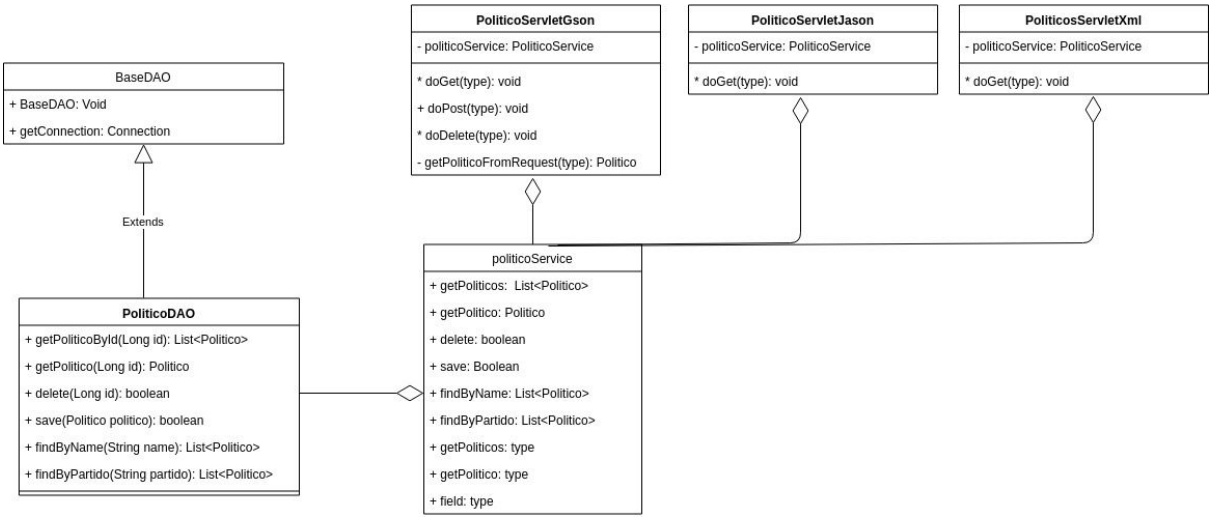
Nome: Gustavo Santos Leandro **Prontuario:** GU166254-6

Avaliação individual

1. Crie um banco de dados chamado políticos com a tabela com os atributos da figura abaixo no MySQL. Faça os inserts dos políticos (TODOS os vereadores) de Guarulhos do site da Câmara:

http://www.camaraguarulhos.sp.gov.br/ords/cmguaru/f?p=309:17:1616429443 1077:: ::P17_ID_CADERNO:906) **(Vale 1 ponto)**

2. Crie o diagrama de classe do WebService. Utilize o draw.io **(1 ponto)**



3. Mostrar a Servlet com retorno em XML e em JSON: **(0,75 pontos)**

4. URL no formato REST com /pagina/id . Por exemplo, http://localhost:8080/Politicos/politico/1. Esse padrão do REST indica que desejamos consultar o político com id = 1. **(0,75 pontos)**

5. Explique com suas palavras o que as classes RegexUtil.java, JAXBUtil.java e ServletUtil.java fazem. **(0,75 pontos)**

   O arquivo RegexUtil.java intepreta a URL e verifica qual tipo de requisição ele tem que realizar, podendo se usar expressão regular para realizar esse tipo de identificação. Ele pode interpretar se se a URL ele é padrão para realizar a identificação. Como exemplo se na URL tiver ID ele retorna somente aquele objeto caso contrário a lista com todos os objetos.

   Já a classe JAXBUtil.java serve para gerar XML sendo que este XML e converte para objeto ou vice e versa, necessita ser enviado como resposta HTTP. Sendo que é necessário passar o tipo de arquivo que está sendo passado, ele serve para trabalhar com o XML .Ele trabalha com métodos que trabalha com XML.

   O ServletUtil.java ele faz o trabalho de escrever o XML ou JSON, como resposta de uma requisição HTTP. Ele tem uma metodos para a escrita de JSON e XML e isso é configurado. Isso mostra as endpoints da aplicação e tem como função gerenciar as rotas com o serviço.

6. Salvar, atualizar e deletar um político com os métodos do HTTP: POST e DELETE **(0,75 pontos)**

7. Utilize o Jersey para criar um webService RESTful, que irá consultar, deletar e atualizar um político: **(3 pontos)**

8. Realize testes unitários com o JUnit no seu webService em xml e em json **(1 ponto)**.

9. Explique o que é REST e o termo RESTful **(0,5 pontos).**
   Resposta: Rest (**Transferência de Estado Representacional**) e um modelo arquitetural de software distribuído, que está sendo bem utilizado no mundo para criar serviços na web e para realizar também integrações de sistema.

   O Rest trabalha com o protocolo HTTP para que criar seus serviços que retornam dados em alguns formatos tais como XML ou JSON, não possui estado entre as comunicações, sendo assim cada comunicação é independente e uniforme, ou seja, trabalha de forma padronizada, porém cada URI tem uma para seja possível consultar informações sobre um determinado dado porém mantendo um padrão para esses tipo de consulta

   Ele pode seguir um padrão para realizar um conjunto de operação tais como GET, POST, PUT e DELETE como padrão para realização tarefas.

   É uma alternativa como para o SOAP e o WSDL.

   O REST tem alguns bons hábitos e padrões que podem ser seguidos para como boa prática para desenvolvimento, tais esses hábitos faz com que sua aplicação distribuída fique com um padrão de mercado, tais como utilizar o PUT para atualizar um dados, POST para armazenar um dados.

   O Termo RESTFul e para indicar em um determinado sistema ou serviço que segue os princípios do REST como exemplo GET, POST, PUT e DELETE , sendo que cada um tem seu respectivo papel para a aplicação. Sendo assim se o WEB SERVICE conter os princípios básicos completo pode se dizer que ele é um WEB SERVICE RESTFul.

10. Explique por que utilizamos o Jersey. De exemplos do código. **(0,5 pontos)**.

    Resposta: O Jersey é uma um framework java para implementar características de arquitetura de REST na aplicação distribuída assim facilitando o desenvolvimento da aplicação, ele gera algumas facilidades tais como o padrão do REST tais como GET, POST, PUT e DELETE.
    Um exemplo que comprova o que foi mostrado acima é:

```
package br.com.livro.rest;
import javax.ws.rs.*;
```

```
import javax.ws.rs.Path;
import javax.ws.rs.core.MediaType;
@Path("/hello")
public class HelloResource {
	@GET
	public String get() {
		return "HTTP GET";
	}
	@POST
	public String post(){
		return "HTTP POST";
	}
		@PUT
	public String put() {
		return "HTTP PUT";
	}
		@DELETE
	public String delete() {
		return "HTTP DELETE";
	}
}
```

Ele deixa como esses tipos de rotas prontas para possamos utilizar de forma mais simples.

Exemplo mostra o path que seria o caminho na qual se pode realizar a requisição e o @get, @post mostra mágicos e indicam o métodos que são chamados em uma requisição http.