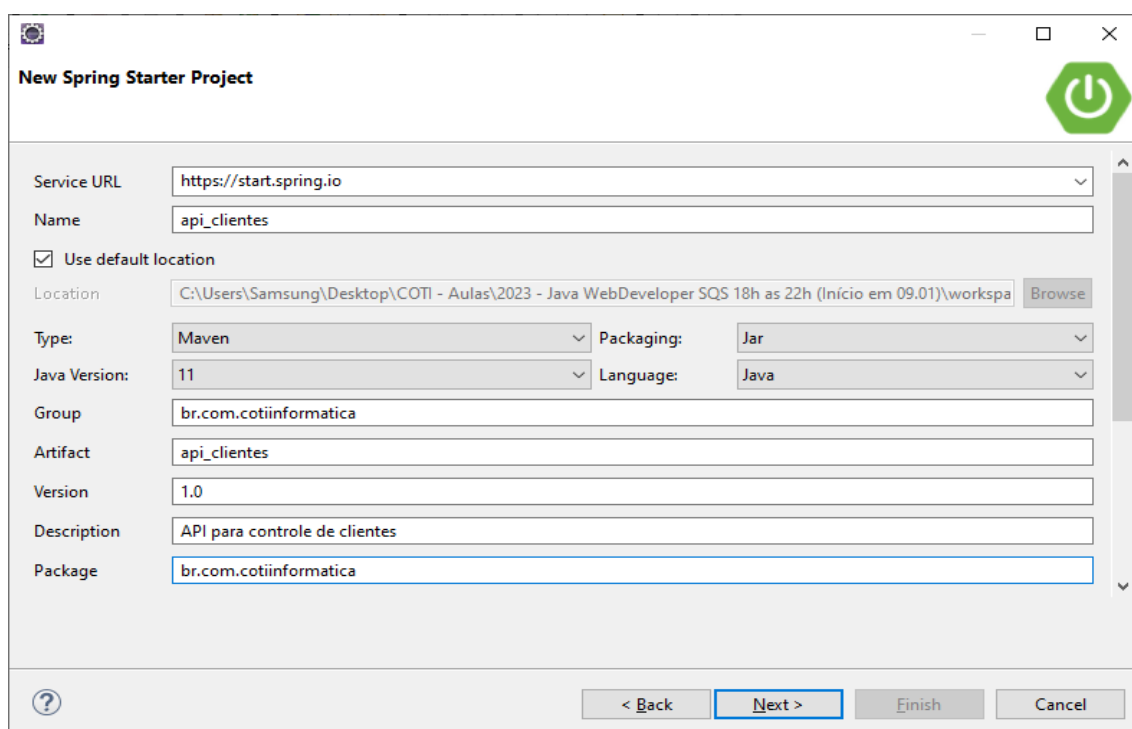
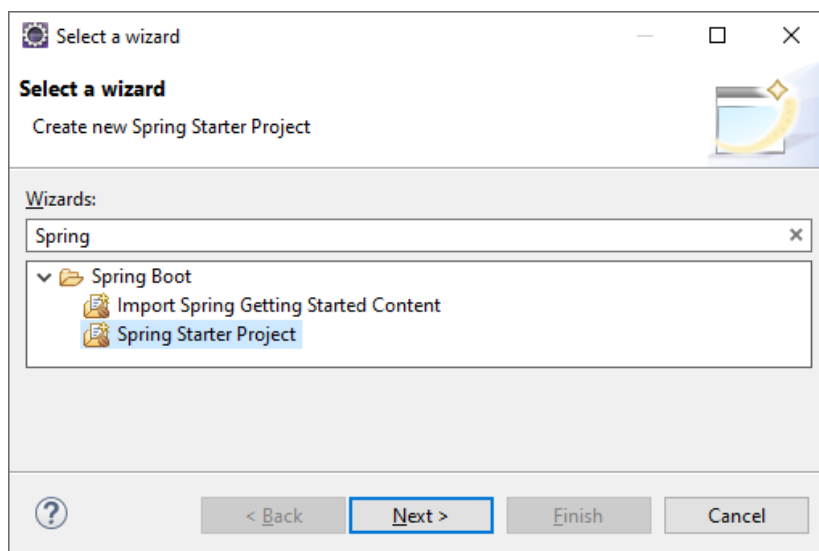
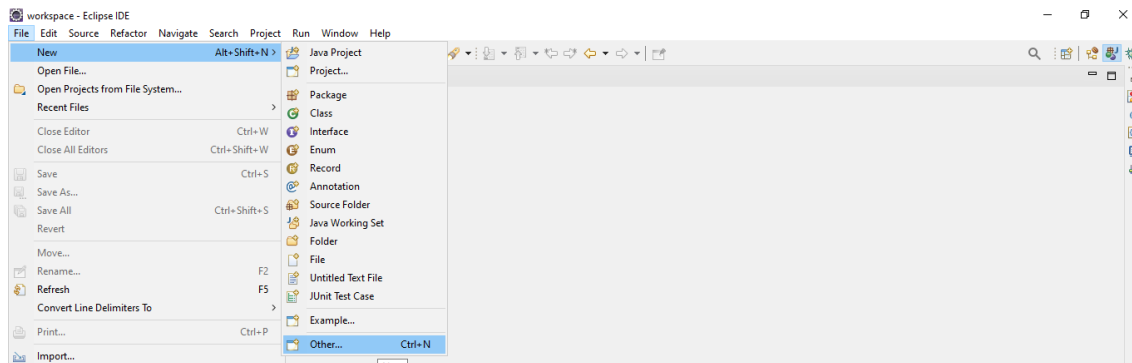
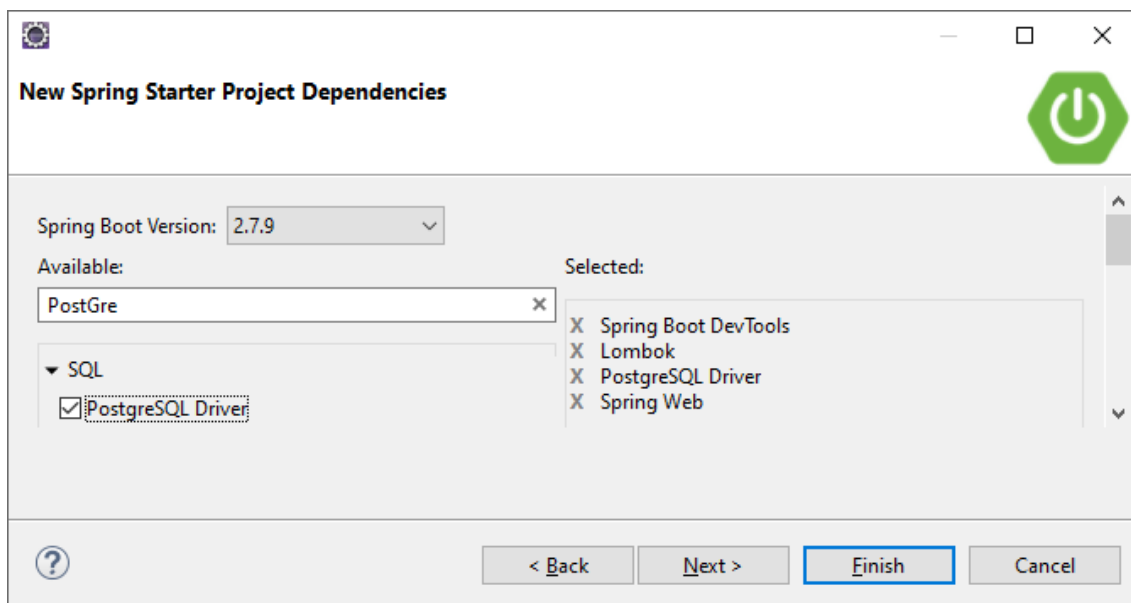


Criando um projeto Spring Boot: FILE / NEW / OTHER



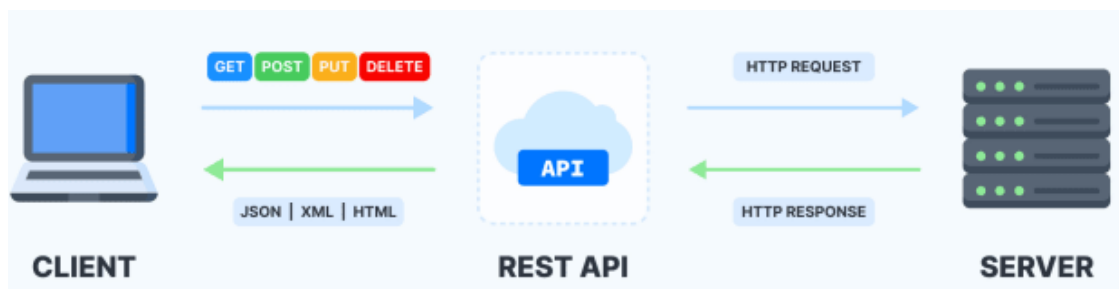


- ✓ api_clientes [boot] [devtools]
 - src/main/java
 - br.com.cotiinformatica
 - ApiClientesApplication.java
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-11]
 - Maven Dependencies
 - src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml

Nesta API, vamos desenvolver um **ENDPOINT para controle de clientes**, que permita cadastrar, consultar, editar e excluir dados de clientes.

