PROJETO - DESAFIO KYROS

****

CRIAR UMA APP MOBILE PARA CADASTRO DE CLIENTES VIA WS

****

FERRAMENTAS UTILIZADAS:

FERRAMENTA	VERSÃO	UTILIDADE
Apache TomCat	apache-tomcat-8.5.38-windows-x64	Hospedagem do WebService
Axis2	axis2-1.7.9-bin	Biblioteca para WS
Eclipse	Oxygen.2 Release (4.7.2)	IDE Java EE
KSoap	ksoap2-android-assembly-3.3.0-jar-with-dependencies	Biblioteca para Consumir WS
MySql	mysql-installer-community-8.0.15.0	Servidor de Banco de Dados
Conector	mssql-connector-java-8.0.14	Conector Mysql x Java
SoapUI	SoapUI-5.2.1-win32-standalone-bin	Teste de WebService

➔ **Instalar o Banco de Dados Mysql (mysql-installer-community-8.0.15.0)**

- Usuário: root
- Senha: root

➔ **Criar Banco de Dados e tabela:**

- Criar Schema / Banco kyros:

```
CREATE SCHEMA `kyros`;
```

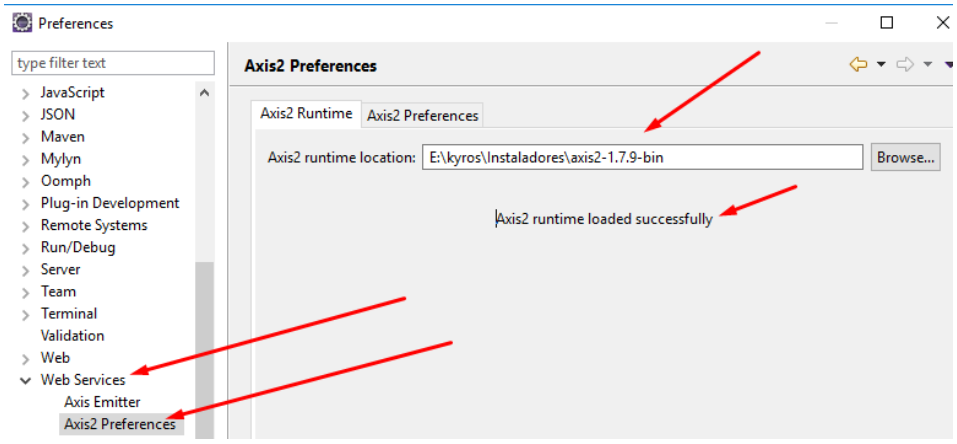
- Criar Tabela Clientes:

```
CREATE TABLE `kyros`.`cliente` (
`id` INT NOT NULL AUTO_INCREMENT,
`nome` VARCHAR(60) NULL,
`cpf` VARCHAR(11) NULL,
`datanasc` DATE NULL,
`telefone` VARCHAR(14) NULL,
PRIMARY KEY (`id`));
```

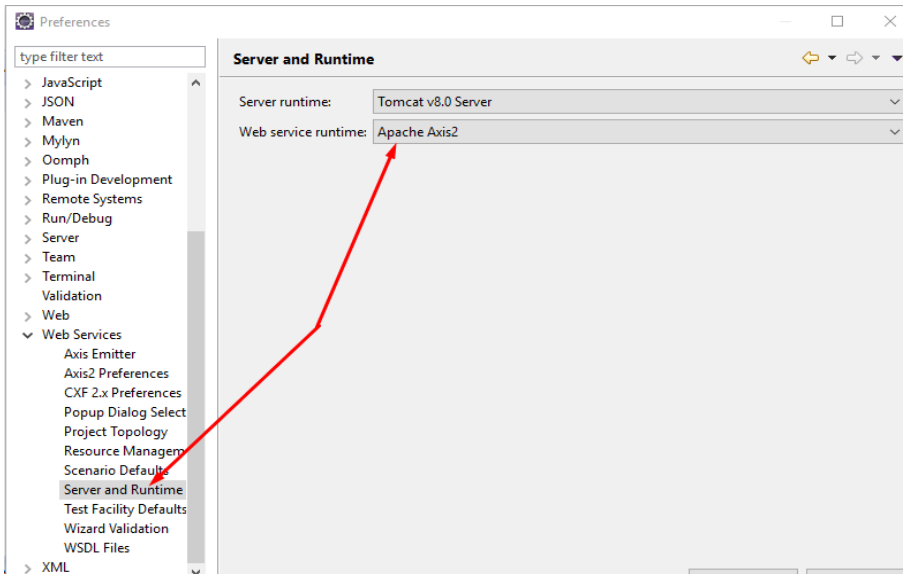
➔ **Extrair e executar o Eclipse (eclipse-jee-oxygen-2-win32-x86_64.zip)**

➔ **Configuar Axis2 no Eclipse**

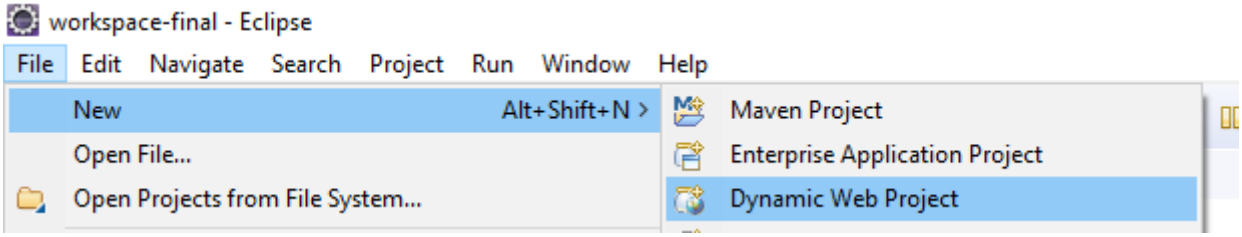
- Vá em Window / Preferences / Web Services / Axis2 Prefences.
- Em “Axis2 runtime location:” e informe o local onde está o Axis2



