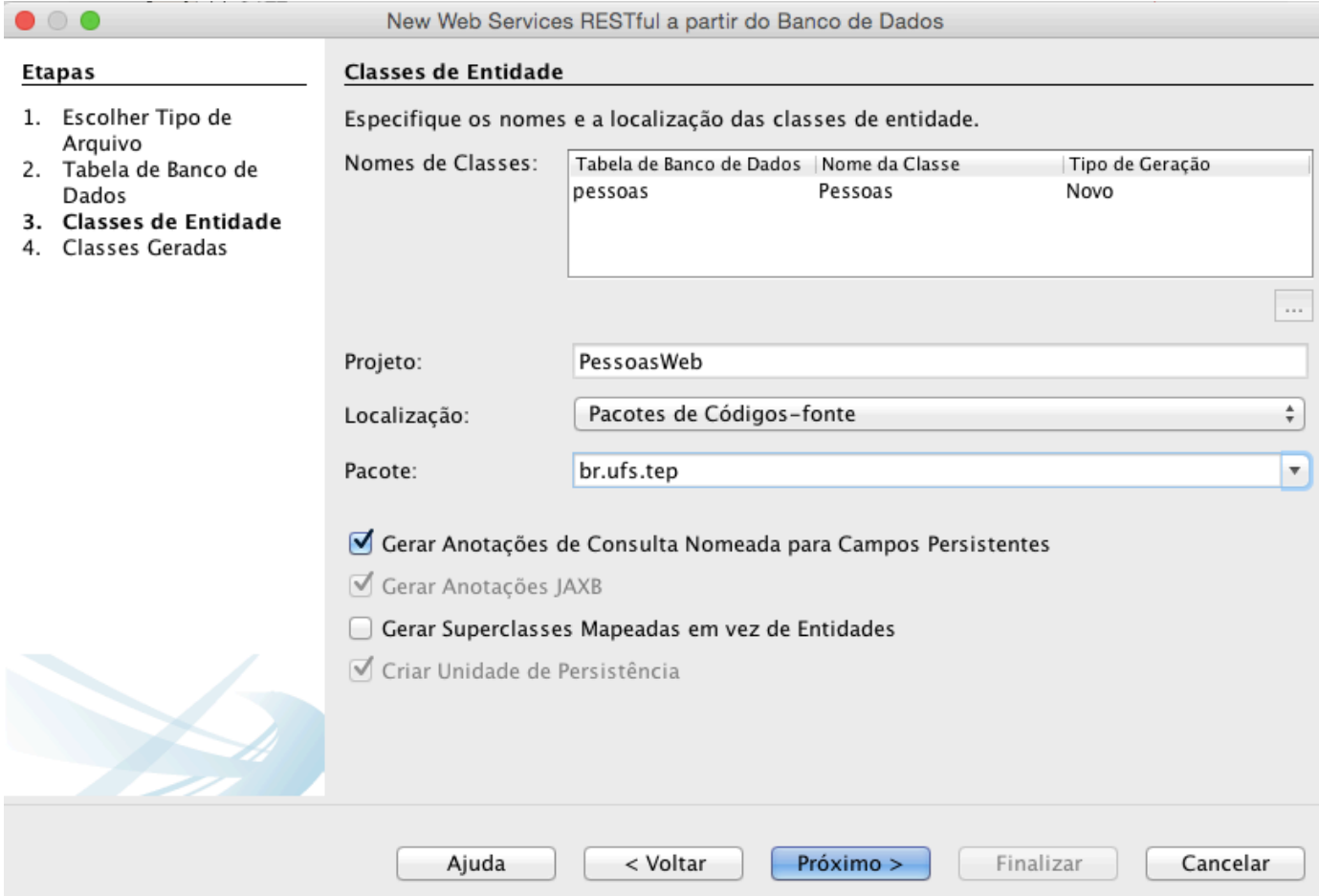
# Criação de web services

- Adicionar a tabela que desejamos criar os web services



# Criação de web services

- Agora definimos o nome do pacote onde as classes serão geradas a classe que representa a tabela



**New Web Services RESTful a partir do Banco de Dados**

**Etapas**

1. Escolher Tipo de Arquivo
2. Tabela de Banco de Dados
- 3. Classes de Entidade**
4. Classes Geradas

**Classes de Entidade**

Especifique os nomes e a localização das classes de entidade.