Exemplo:

/api/clientes

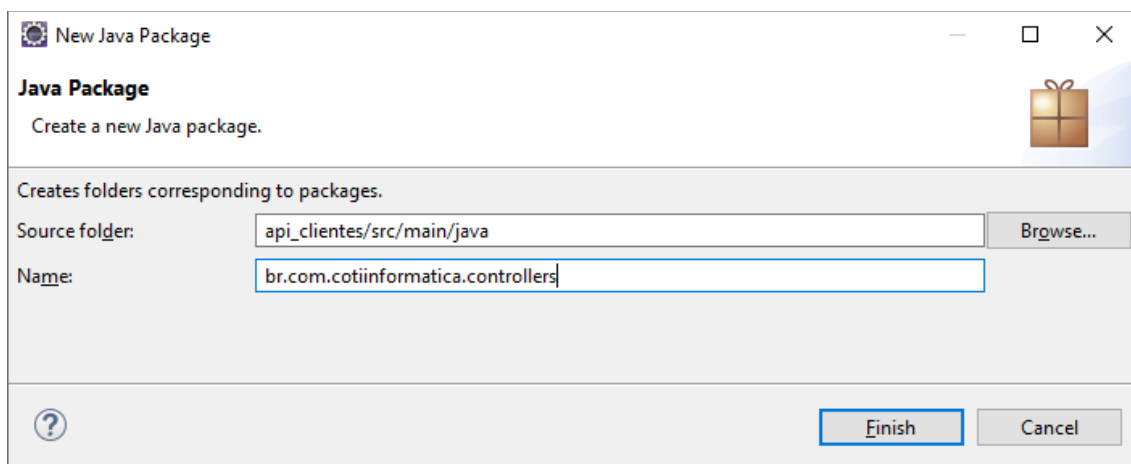


Métodos:

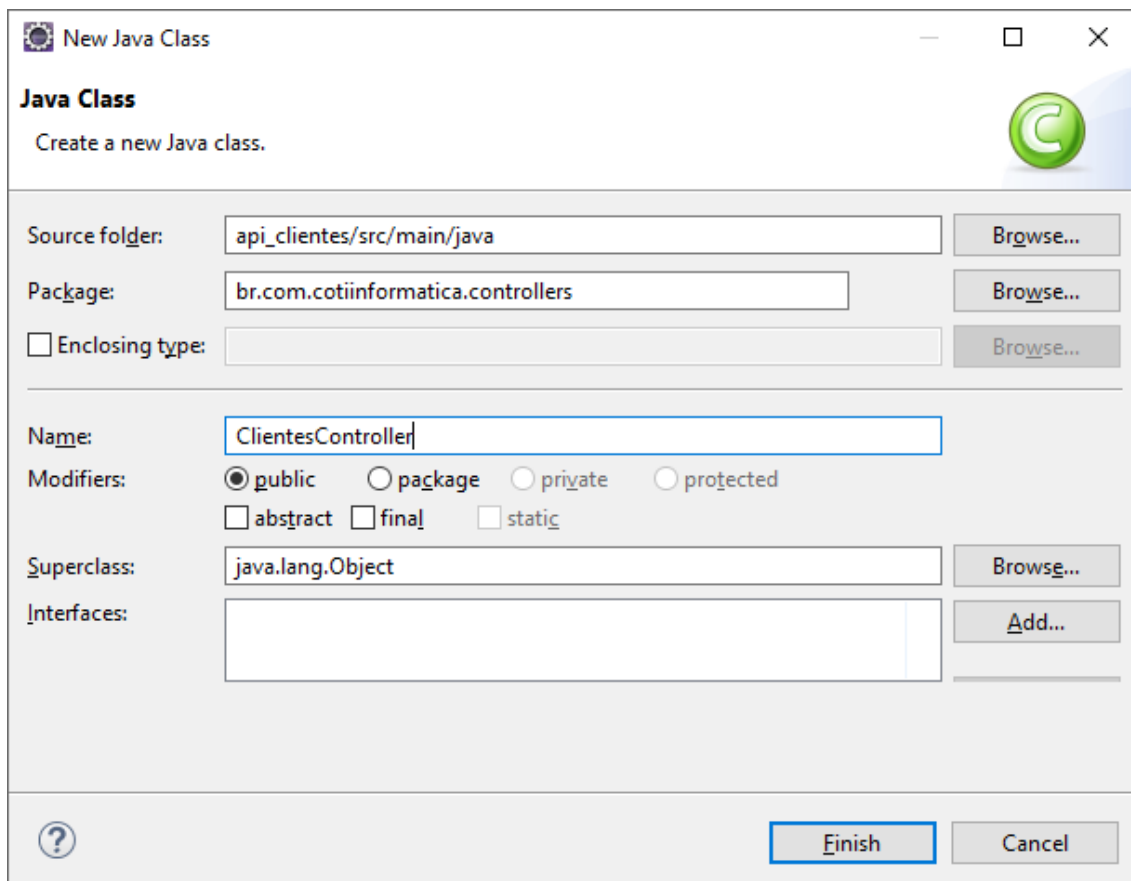
- **POST** /api/clientes (Serviço para cadastro de clientes)
- **PUT** /api/clientes (Serviço para edição de clientes)
- **DELETE** /api/clientes (Serviço para exclusão de clientes)
- **GET** /api/clientes (Serviço para consulta de clientes)

Para isso, vamos começar criando um controlador:

/controllers/**ClientesController.java**



The 'New Java Package' dialog box is shown. It has a title bar with a question mark icon and standard window controls. The main title is 'Java Package' with a gift icon. Below the title, it says 'Create a new Java package.' and 'Creates folders corresponding to packages.' There are two input fields: 'Source folder:' with the value 'api_clientes/src/main/java' and a 'Browse...' button, and 'Name:' with the value 'br.com.cotiinformatica.controllers'. At the bottom, there is a 'Finish' button and a 'Cancel' button.



The 'New Java Class' dialog box is shown. It has a title bar with a question mark icon and standard window controls. The main title is 'Java Class' with a green 'C' icon. Below the title, it says 'Create a new Java class.' There are three input fields: 'Source folder:' with the value 'api_clientes/src/main/java' and a 'Browse...' button, 'Package:' with the value 'br.com.cotiinformatica.controllers' and a 'Browse...' button, and 'Enclosing type:' which is empty and has a 'Browse...' button. Below these, there is a 'Name:' field with the value 'ClientesController'. There are four radio buttons for 'Modifiers': 'public' (selected), 'package', 'private', and 'protected'. There are three checkboxes: 'abstract', 'final', and 'static', all of which are unchecked. There is a 'Superclass:' field with the value 'java.lang.Object' and a 'Browse...' button. There is an 'Interfaces:' field which is empty and has an 'Add...' button. At the bottom, there is a 'Finish' button and a 'Cancel' button.

@RestController

Annotation para mapear um classe do projeto Spring Boot definindo-a como um controlador de API que utiliza o padrão REST.

```
package br.com.cotiinformatica.controllers;
```

```
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
public class ClientesController {

}
```

Criando os métodos do controlador:

Definindo os ENDPOINTS da API.

```
package br.com.cotiinformatica.controllers;
```

```
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
public class ClientesController {

    @PostMapping("/api/clientes")
    public ResponseEntity<String> post() {
        return null;
    }

    @PutMapping("/api/clientes")
    public ResponseEntity<String> put() {
        return null;
    }

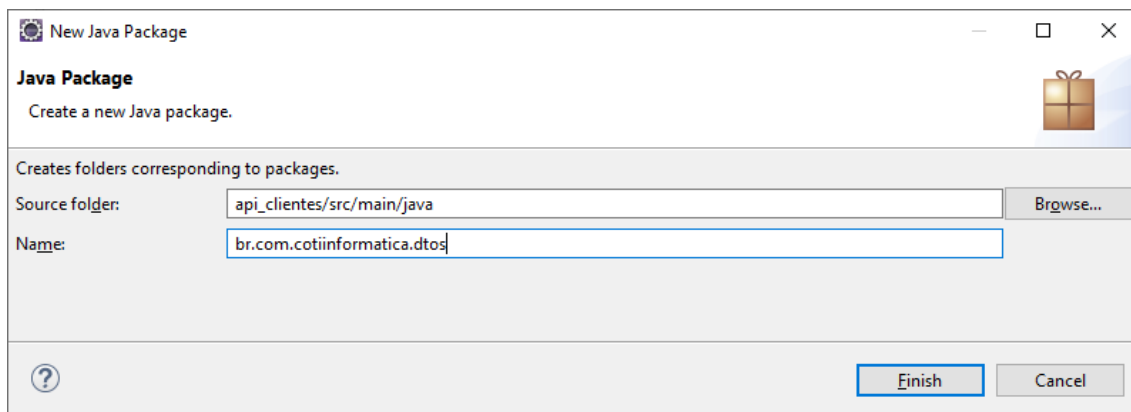
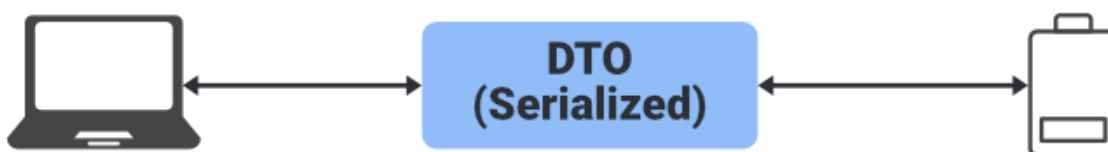
    @DeleteMapping("/api/clientes")
    public ResponseEntity<String> delete() {
        return null;
    }

    @GetMapping("/api/clientes")
    public ResponseEntity<String> getAll() {
        return null;
    }
}
```

```
@GetMapping("/api/clientes")
public ResponseEntity<String> getById() {
    return null;
}
```