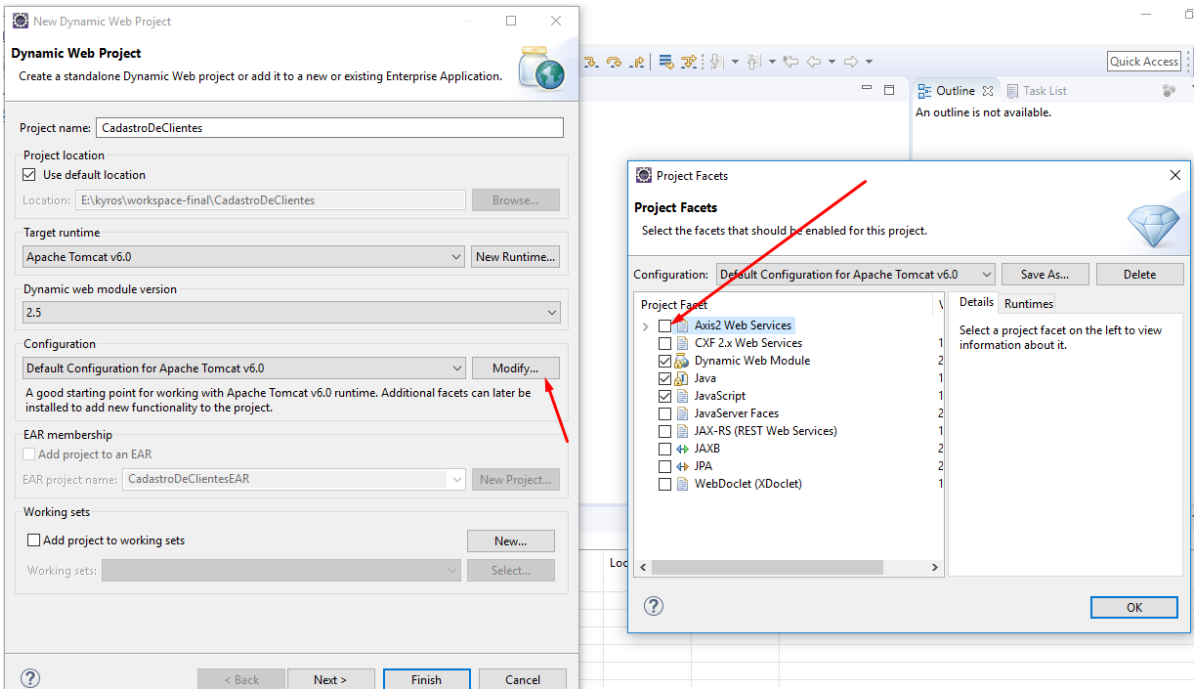
- Ir em “Server and runtime” e em “Web servisse runtime” e selecionar “Apache Axis2”



➔ Criar um Projeto Web
  - Em New / Dynamic Web Project



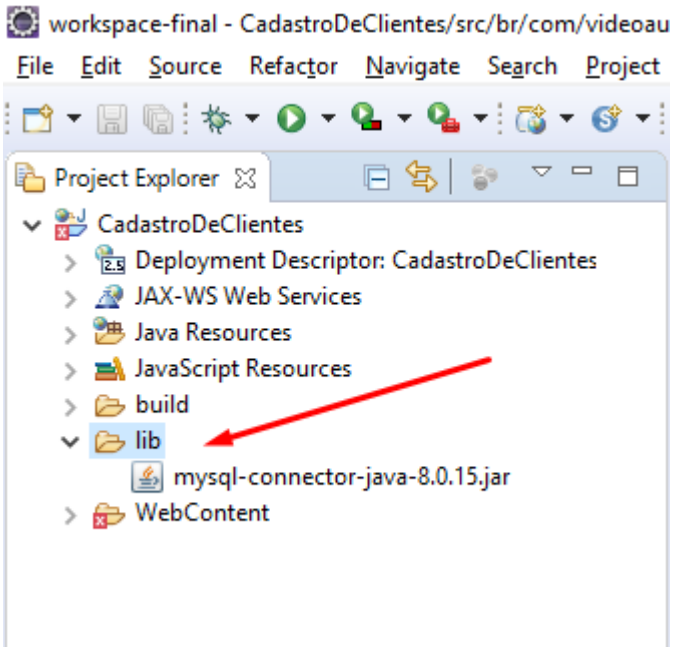
➔ Informar o Nome do Projeto;
➔ Se necessário, configurar uma "Target runtime";
➔ Vamos usar o Apache Tomcat v6.0;
➔ Em "Configuration", clicar no botão "Modify" e selecionar Axis2 Web Services;
➔ Ok e Finish;



➔ No Projeto, em Java Resources / scr, criar a classe Conecta.java no pacote br.com.kyros.cadastro:

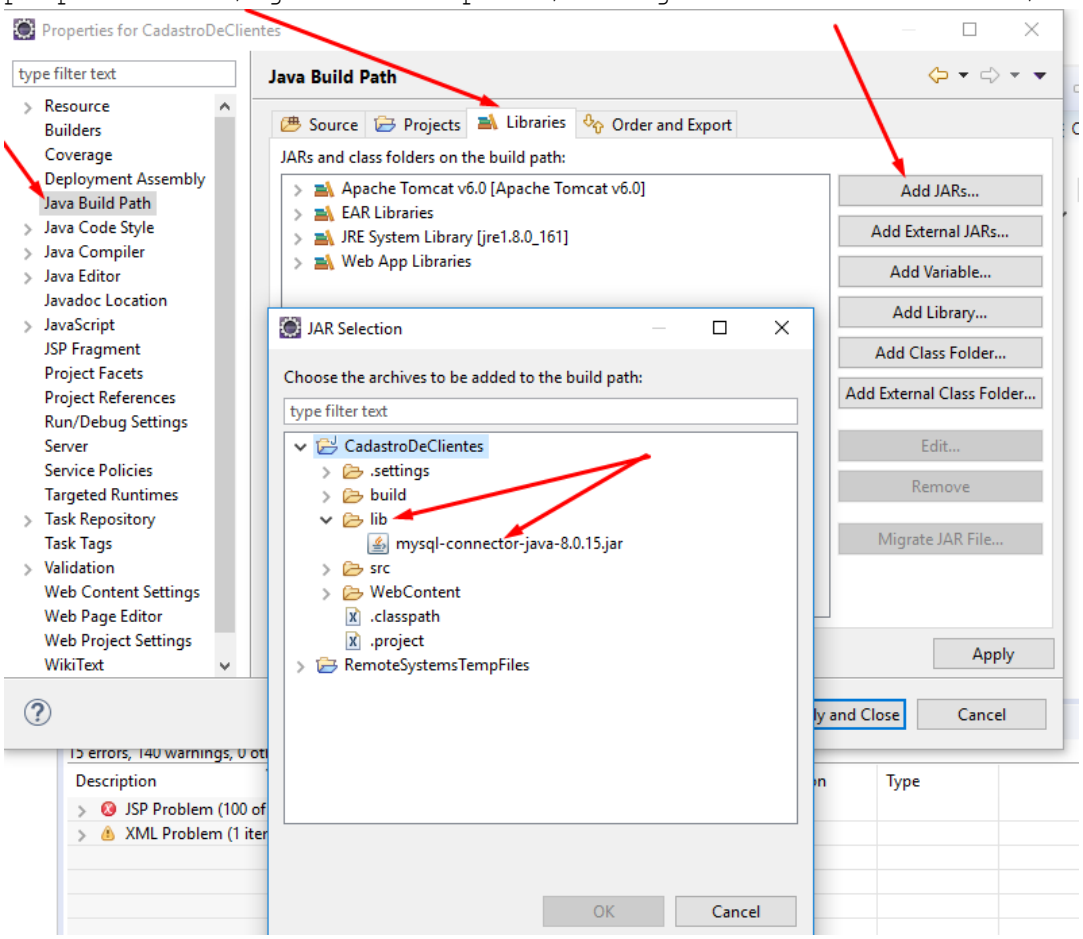
```
package br.com.kyros.cadastro;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

class ConectaMySql {

   private static final String URL =
"jdbc:mysql://localhost/kyros?useTimezone=true&serverTimezone=UTC";
   private static final String USER = "root";
   private static final String SENHA = "root";

   public static Connection obtemConexao() throws SQLException
   {
         try {
               Class.forName("com.mysql.cj.jdbc.Driver");

         } catch (ClassNotFoundException e) {
               e.printStackTrace();
         }

         return DriverManager.getConnection(URL, USER, SENHA);
   }
}
```

➔ Criar o um Folter "lib" no projeto e arrastar a biblioteca de conexão do mysql (mysql-connector-java-8.0.15) para ela:



➔ Depois, copiar a biblioteca para a pasta lib do Apache Tomcat (E:\kyros\Instaladores\apache-tomcat-6.0.41\lib)

➔ Em seguida, clicar com o botão direito do mouse sobre o projeto e ir em propriedades / java build path / add jars e selecionar em /lib o conector.



➔ Criar uma classe Clientes.java

```
package br.com.kyros.cadastro;

public class Cliente {

	private int id;
	private String nome;
	private String cpf;
	private String dataNasc;
	private String email;
	private String telefone;

	public Clientes() {

	}

	public Cliente(int id, String nome, String cpf, String dataNasc, String telefone) {

		this.id = id;
		this.nome = nome;
		this.cpf = cpf;
		this.dataNasc = dataNasc;
		this.email = email;
		this.telefone = telefone;
	}

	public int getId() {
		return id;
	}

	public void setId(int id) {
		this.id = id;
```

```
	}

	public String getNome() {
		return nome;
	}

	public void setNome(String nome) {
		this.nome = nome;
	}

	public String getCpf() {
		return cpf;
	}

	public void setCpf(String cpf) {
		this.cpf = cpf;
	}

	public String getDataNasc() {
		return dataNasc;
	}

	public void setDataNasc(String dataNasc) {
		this.dataNasc = dataNasc;
	}

	public String getEmail() {
		return email;
	}

	public void setEmail(String email) {
		this.email = email;
	}

	public String getTelefone() {
		return telefone;
	}

	public void setTelefone(String telefone) {
		this.telefone = telefone;
	}

}
```

