# WebService para o Repositório IFC

**João Paulo Back**[1], **Luis Gustavo Block Nienkotter**[1], **Luis Filipe Pedroso**[1]

[1]Instituto Federal Catarinense – Campus Rio do Sul
R. Abraham Lincoln, 210 - Jardim América, Rio do Sul - SC, 89160-202

joaoback47@gmail.com, luis.block.nienk@gmail.com, luis_filipe42@outlook.com

***Abstract.*** *This article aims to present an API that uses the concept of REST (Representation State Transfer) architecture, which helps in storing articles for the repository of the Federal Institute of Santa Catarina. The article has a presentation of the routes and at the end a demonstration of them. At the end, a final conclusion about the work will be presented.*

***Resumo.*** *Este artigo visa a apresentação de uma API que utiliza o conceito de arquitetura REST (Representation State Transfer), que irá auxiliar no armazenamento de artigos para o repositório do Instituto Federal Catarinense. O artigo conta com uma apresentação das rotas criadas e ao final uma demonstração das mesmas. Ao final, será apresentado uma conclusão final sobre o trabalho.*

## 1. Introdução

Atualmente, serviços web são considerados como a forma promissora e mais rápida do desenvolvimento de aplicações web, tendo afetado diretamente na evolução do desenvolvimento de aplicações mobile, sendo esses dois uma combinação muito importante para a computação nos próximos anos ( PILIOURA, et al, 2002). Assim, devido a capacidade inferior de processamento dos dispositivos móveis, uma aplicação que adota somente uma interface amigável sem expor o usuário ao alto processamento de dados, é uma solução viável, e é onde os webservices podem ser utilizados.

Portanto, a utilização de webservices no lado do servidor é algo viável pois é uma plataforma que traz um alto encapsulamento de dados, além de um alto nível de portabilidade.

Por sua vez, este trabalho tem por objetivo a apresentação de uma API para o repositório do IFC. Em uma primeira abordagem, serão apresentados conceitos sobre o que é e do que é composto um Webservice REST. Em seguida serão expostos como estão organizados a estrutura das rotas da API e em seguida, os resultados de algumas das principais rotas da aplicação. Por fim, é realizado uma conclusão geral sobre o trabalho.

## 2. Webservices REST

Um WebService REST (Representation State Transfer) é um modelo de arquitetura, descrito por Roy Fielding (um dos criadores do HTTP), que combina alguns estilos arquiteturais, tais como o Cliente-servidor, o Sem Estado, o Cache, Interface Uniforme, dentre outros (RUBACK, 2009). De outra forma, a arquitetura REST pode ser considerada como um conjunto de princípios, que se utilizados corretamente, pode trazer diversos benefícios para uma aplicação web.

### 2.1. Elementos de uma Arquitetura REST

A arquitetura REST conta com diversos elementos, onde seu principal foco está nos papéis dos componentes, e não na implementação em sí. Abaixo, está descrito alguns dos principais elementos da arquitetura REST.

Recursos são a abstração chave de uma arquitetura REST. É através dele que qualquer informação pode ser representada, tais como imagens, documentos, ou qualquer outro tipo de dado. Assim, o REST utiliza de um identificador de recurso para realizar a identificação de um recurso específico em uma interação entre os componentes (RUBACK, 2009).

A representação é outro elemento da arquitetura REST. Sua principal função está na maneira de transferir a informação do estado atual de um recurso entre os componentes. Mais especificamente, uma representação é composta dos dados (sequência de bytes) e meta-dados utilizados para descrever estes dados. Outra informação presenta na representação é a informação de controle, que pré define uma mensagem entre os componentes, onde através disso, é possível parametrizar uma requisição o comportamento de alguns elementos. A representação também define o formato dos dados, ou media-type(RUBACK, 2009).