DTOs – Data Transfer Objects

Criando os objetos para transferência de dados com a finalidade de modelar os dados de requisição e resposta da API.



```
package br.com.cotiinformatica.dtos;
```

```
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import lombok.ToString;
```

```
@Setter
@Getter
@NoArgsConstructor
@AllArgsConstructor
@ToString
public class PostClientesDTO {

    private String nome;
    private String cpf;
    private String email;
    private String telefone;
}
```

```
package br.com.cotiinformatica.dtos;
```

```
import lombok.AllArgsConstructor;  
import lombok.Getter;  
import lombok.NoArgsConstructor;  
import lombok.Setter;  
import lombok.ToString;
```

```
@Setter
```

```
@Getter
```

```
@NoArgsConstructor
```

```
@AllArgsConstructor
```

```
@ToString
```

```
public class PutClientesDTO {
```

```
    private Integer idCliente;
```

```
    private String nome;
```

```
    private String cpf;
```

```
    private String email;
```

```
    private String telefone;
```

```
}
```

```
package br.com.cotiinformatica.dtos;
```

```
import java.util.Date;
```

```
import lombok.AllArgsConstructor;
```

```
import lombok.Getter;
```

```
import lombok.NoArgsConstructor;
```

```
import lombok.Setter;
```

```
import lombok.ToString;
```

```
@Setter
```

```
@Getter
```

```
@NoArgsConstructor
```

```
@AllArgsConstructor
```

```
@ToString
```

```
public class GetClientesDTO {
```

```
    private Integer idCliente;
```

```
    private String nome;
```

```
    private String cpf;
```

```
    private String email;
```

```
    private String telefone;
```

```
    private Date dataCadastro;
```

```
}
```

Voltando no controlador:

/controllers/**CientesController.java**

```
package br.com.cotiinformatica.controllers;
```

```
import java.util.List;
```

```
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.DeleteMapping;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.PutMapping;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RestController;
```

```
import br.com.cotiinformatica.dtos.GetClientesDTO;  
import br.com.cotiinformatica.dtos.PostClientesDTO;  
import br.com.cotiinformatica.dtos.PutClientesDTO;
```

```
@RestController
```

```
public class CientesController {
```

```
    @PostMapping("/api/clientes")  
    public ResponseEntity<GetClientesDTO> post  
        (@RequestBody PostClientesDTO dto) {  
        return null;  
    }
```

```
    @PutMapping("/api/clientes")  
    public ResponseEntity<GetClientesDTO> put  
        (@RequestBody PutClientesDTO dto) {  
        return null;  
    }
```

```
    @DeleteMapping("/api/clientes/{id}")  
    public ResponseEntity<GetClientesDTO> delete  
        (@PathVariable("id") Integer idCliente) {  
        return null;  
    }
```

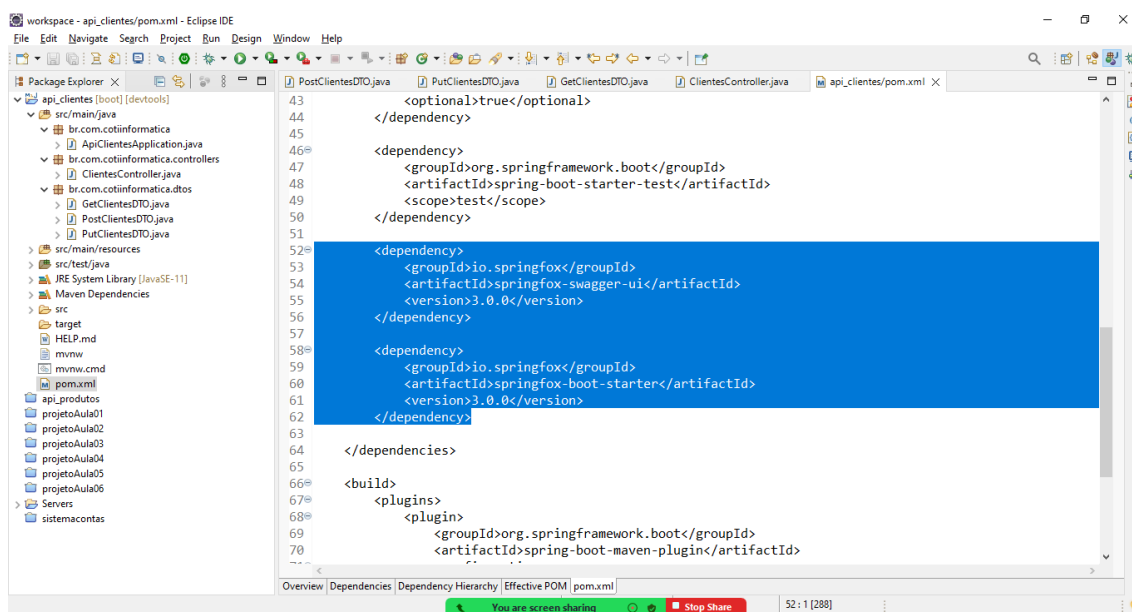
```
    @GetMapping("/api/clientes")  
    public ResponseEntity<List<GetClientesDTO>> getAll() {  
        return null;  
    }
```

```
    @GetMapping("/api/clientes/{id}")  
    public ResponseEntity<GetClientesDTO> getById  
        (@PathVariable("id") Integer idCliente) {  
        return null;  
    }
```

```
}
```

Swagger

Gerando a documentação mostrando os ENDPOINTS e métodos da API.
Incluindo no **pom.xml**



<dependency>

<groupId>**io.springfox**</groupId>

<artifactId>**springfox-swagger-ui**</artifactId>

<version>**3.0.0**</version>

</dependency>

<dependency>

<groupId>**io.springfox**</groupId>

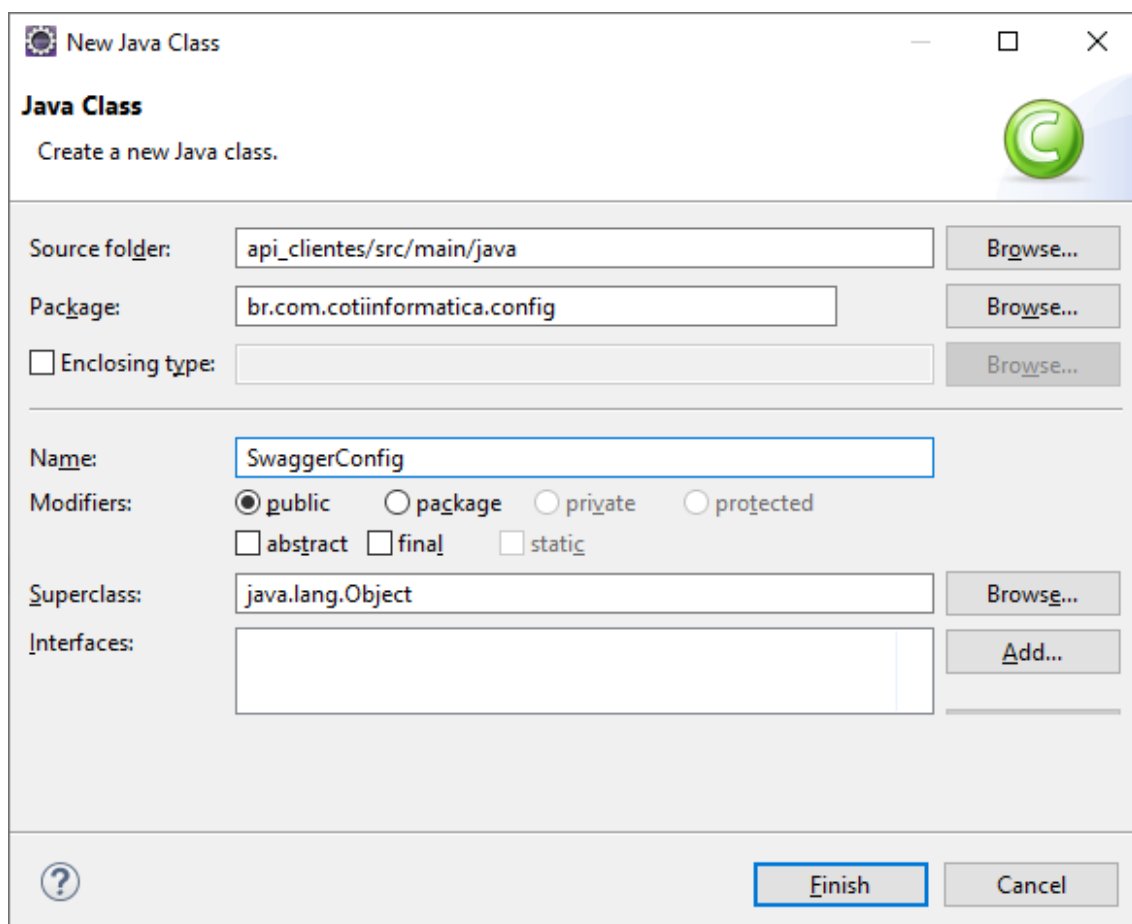
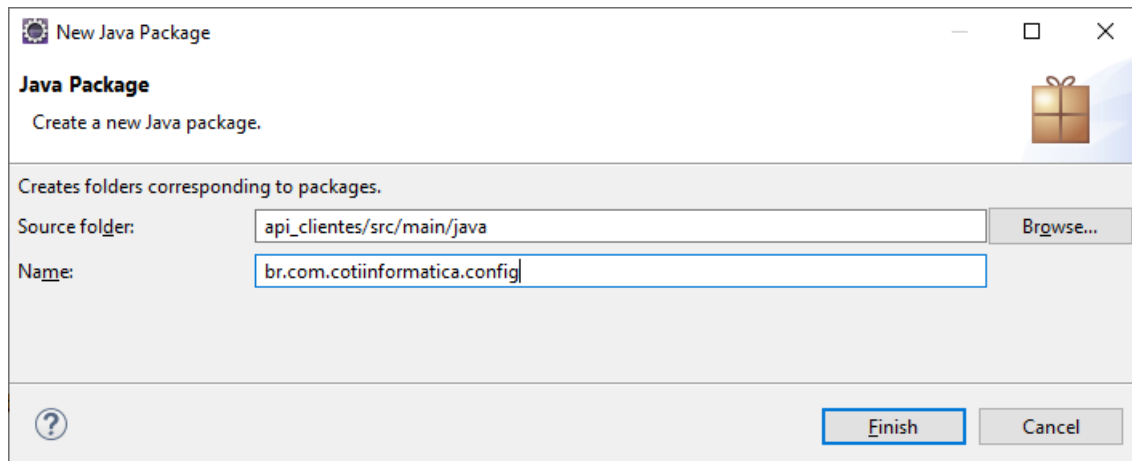
<artifactId>**springfox-boot-starter**</artifactId>