➔ Criar a Classe ClientesDAO.java

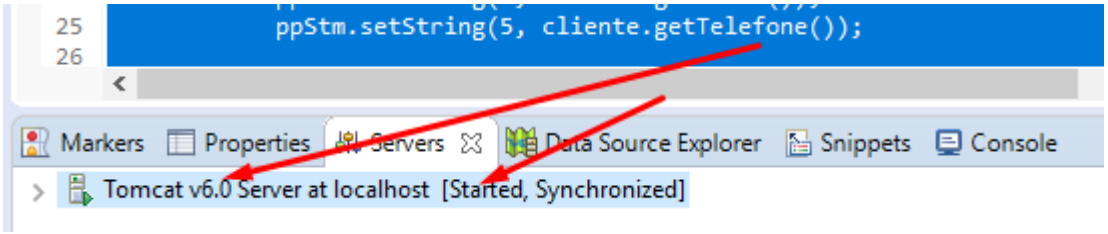
```
package br.com.kyros.cadastro;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;

public class ClientesDAO {

	public boolean inserir(Clientes cliente) {

		try {

			new ConectaMySql();
			Connection conn = ConectaMySql.conecta();

			String sql = "insert into clientes values (null, ?, ?, ?, ?, ?)";

			PreparedStatement ppStm = conn.prepareStatement(sql);

			ppStm.setString(1, cliente.getNome());
			ppStm.setString(2, cliente.getCpf());
			ppStm.setString(3, cliente.getDataNasc());
			ppStm.setString(4, cliente.getEmail());
			ppStm.setString(5, cliente.getTelefone());

			ppStm.executeUpdate();
```

```
                ppStm.close();

            } catch (Exception e) {
                e.printStackTrace();
                return false;
            }

            return true;
        }


        public boolean excluir(Clientes cliente) {

            try {

                new ConectaMySql();
                Connection conn = ConectaMySql.conecta();

                String sql = "delete from clientes where id = ?";

                PreparedStatement ppStm = conn.prepareStatement(sql);

                ppStm.setInt(1, cliente.getId());

                ppStm.executeUpdate();

                ppStm.close();

            } catch (Exception e) {
                e.printStackTrace();
                return false;
            }

            return true;
        }

        public boolean atualizar(Clientes cliente) {

try {

                Connection conn = new ConectaMySql().conecta();

                String sql = "update clientes set nome = ?, cpf = ?, datanasc = ?, email =
?, telefone = ? where id = ?";

                PreparedStatement ppStm = conn.prepareStatement(sql);

                ppStm.setString(1, cliente.getNome());
                ppStm.setString(2, cliente.getCpf());
                ppStm.setString(3, cliente.getDataNasc());
                ppStm.setString(4, cliente.getEmail());
                ppStm.setString(5, cliente.getTelefone());

                ppStm.executeUpdate();
                ppStm.close();

            } catch (Exception e) {
                e.printStackTrace();
                return false;
            }

            return true;
        }

        public ArrayList<Clientes> buscarTodosClientes(){

            Clientes cli = null;

            ArrayList<Clientes> lista = new ArrayList<Clientes>();

            try {
```

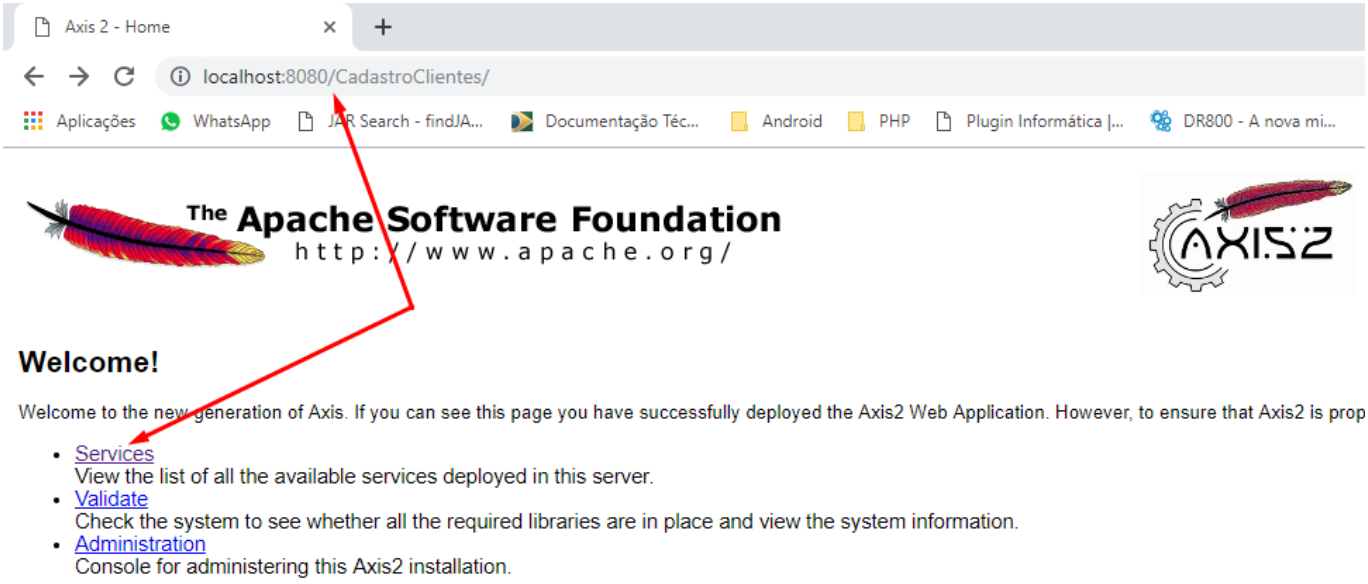
```
			Connection conn = new ConectaMySql().conecta();

			String sql = "select * from clientes order by nome";

			PreparedStatement ppStm = conn.prepareStatement(sql);

			ppStm.executeQuery(sql);

			ResultSet rs = ppStm.executeQuery();

			while(rs.next())
			{

				cli = new Clientes();
				cli.setId(rs.getInt(1));
				cli.setNome(rs.getString(2));
				cli.setCpf(rs.getString(3));
				cli.setDataNasc(rs.getString(4));
				cli.setEmail(rs.getString(5));
				cli.setTelefone(rs.getString(6));

				lista.add(cli);
			}

			ppStm.close();

		} catch (Exception e) {
			e.printStackTrace();
		}

		return lista;
	}

	public Clientes buscarClientePorID(int id) {

		Clientes cli = null;

		try {

			Connection conn = new ConectaMySql().conecta();

			String sql = "select * from usuario where id = ?";

			PreparedStatement ppStm = conn.prepareStatement(sql);
			ppStm.setInt(1, id);

			ResultSet rs = ppStm.executeQuery();

			if(rs.next())
			{
				cli = new Clientes();

				cli.setId(rs.getInt(1));
				cli.setNome(rs.getString(2));
				cli.setCpf(rs.getString(3));
				cli.setDataNasc(rs.getString(4));
				cli.setEmail(rs.getString(5));
				cli.setTelefone(rs.getString(6));

			}else
			{
				return cli;
			}

			ppStm.close();

		} catch (Exception e) {
			e.printStackTrace();
		}

		return cli;
	}
```

```
    public boolean excluir(int id) {

        return excluir(new Clientes(id, "", "", "", ""));
    }

}
```