Nomes de Classes:

Tabela de Banco de Dados	Nome da Classe	Tipo de Geração
pessoas	Pessoas	Novo

...

Projeto:

Localização:

Pacote:

☒ Gerar Anotações de Consulta Nomeada para Campos Persistentes

☒ Gerar Anotações JAXB

☐ Gerar Superclasses Mapeadas em vez de Entidades

☒ Criar Unidade de Persistência

Ajuda < Voltar **Próximo >** Finalizar Cancelar

# Criação de web services

- Por último definimos o nome do pacote onde as outras classes serão geradas

New Web Services RESTful a partir do Banco de Dados

**Etapas**

1. Escolher Tipo de Arquivo
2. Tabela de Banco de Dados
3. Classes de Entidade
4. **Classes Geradas**

**Classes Geradas**

Projeto:

Localização:

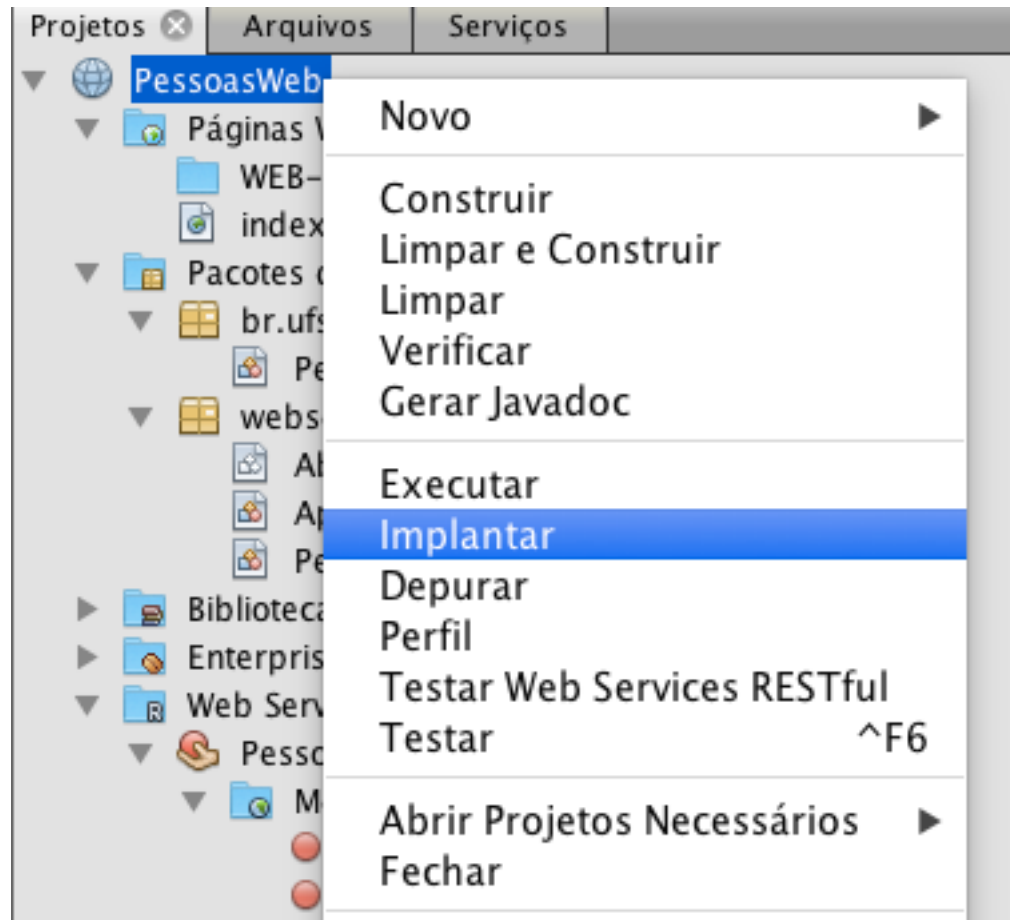
Pacote do Recurso\:

Ajuda < Voltar Próximo > Finalizar Cancelar



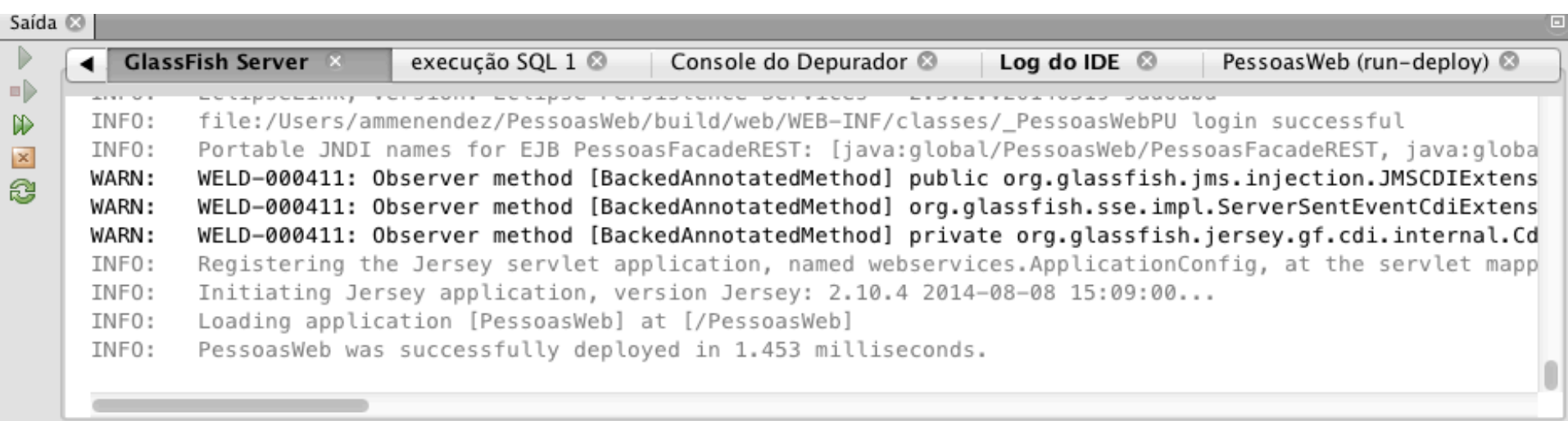
# Deploy dos web services

- Para poder usar os web services gerados precisamos fazer o deploy (implantação) do projeto



# Deploy dos web services

- Na console do GlassFish deve ser mostrado que o projeto foi implantado corretamente