<version>**3.0.0**</version>

</dependency>

Criando a classe de configuração para o Swagger:

/config/**SwaggerConfig.java**



```
package br.com.cotiinformatica.config;
```

```
import java.util.Collections;
```

```
import org.springframework.context.annotation.Bean;
```

```
import org.springframework.context.annotation.Configuration;
```

```
import org.springframework.web.servlet.config.annotation.EnableWebMvc;

import springfox.documentation.builders.PathSelectors;
import springfox.documentation.builders.RequestHandlerSelectors;
import springfox.documentation.service.ApiInfo;
import springfox.documentation.service.Contact;
import springfox.documentation.spi.DocumentationType;
import springfox.documentation.spring.web.plugins.Docket;
import springfox.documentation.swagger2.annotations.EnableSwagger2;

@Configuration
@EnableWebMvc
@EnableSwagger2
public class SwaggerConfig {

    @Bean
    public Docket api() {

        return new Docket(DocumentationType.SWAGGER_2)
            .select()
            .apis(RequestHandlerSelectors
                .basePackage("br.com.cotiinformatica"))
            .paths(PathSelectors.ant("/**"))
            .build()
            .apiInfo(apiInfo());
    }

    private ApiInfo apiInfo() {

        return new ApiInfo(
            "API para controle de clientes",
            "Sistema Spring Boot API",
            "Versão 1.0",
            "http://www.cotiinformatica.com.br",
            new Contact("COTI Informática",
                "http://www.cotiinformatica.com.br",
                "contato@cotiinformatica.com.br"),
            "Licença da API",
            "http://www.cotiinformatica.com.br",
            Collections.emptyList()
        );
    }
}
```

Voltando no controlador e mapear comentários que descrevam as operações da API:

```
package br.com.cotiinformatica.controllers;
```

```
import java.util.List;
```

```
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.DeleteMapping;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.PutMapping;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RestController;
```

```
import br.com.cotiinformatica.dtos.GetClientesDTO;  
import br.com.cotiinformatica.dtos.PostClientesDTO;  
import br.com.cotiinformatica.dtos.PutClientesDTO;  
import io.swagger.annotations.Api;  
import io.swagger.annotations.ApiOperation;
```

@Api(tags = "Clientes")

@RestController

```
public class ClientesController {
```

@ApiOperation("Serviço para cadastro de clientes.")

@PostMapping("/api/clientes")

```
public ResponseEntity<GetClientesDTO> post  
    (@RequestBody PostClientesDTO dto) {  
    return null;  
}
```

@ApiOperation("Serviço para edição de clientes.")

@PutMapping("/api/clientes")

```
public ResponseEntity<GetClientesDTO> put  
    (@RequestBody PutClientesDTO dto) {  
    return null;  
}
```

@ApiOperation("Serviço para exclusão de clientes.")

@DeleteMapping("/api/clientes/{id}")

```
public ResponseEntity<GetClientesDTO> delete  
    (@PathVariable("id") Integer idCliente) {  
    return null;  
}
```

@ApiOperation("Serviço para consulta de clientes.")

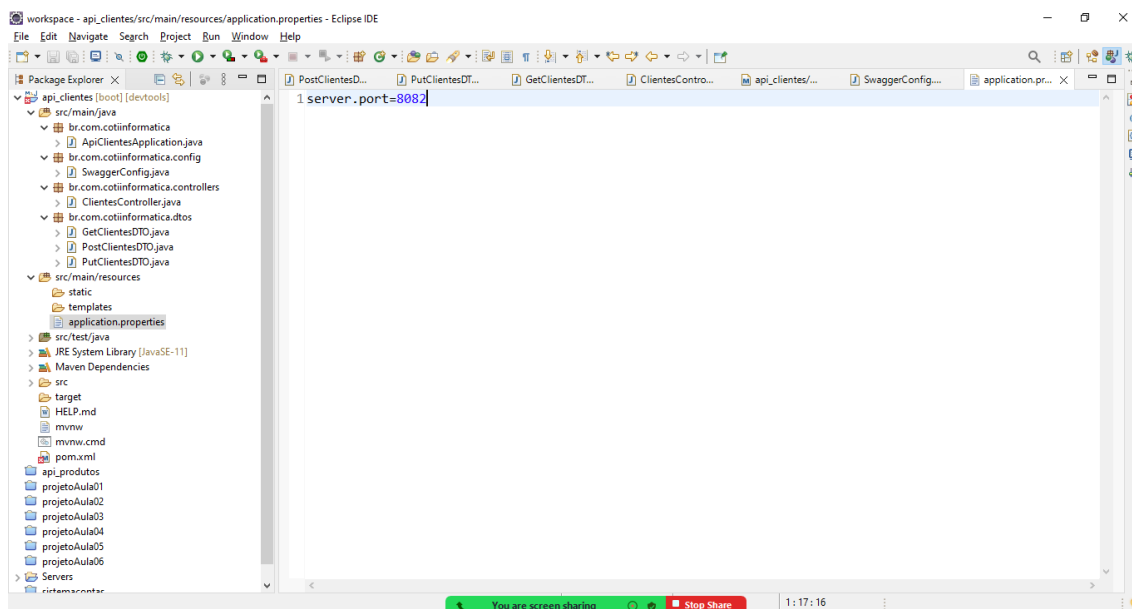
@GetMapping("/api/clientes")

```
public ResponseEntity<List<GetClientesDTO>> getAll() {  
    return null;  
}
```

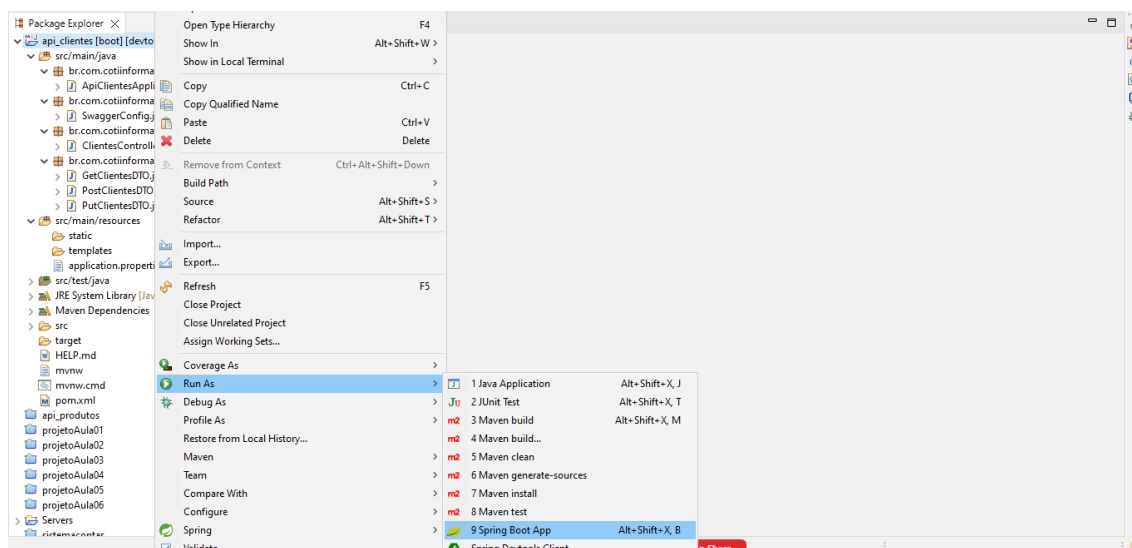
```
@ApiOperation("Serviço para consulta de cliente por id.")
@GetMapping("/api/clientes/{id}")
public ResponseEntity<GetClientesDTO> getById
    (@PathVariable("id") Integer idCliente) {
    return null;
}
}
```

Por último, vamos configurar a porta do TOMCAT para executar o projeto:
/application.properties

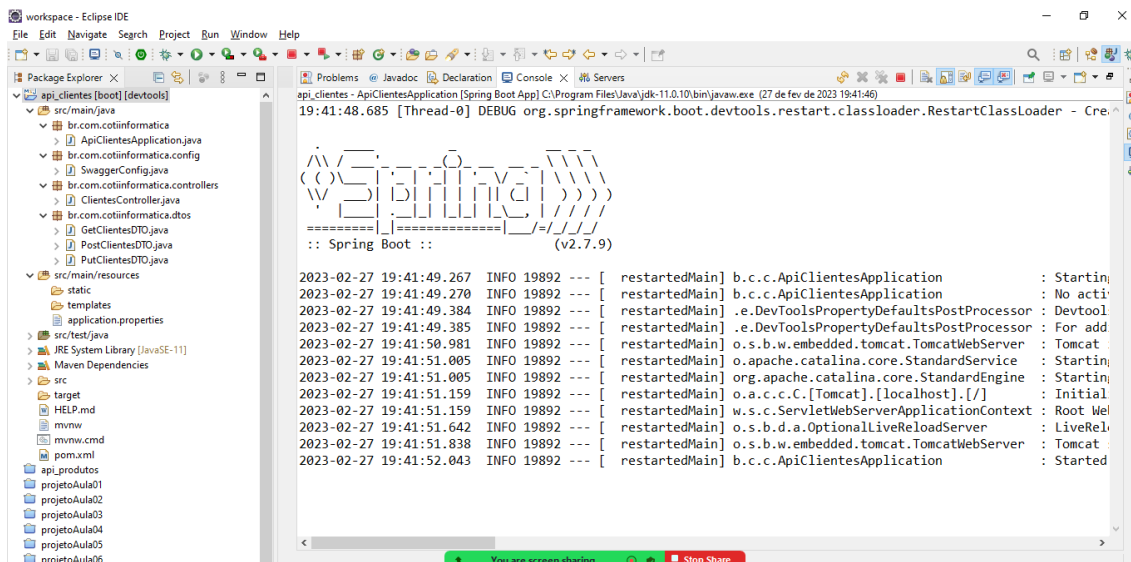
server.port=8082



Executando:



Executando:



```

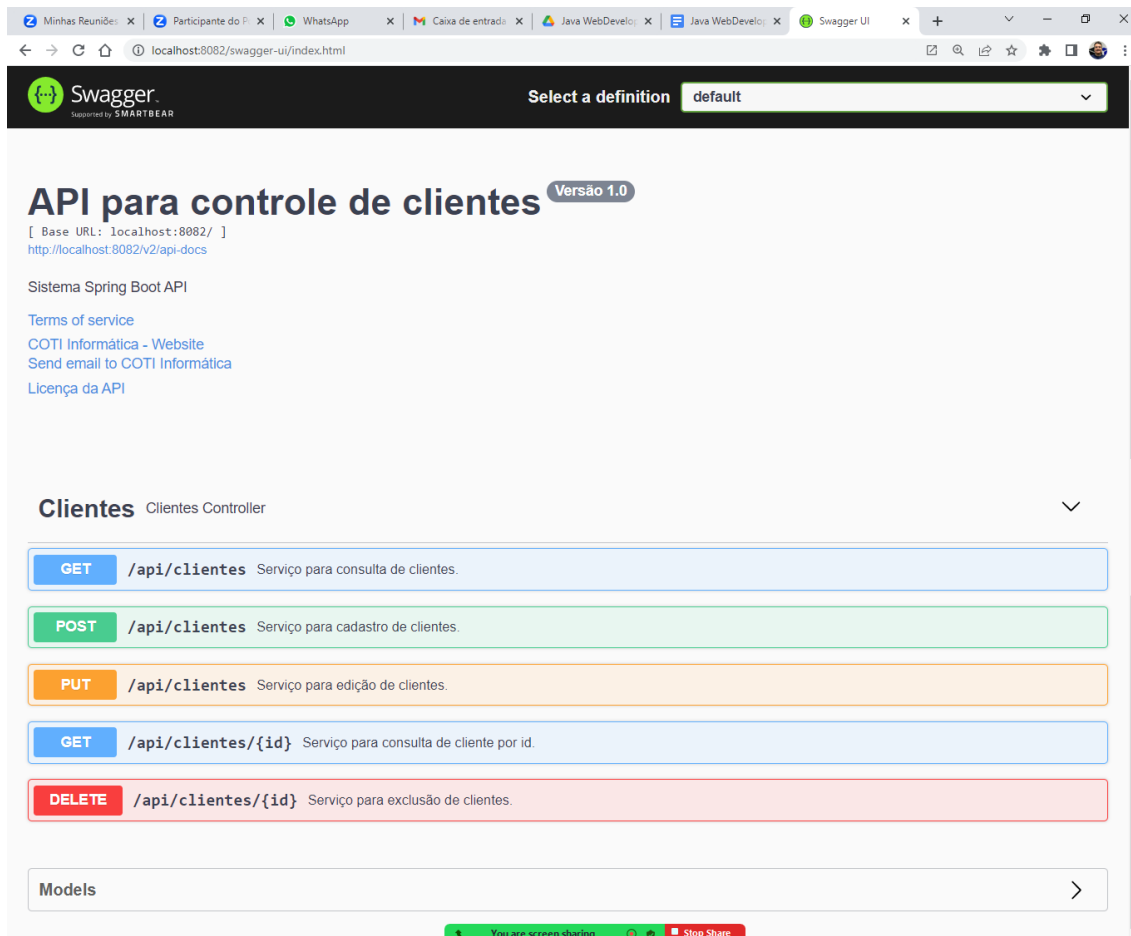
workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Package Explorer X
api_clientes [boot] [devtools]
  src/main/java
    br.com.cotiinformatica
      ApiClientesApplication.java
      ApiClientesApplicationConfig.java
      SwaggerConfig.java
      br.com.cotiinformatica.controllers
        ClientesController.java
      br.com.cotiinformatica.dtos
        GetClientesDTO.java
        PostClientesDTO.java
        PutClientesDTO.java
    src/main/resources
      static
      templates
      application.properties
      src/test/java
      JRE System Library [JavaSE-11]
      Maven Dependencies
      target
      HELP.md
      mvnw
      mvnw.cmd
      pom.xml
      api_produtos
      projetoAula01
      projetoAula02
      projetoAula03
      projetoAula04
      projetoAula05
      projetoAula06

Console X
api_clientes - ApiClientesApplication [Spring Boot App] C:\Program Files\Java\jdk-11.0.10\bin\javaw.exe (27 de fev de 2023 19:41:46)
19:41:48.685 [Thread-0] DEBUG org.springframework.boot.devtools.restart.classloader.RestartClassLoader - Cre
Spring Boot
=====
:: Spring Boot ::
(v2.7.9)

2023-02-27 19:41:49.267 INFO 19892 --- [ restartedMain] b.c.c.ApiClientesApplication : Starting
2023-02-27 19:41:49.270 INFO 19892 --- [ restartedMain] b.c.c.ApiClientesApplication : No acti
2023-02-27 19:41:49.384 INFO 19892 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtool
2023-02-27 19:41:49.385 INFO 19892 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For add
2023-02-27 19:41:50.981 INFO 19892 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat
2023-02-27 19:41:51.005 INFO 19892 --- [ restartedMain] o.apache.catalina.core.StandardService : Startin
2023-02-27 19:41:51.005 INFO 19892 --- [ restartedMain] org.apache.catalina.core.StandardEngine : Startin
2023-02-27 19:41:51.159 INFO 19892 --- [ restartedMain] o.a.c.c.C.[Tomcat].[/] : Initial
2023-02-27 19:41:51.159 INFO 19892 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root Wel
2023-02-27 19:41:51.642 INFO 19892 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveRel
2023-02-27 19:41:51.838 INFO 19892 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat
2023-02-27 19:41:52.043 INFO 19892 --- [ restartedMain] b.c.c.ApiClientesApplication : Started
  
```

Acessando

<http://localhost:8082/swagger-ui/index.html>



Swagger
Supported by SMARTBEAR

Select a definition default

API para controle de clientes Versão 1.0

[Base URL: localhost:8082/]
<http://localhost:8082/v2/api-docs>

Sistema Spring Boot API

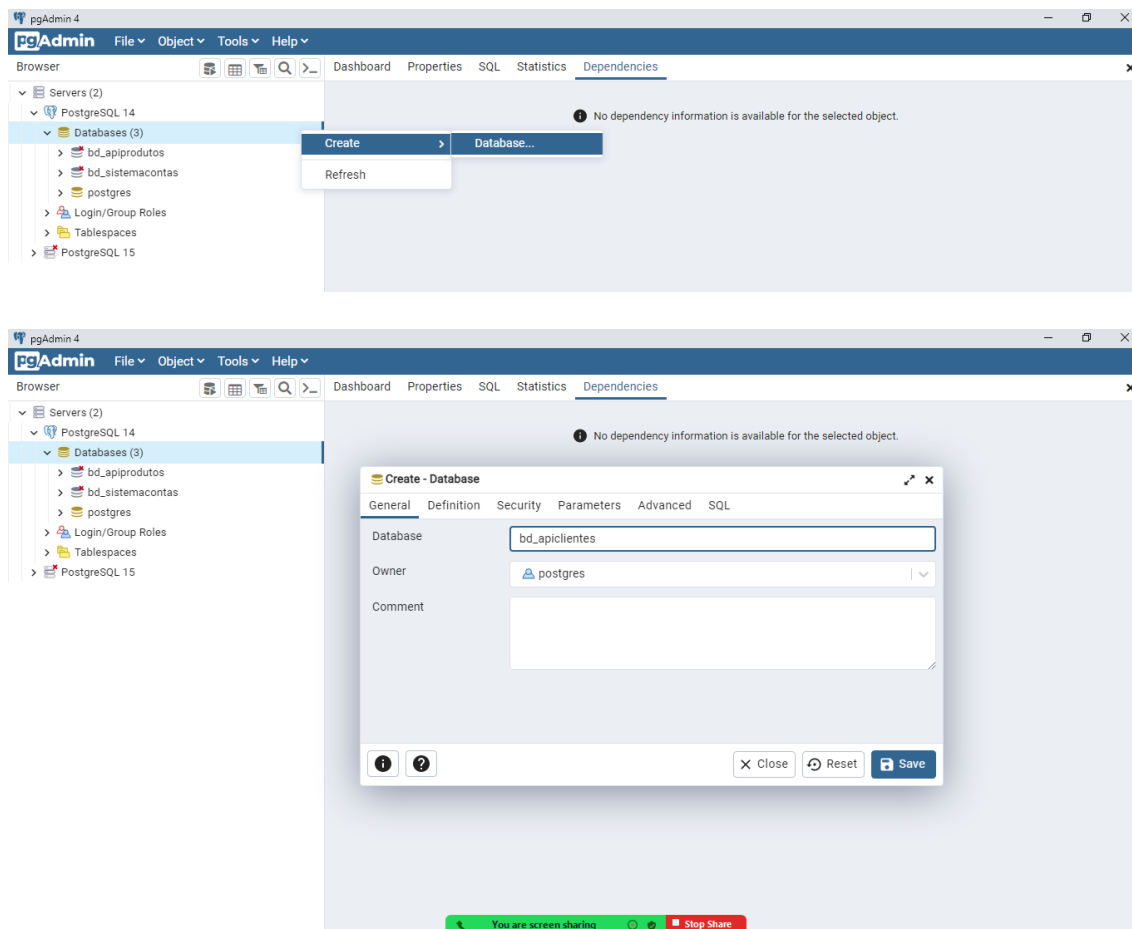
[Terms of service](#)
[COTI Informática - Website](#)
[Send email to COTI Informática](#)
[Licença da API](#)