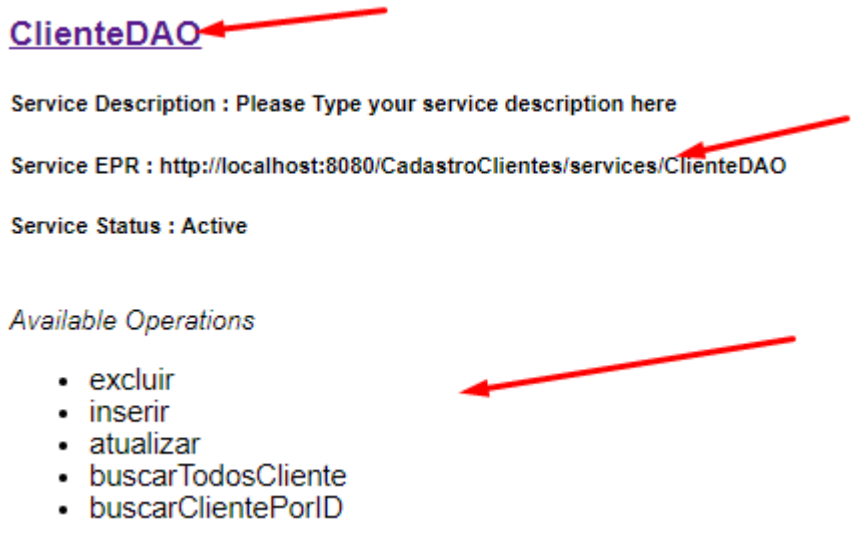
➔ Verificar se o TomCat esta iniciado:



➔ Testar o TomCat no navegador: localhost:8080/CadastroClientes:



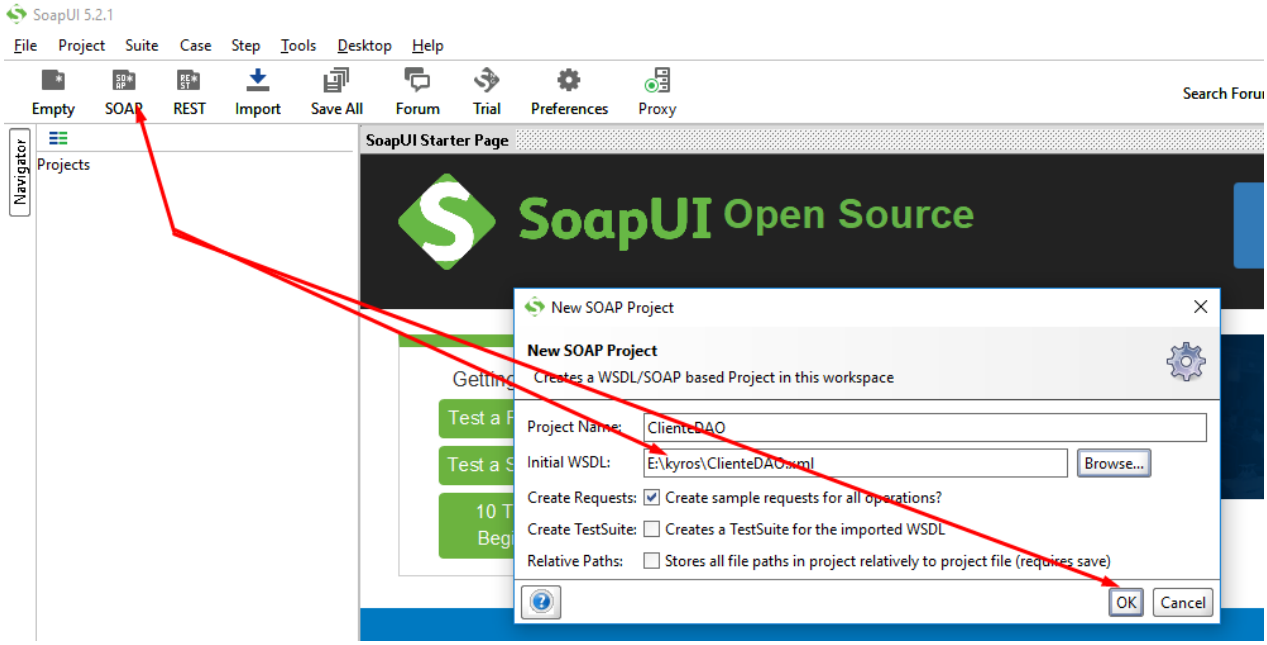
➔ Em Services, contém a lista dos serviços oferecidos:



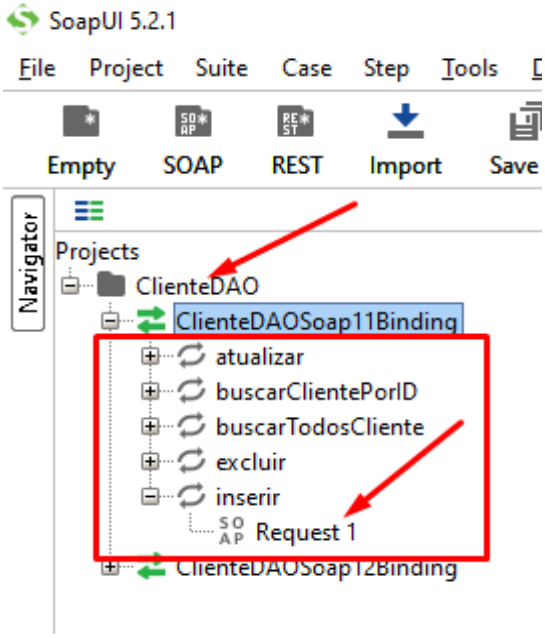
➔ Clicando em ClienteDAO, teremos acesso ao arquivo http://localhost:8080/CadastroClientes/services/ClienteDAO?wsdl no formato xml, onde contém todos os serviços. Salvar em uma pasta...

➔ Para testar cada serviço, utilizei o SoapUI (E:\kyros\Instaladores\ SoapUI-5.2.1\bin\soapui.bat)

- ⇨ Criar um novo Projeto SOAP;
- ⇨ Carregar o arquivo WSDL salvo anteriormente;

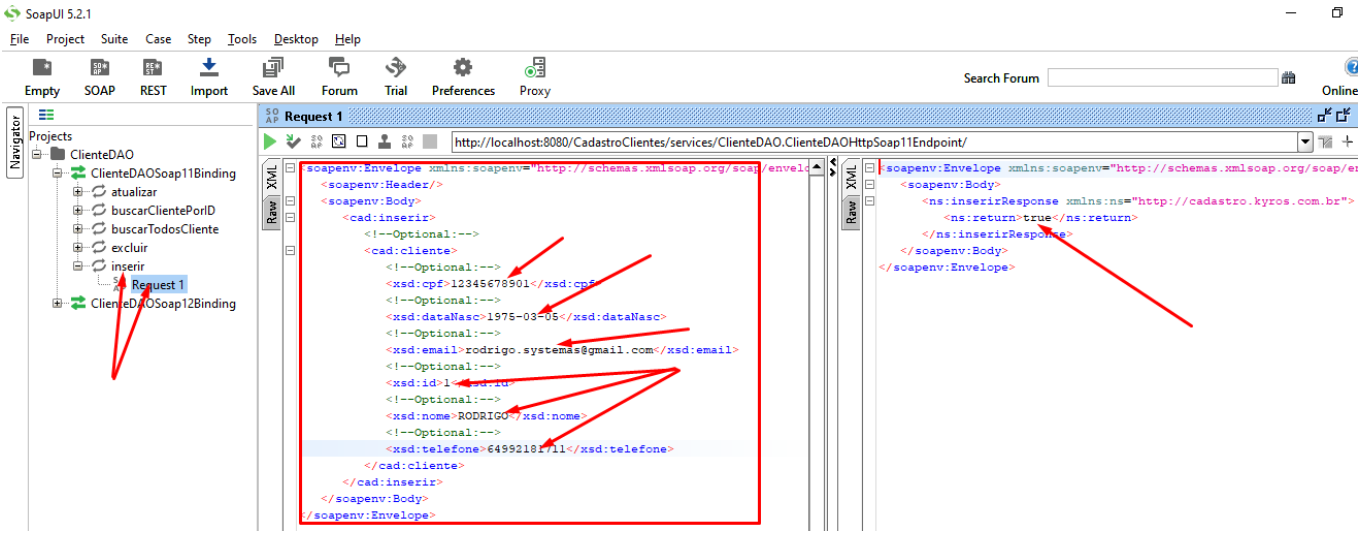


➔ Será criado o projeto com o nome do ArquivoDAO, seus serviços e respectivos requests:

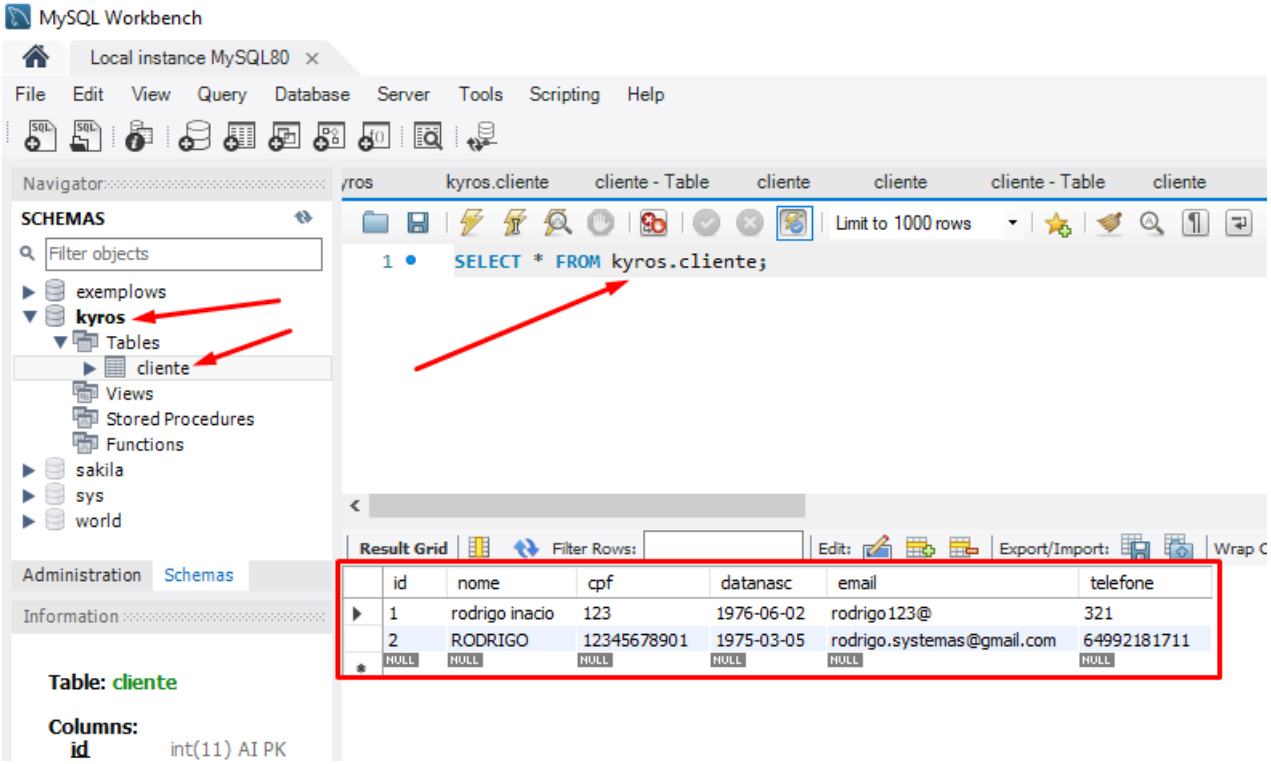


➔ Testado os serviços:

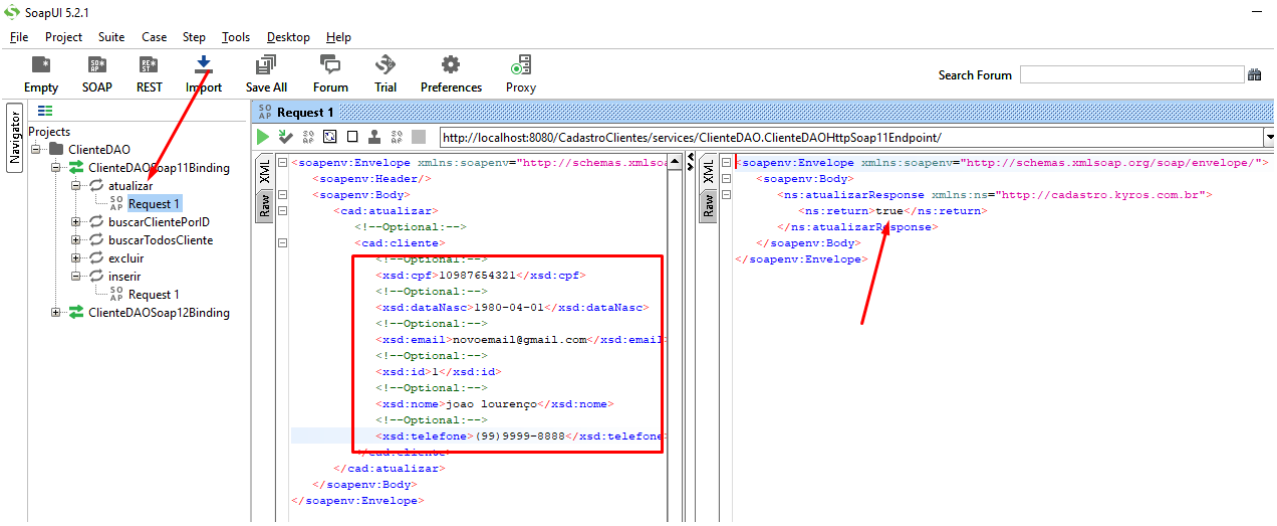
- o Inserir:



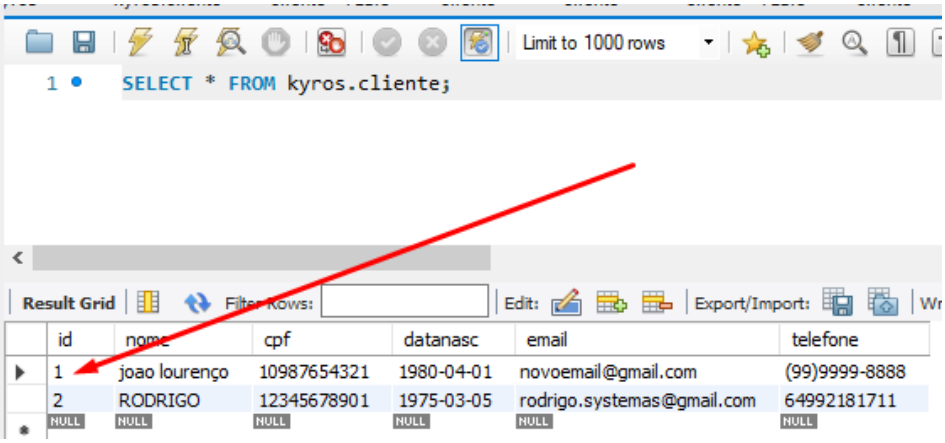
➔ Os retornos do serviço solicitado são mostrados do lado direito, no caso do insert, retornou true e podemos ver os dados no banco de dados:



