
	<p>Ministério da Educação Universidade Tecnológica Federal do Paraná Campus Toledo Engenharia de Computação</p>	
---	---	---

## Sistemas Distribuídos

### Webservice REST para cadastro de vacinação contra COVID-19

O objetivo do trabalho é implementar dois recursos para gerenciar um sistema simplificado de vacinação: Pacientes e Imunizações. Não é necessário que seja utilizado um sistema de banco de dados (caso queiram, não há problema). A linguagem de implementação também é de livre escolha, ficando como sugestão o uso do CherryPy, visto em aula. Não é necessário implementar o frontend (lado cliente), mas se quiserem fazer, para os testes, também é livre.

Cada recurso do tipo Paciente contém, ao menos, os campos de CPF, nome, idade e telefone de contato. Considere, nesse ponto, que o CPF chega sem pontuação e já validado. Verifique apenas se a requisição de criação do recurso vem com todos os campos pedidos. A atualização do recurso Paciente pode alterar seus dados cadastrais, mas não informações relativas às doses aplicadas, que deverão ser feitas no recurso Imunização. Ao se pedir os dados de um Paciente, os dados de imunização também são retornados. O paciente poderá ter uma ou duas doses de imunização, a depender do fabricante.

Já o recurso do tipo Imunização terá lote, data de aplicação, fabricante e dose aplicada (primeira ou segunda, pode ser armazenada como número), além do status (aplicada ou agendada). O uso de GET retorna os dados de uma dada Imunização (vindo na URI `/pacientes/id_paciente/imunizacoes` ou `/pacientes/id_paciente/imunizacoes/1` ou `/pacientes/id_paciente/imunizacoes/2`). O método de inserção (POST) agendará a primeira dose (com o status de agendamento, mas sem lote e fabricante). Os métodos de atualização (PUT) serão para aplicar a primeira dose (o que já agenda a segunda, caso necessário, e altera as informações de lote e fabricante) e a segunda dose (alterando as informações de lote e fabricante também). Pode ser feito um método de atualização para a data de agendamento.

Para os casos em que as restrições forem violadas, utilizem os códigos de erro HTTP. Por exemplo, se tentar pegar dados de um recurso inexistente, retorne um erro 404. Se a requisição não tiver todos os dados necessários, retorne um erro 400 (Bad Request). Analise sempre cada caso e escolha o tipo de resposta que será enviado (a aula prática discute um pouco desse assunto).

Vocês podem utilizar como base a aula prática de webservices e o código exemplo gerado, que está junto com este trabalho. O prazo do trabalho passa a contar a partir de 24/07/2021 (data de seu lançamento). A data de entrega será 04/08/2021, 23:59, via Moodle.

Sumarizando:

- Data de disponibilização: 24/07/2021;
- Data de entrega: 04/08/2021, 23:59;
- Valor: peso 2 nos trabalhos práticos (essas atividades valem 60% da nota final, com peso total das atividades igual a 7);
- Individual, entrega via Moodle;
- Canais de atendimento: PAluno, Discord, Whatsapp e email.