Clientes Clientes Controller

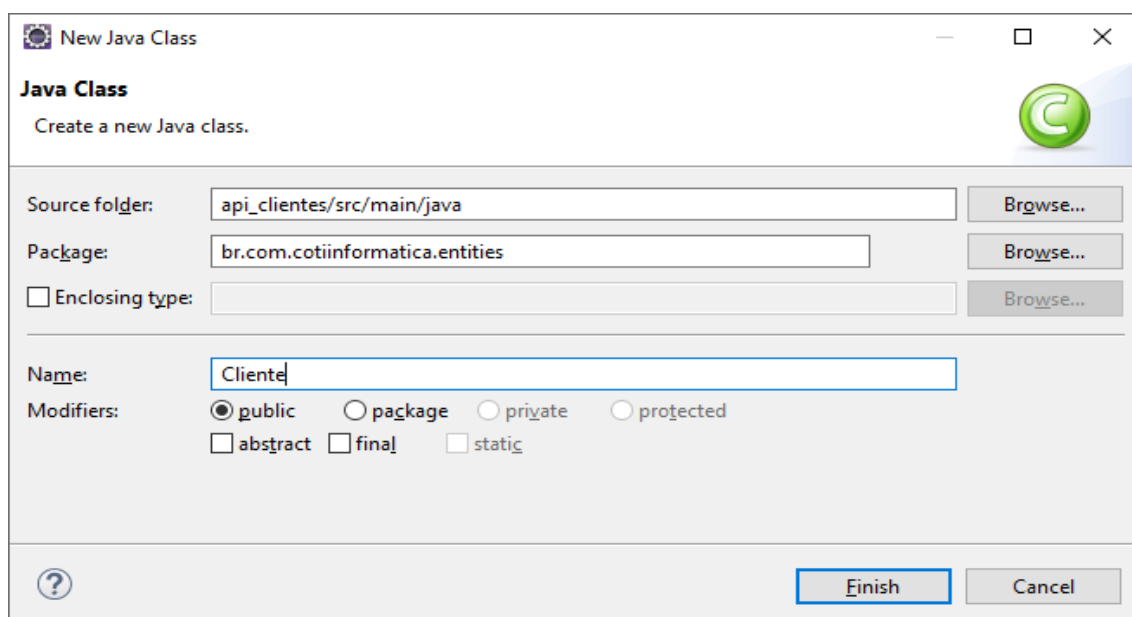
- GET** /api/clientes Serviço para consulta de clientes.
- POST** /api/clientes Serviço para cadastro de clientes.
- PUT** /api/clientes Serviço para edição de clientes.
- GET** /api/clientes/{id} Serviço para consulta de cliente por id.
- DELETE** /api/clientes/{id} Serviço para exclusão de clientes.

Models

Criando o banco de dados: PostgreSQL:



Criando a classe de entidade Modelo do banco de dados.



```
package br.com.cotiinformatica.entities;

import java.util.Date;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import lombok.ToString;

@Setter
@Getter
@NoArgsConstructor
@AllArgsConstructor
@ToString
public class Cliente {

    private Integer idCliente;
    private String nome;
    private String cpf;
    private String email;
    private String telefone;
    private Date dataCadastro;

}
```

Spring Data

Conjunto de tecnologias do framework Spring para implementarmos repositórios de acesso a banco de dados nos projetos.



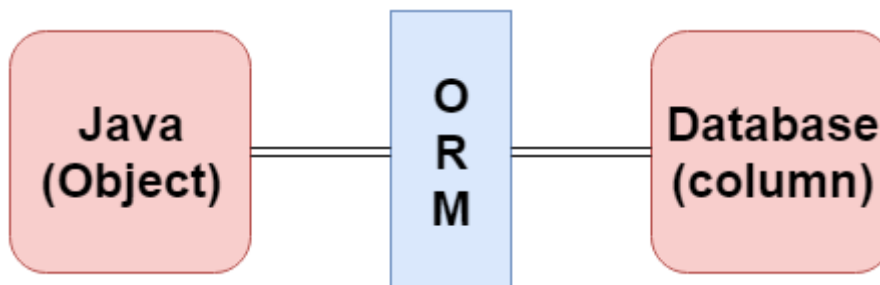
Spring Data

É composto de bibliotecas como:

- JDBC Template
- Hibernate e JPA (Java Persistence API)
- Drivers para bancos de dados não relacionais (NoSQL)

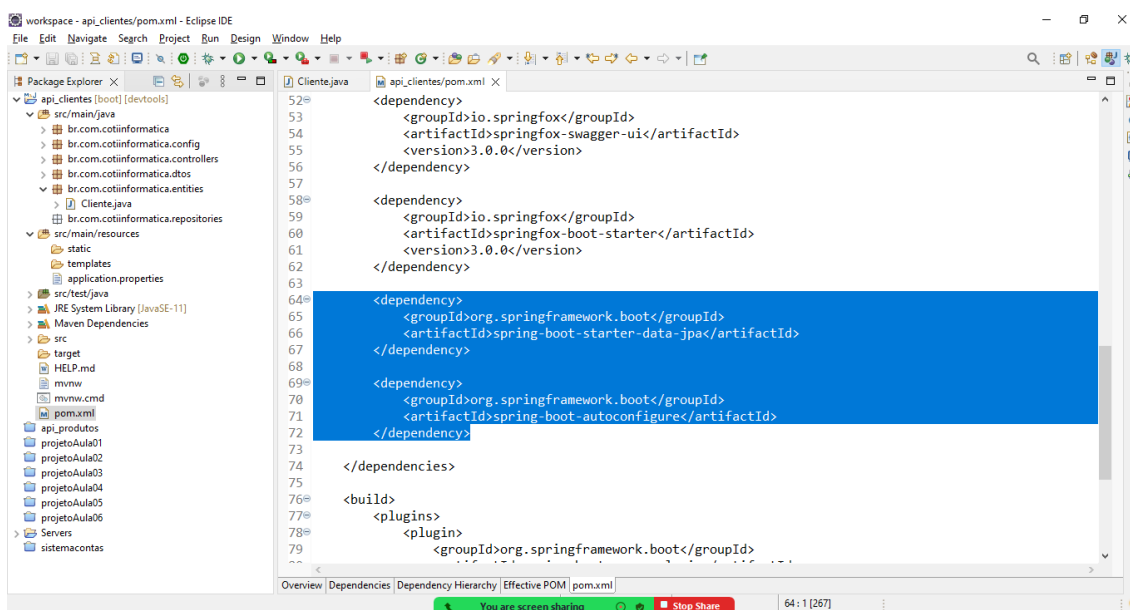
Hibernate & JPA (Java Persistence API)

São bibliotecas do Spring Data voltadas para conexão com bancos de dados utilizando um padrão chamado **ORM – Mapeamento Objeto Relacional**.



Primeiro, precisamos configurar as bibliotecas para o uso do Spring Data com Hibernate e JPA:

/pom.xml



```
<dependency>
```

```
  <groupId>org.springframework.boot</groupId>
```

```
  <artifactId>spring-boot-starter-data-jpa</artifactId>
```

```
</dependency>
```

```
<dependency>
```

```
  <groupId>org.springframework.boot</groupId>
```

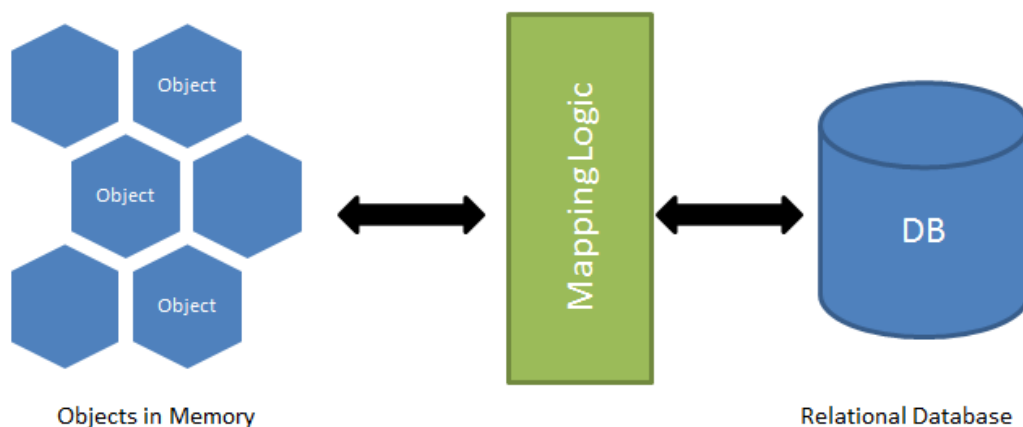
```
  <artifactId>spring-boot-autoconfigure</artifactId>
```

```
</dependency>
```


Primeiro passo:

Precisamos mapear as entidades do projeto, configurando quais tabelas do banco de dados serão associadas a cada classe de entidade do sistema.

O/R Mapping



Para fazermos o mapeamento, vamos utilizar a seguinte biblioteca:
`javax.persistence (JPA)`

@Entity

Define a classe como uma entidade do banco de dados. Toda classe mapeada como uma tabela do banco de dados deve obrigatoriamente ter essa anotação.

@Table

Define o nome da tabela do banco de dados

@Id

Define o campo chave primária da tabela, toda entidade mapeada deve obrigatoriamente ter uma chave primária.

@GeneratedValue

Define que um campo inteiro terá o seu conteúdo gerado automaticamente pelo banco de dados.