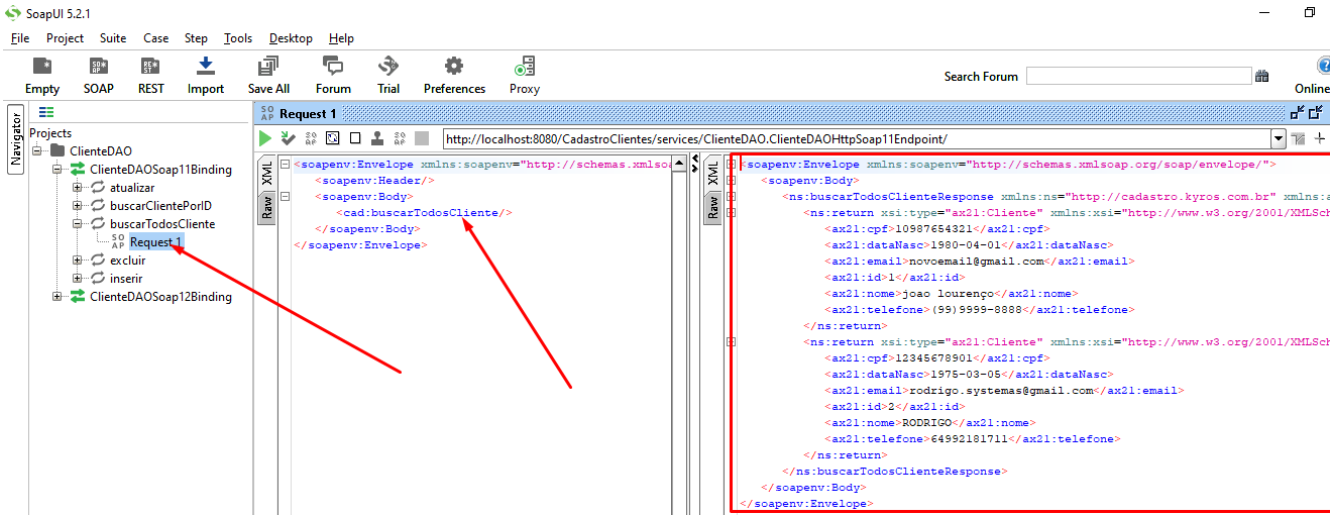
➔ Alterar:



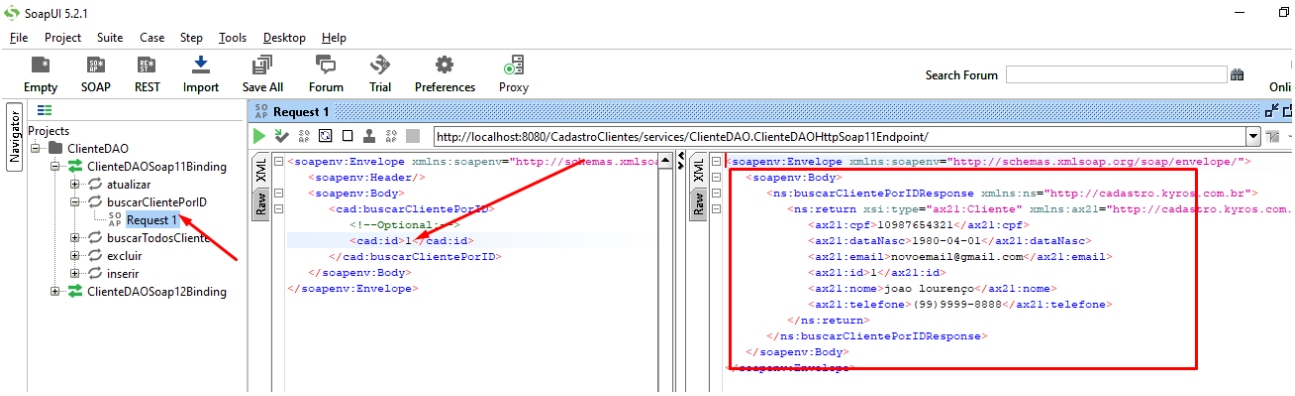
➪ Alteração realizada no banco



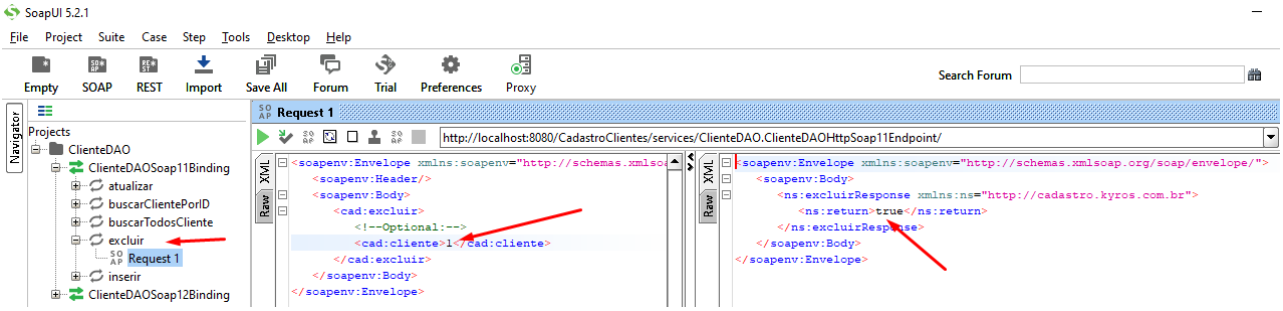
➔ Busca todos os Clientes:



➔ Buscar por Id:



➔ Exclusão:



⇨ Exclusão no Banco:

```
1 •  SELECT * FROM kyros.cliente;
```