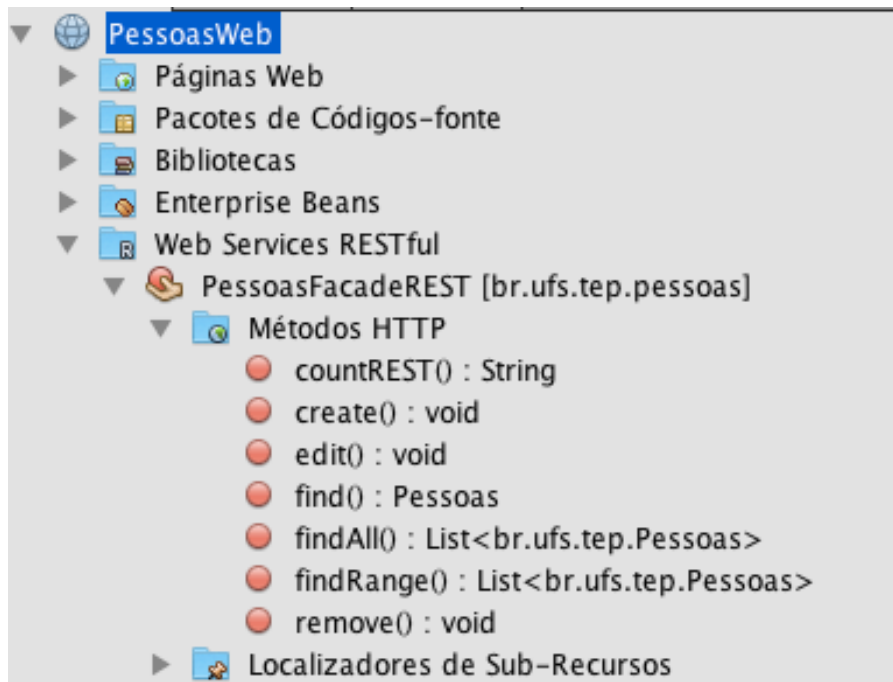


The screenshot shows the 'Saída' (Output) window of an IDE, specifically the 'GlassFish Server' tab. The console displays the following log messages:

```
INFO: file:/Users/ammenendez/PessoasWeb/build/web/WEB-INF/classes/_PessoasWebPU login successful
INFO: Portable JNDI names for EJB PessoasFacadeREST: [java:global/PessoasWeb/PessoasFacadeREST, java:globa
WARN: WELD-000411: Observer method [BackedAnnotatedMethod] public org.glassfish.jms.injection.JMSCDIExtens
WARN: WELD-000411: Observer method [BackedAnnotatedMethod] org.glassfish.sse.impl.ServerSentEventCdiExtens
WARN: WELD-000411: Observer method [BackedAnnotatedMethod] private org.glassfish.jersey.gf.cdi.internal.Cd
INFO: Registering the Jersey servlet application, named webservices.ApplicationConfig, at the servlet mapp
INFO: Initiating Jersey application, version Jersey: 2.10.4 2014-08-08 15:09:00...
INFO: Loading application [PessoasWeb] at [/PessoasWeb]
INFO: PessoasWeb was successfully deployed in 1.453 milliseconds.
```

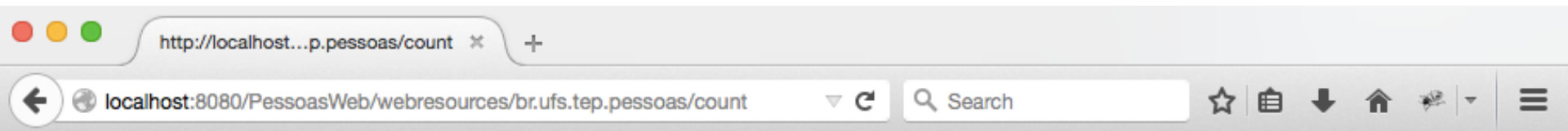
# Teste dos web services

- Podemos testar se os web services foram gerados de forma correta
- Abra o pacote Web Services RESTful e clique com o botão direito sobre o método countREST()



# Teste dos web services

- O browser será aberto e a quantidade de registros na tabela deverá ser mostrado
- Anote a URL do recurso pois ela será usada para a chamada dos web services



# A classe PessoasFacadeREST

- Nesta classe estão os métodos que podem ser chamados usando web services
- Algumas pequenas alterações podem ser efetuadas para facilitar as operações

```
@POST
@Override
@Path("inserir")
@Consumes("application/json")
public void create(Pessoas entity) {
    super.create(entity);
}
```

```
@PUT
@Path("editar/{id}")
@Consumes("application/json")
public void edit(@PathParam("id") Integer id, Pessoas entity) {
    super.edit(entity);
}
```

# A classe PessoasFacadeREST

```
@DELETE
```

```
@Path("remover/{id}")
```

```
public void remove(@PathParam("id") Integer id) {  
    super.remove(super.find(id));  
}
```

```
@GET
```

```
@Path("buscar/{id}")
```

```
@Produces("application/json")
```

```
public Pessoas find(@PathParam("id") Integer id) {  
    return super.find(id);  
}
```

```
@GET
```

```
@Override
```

```
@Produces({"application/json"})
```

```
public List<Pessoas> findAll() {  
    return super.findAll();  
}
```

# Criação de web services

- Moleza...