@Column

Mapeia as características de uma coluna da tabela.

@Temporal

Mapeia campos do tipo Data e Hora.

```
package br.com.cotiinformatica.entities;

import java.util.Date;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import lombok.ToString;

@Entity
@Table(name = "cliente")
@Setter
@Getter
@NoArgsConstructor
@AllArgsConstructor
@ToString
public class Cliente {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "idcliente")
    private Integer idCliente;

    @Column(name = "nome", length = 150, nullable = false)
    private String nome;

    @Column(name = "cpf", length = 11, nullable = false,
            unique = true)
    private String cpf;

    @Column(name = "email", length = 50, nullable = false,
            unique = true)
    private String email;

    @Column(name = "telefone", length = 14, nullable = false)
    private String telefone;

    @Temporal(TemporalType.DATE)
    @Column(name = "datacadastro", nullable = false)
    private Date dataCadastro;

}
```

/application.properties

Mapeamento da conexão com o banco de dados (**DATA SOURCE**):

server.port=8082

spring.datasource.url=jdbc:postgresql://localhost:5432/bd_apiclientes

spring.datasource.driver-class-name=org.postgresql.Driver

spring.datasource.username=postgres

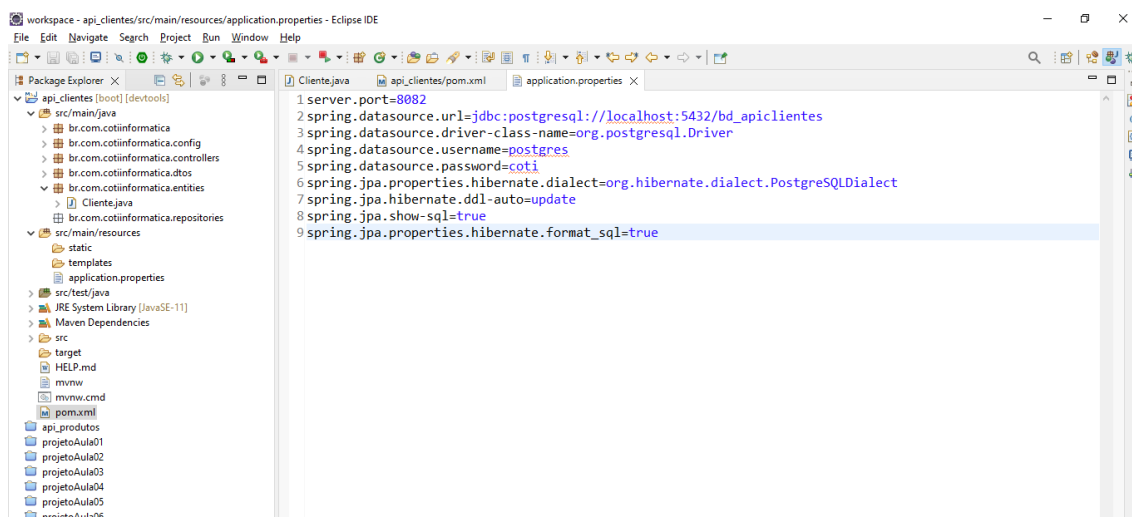
spring.datasource.password=coti

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect

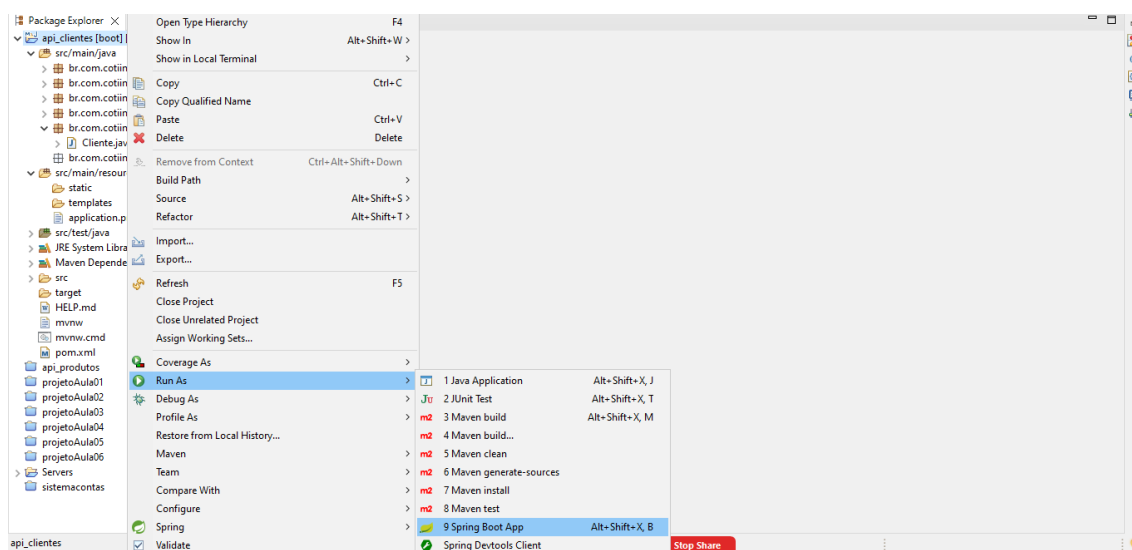
spring.jpa.hibernate.ddl-auto=update

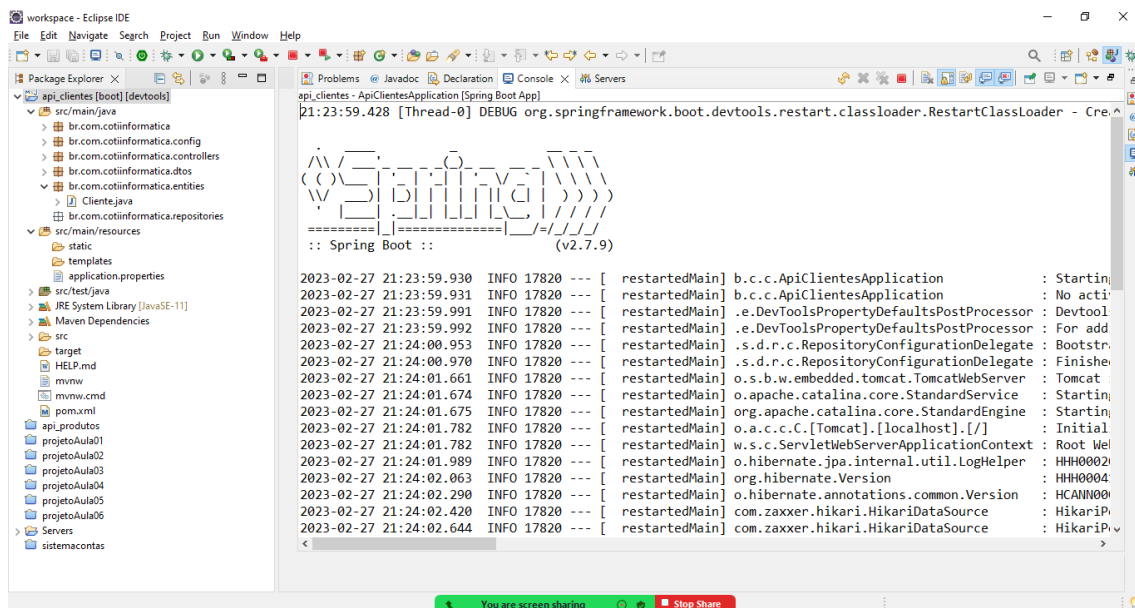
spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format_sql=true



Executando o projeto:





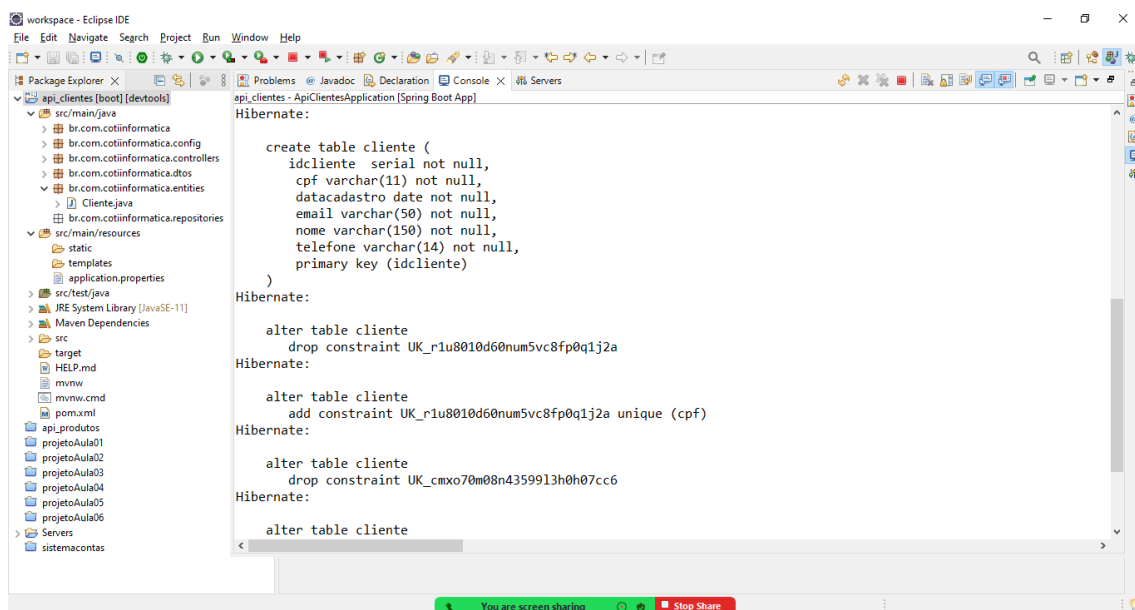
```

workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Package Explorer X
src/main/java
  br.com.cotinformatica
  br.com.cotinformatica.config
  br.com.cotinformatica.controllers
  br.com.cotinformatica.dtos
  br.com.cotinformatica.entities
  Cliente.java
  br.com.cotinformatica.repositories
src/main/resources
  static
  templates
  application.properties
src/test/java
  JRE System Library [JavaSE-11]
  Maven Dependencies
  src
  target
  HELP.md
  mvnw.cmd
  pom.xml
api_produtos
projetoAula01
projetoAula02
projetoAula03
projetoAula04
projetoAula05
projetoAula06
Servers
sistemacontas

api_clientes - ApiClientesApplication [Spring Boot App]
21:23:59.428 [Thread-0] DEBUG org.springframework.boot.devtools.restart.classloader.RestartClassLoader - Cre
=====
:: Spring Boot ::
(v2.7.9)

2023-02-27 21:23:59.930 INFO 17820 --- [ restartedMain] b.c.c.ApiClientesApplication : Starting
2023-02-27 21:23:59.931 INFO 17820 --- [ restartedMain] b.c.c.ApiClientesApplication : No acti
2023-02-27 21:23:59.991 INFO 17820 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtool
2023-02-27 21:23:59.992 INFO 17820 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For add
2023-02-27 21:24:00.953 INFO 17820 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Bootstr
2023-02-27 21:24:00.970 INFO 17820 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Finishe
2023-02-27 21:24:01.661 INFO 17820 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat
2023-02-27 21:24:01.674 INFO 17820 --- [ restartedMain] o.apache.catalina.core.StandardService : Startin
2023-02-27 21:24:01.675 INFO 17820 --- [ restartedMain] org.apache.catalina.core.StandardEngine : Startin
2023-02-27 21:24:01.782 INFO 17820 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initial
2023-02-27 21:24:01.782 INFO 17820 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root Wel
2023-02-27 21:24:02.063 INFO 17820 --- [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH00002
2023-02-27 21:24:02.290 INFO 17820 --- [ restartedMain] org.hibernate.Version : HHH000004
2023-02-27 21:24:02.420 INFO 17820 --- [ restartedMain] o.hibernate.annotations.common.Version : HCANN000
2023-02-27 21:24:02.420 INFO 17820 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariP
2023-02-27 21:24:02.644 INFO 17820 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariP

```



```

workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Package Explorer X
src/main/java
  br.com.cotinformatica
  br.com.cotinformatica.config
  br.com.cotinformatica.controllers
  br.com.cotinformatica.dtos
  br.com.cotinformatica.entities
  Cliente.java
  br.com.cotinformatica.repositories
src/main/resources
  static
  templates
  application.properties
src/test/java
  JRE System Library [JavaSE-11]
  Maven Dependencies
  src
  target
  HELP.md
  mvnw.cmd
  pom.xml
api_produtos
projetoAula01
projetoAula02
projetoAula03
projetoAula04
projetoAula05
projetoAula06
Servers
sistemacontas

api_clientes - ApiClientesApplication [Spring Boot App]
Hibernate:
create table cliente (
  idcliente serial not null,
  cpf varchar(11) not null,
  datacadastro date not null,
  email varchar(50) not null,
  nome varchar(150) not null,
  telefone varchar(14) not null,
  primary key (idcliente)
)
Hibernate:
alter table cliente
drop constraint UK_r1u8010d60num5vc8fp0q1j2a
Hibernate:
alter table cliente
add constraint UK_r1u8010d60num5vc8fp0q1j2a unique (cpf)
Hibernate:
alter table cliente
drop constraint UK_cmoxo70m08n4359913h0h07cc6
Hibernate:
alter table cliente

```

Hibernate:

```

create table cliente (
  idcliente serial not null,
  cpf varchar(11) not null,
  datacadastro date not null,
  email varchar(50) not null,
  nome varchar(150) not null,
  telefone varchar(14) not null,
  primary key (idcliente)
)

```

Hibernate:

```

alter table cliente
drop constraint UK_r1u8010d60num5vc8fp0q1j2a

```

Hibernate:

```
alter table cliente
add constraint UK_r1u8010d60num5vc8fp0q1j2a unique (cpf)
```

Hibernate:

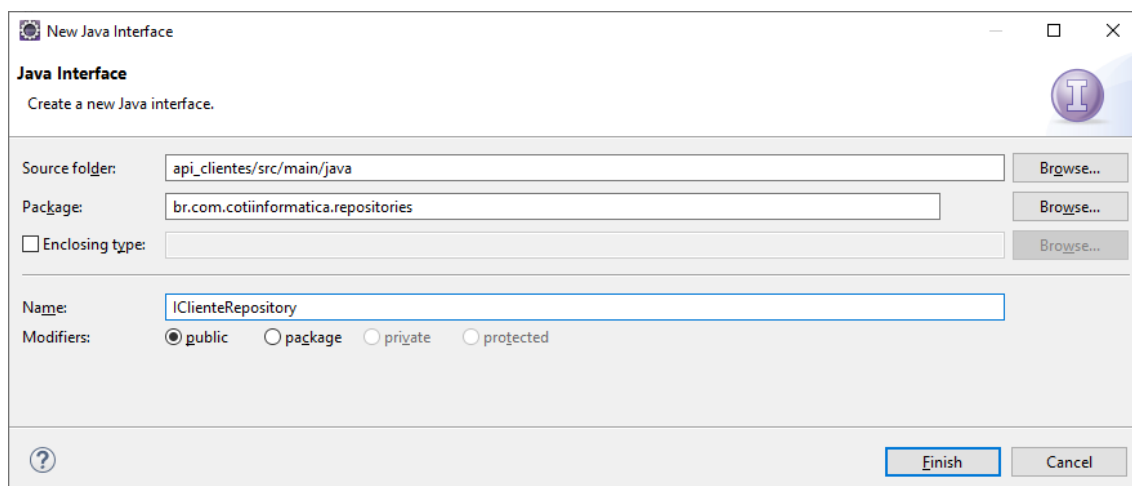
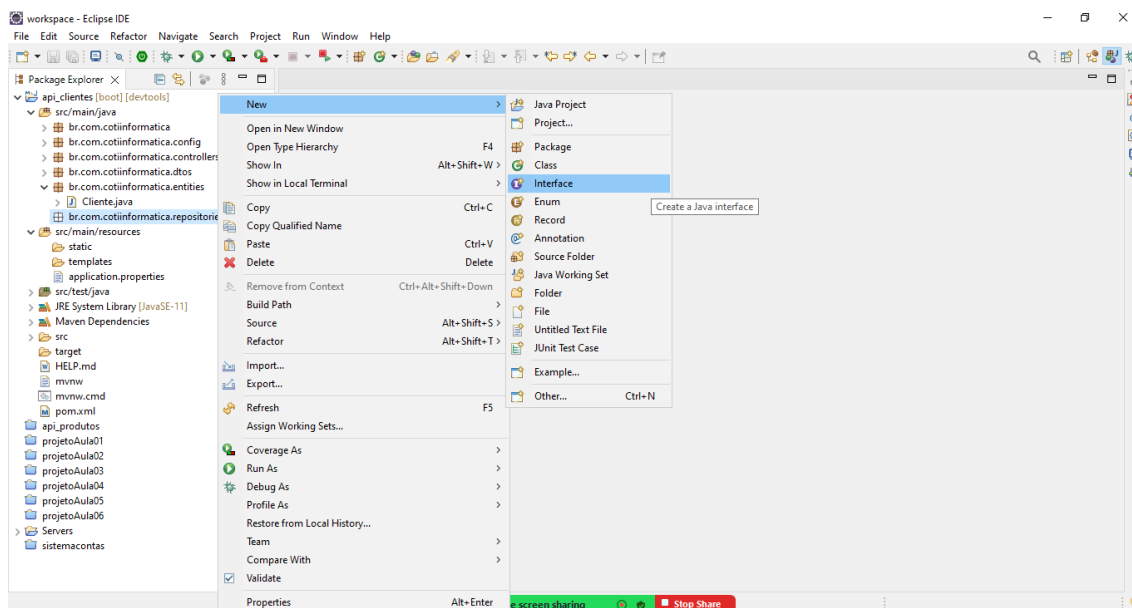
```
alter table cliente
drop constraint UK_cmoxo70m08n4359913h0h07cc6
```

Hibernate:

```
alter table cliente
add constraint UK_cmoxo70m08n4359913h0h07cc6 unique (email)
```

Criando os repositórios do projeto:

Desenvolvendo os repositórios através de interfaces.



@Repository

Mapeando a interface como um repositório do Spring Data, que permitirá o uso dos métodos do Hibernate e JPA.

```
package br.com.cotiinformatica.repositories;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import br.com.cotiinformatica.entities.Cliente;

@Repository
public interface IClienteRepository
    extends JpaRepository<Cliente, Integer> {

}
```

Publicando no GITHUB:

```
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 -
Java WebDeveloper SQS 18h as 22h (Início em
09.01)/workspace/api_clientes
$ git init
```

```
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 -
Java WebDeveloper SQS 18h as 22h (Início em
09.01)/workspace/api_clientes (master)
$ git branch -m main
```

```
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 -
Java WebDeveloper SQS 18h as 22h (Início em
09.01)/workspace/api_clientes (main)
$ git add .
```

```
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 -
Java WebDeveloper SQS 18h as 22h (Início em
09.01)/workspace/api_clientes (main)
$ git commit -m 'first commit'
```

```
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 -
Java WebDeveloper SQS 18h as 22h (Início em
09.01)/workspace/api_clientes (main)
$ git remote add origin
https://github.com/smendescoti/api_clientes.git
```

```
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 -
Java WebDeveloper SQS 18h as 22h (Início em
09.01)/workspace/api_clientes (main)
$ git push -u origin main
```

Continua...