Já os conectores, são outros elementos que compôem a estrutura REST. Eles são responsáveis por interligar os elementos, encapsulando as atividades de acesso aos recursos e para transferir o estado de um componente (RUBACK, 2009). De acordo com Fielding, a utilização desses conectores provê uma interface abstrata para os componentes, o que deixa o sistema mais simples, separando suas responsabilidades e diminuindo o acoplamento.

### 2.2. Interface Uniforme

Para realizar a padronização das representações, a arquitetura REST utiliza uma interface padronizada entre os componentes. Na prática, essa interface é aplicada através dos métodos HTTP.

**Tabela 1. Métodos HTTP. Fonte: Autor.**

| Método HTTP | Objetivo |
|---|---|
| POST | Criação de um recurso |
| GET | Obter a representação de um recurso |
| PUT | Criar ou modificar recurso existente |
| DELETE | Remover recurso existente |

Para a utilização destes recursos, todas estas requisições devem ser realizadas por um cliente HTTP, através tendo assim as respostas solicitadas.

## 3. Desenvolvimento

Utilizando a arquitetura REST e os metodos padroes do HTTP foi então montado um banco de dados, que irá servir para guardar todos os dados que precisem ser salvos no repositório e que posteriormente serão utilizados pelos usuários.

Com os dados podendo ser salvos é possivel então criar um sistema que provem esses dados para os usuarios, para isso, será criado um serviço web.

Para cada tabela do banco de dados será criado o seu respectivo endpoint, endpoint seria a url que o usuário vai utilizar para ter acesso aos dados, alêm de endpoints mais especificos, que serão primariamente utilizados por outros programadores.

### 3.1. Banco de Dados

Para o banco de dados do repositório IFC foi utilizado o Postgres, o Postgres é um sistema de banco de dados relacional, código aberto que se utiliza da linguagem SQL para realizar suas operações (HESLEY, 2007).

Abaixo, na tabela 2, estão algumas das rotas presentes na API.

**Tabela 2. Rotas da API. Fonte: Autor.**

| Entidade | Método | URI |
|---|---|---|
| User | GET | /users |
| User | GET | /users/:id |
| UserStatistics | POST | /users/statistics |
| Course | GET | /courses |
| Course | GET | /courses/:id |
| Course | DELETE | /courses/:id |
| Article | GET | /articles |
| Article | POST | /articles |
| ArticleStatistics | POST | /articles/statistics |
| Event | GET | /events |
| Event | GET | /events/:year/years |
| Event | POST | /events |
| UserArticles | DELETE | /events/:id |
| CourseArticles | DELETE | /events/:id |
| Charts | GET | /charts/articles |
| Charts | GET | /charts/articles/years |
| Charts | GET | /charts/courses/views |
| Charts | GET | /charts/courses/years |

## 4. Resultados e Discussões

Para realizar a discussão dos resultados, foram realizados quatro principais testes: o primeiro teste foi realizado na rota /articles, o segundo, na rota /events, o terceiro na rota /users, e o quarto na rota /courses. Abaixo, figuras das saídas das rotas citadas.

```
[
  {
    "id": 1,
    "title": "Brain–computer interfaces for dissecting cognitive processes underlying sensorimotor control",
    "abstract": "Objective: The goal of the research is to develop three serious games using design for the older adult population, in order to evaluate the cognitive performance of older adults.",
    "altabstract": "Lorem ipsum dolor",
    "keywords": "T1, T2, T3",
    "tags": "HCI",
    "year": 2019,
    "date": "2019-11-01T00:00:00.000Z",
    "language": "pt-br",
    "type": "ARTIGO_CIENTIFICO",
    "aproved": false,
    "url": "",
    "event_id": 1,
    "createdAt": "2019-11-26T11:15:09.592Z",
    "updatedAt": "2019-11-26T11:15:09.592Z",
    "file_id": null,
    "ArticlesStatistic": null,
    "Event": {
      "id": 1,
      "name": "ERES",
      "year": 2019,
      "url": "xxx",
      "createdAt": "2019-11-26T11:14:56.575Z",
      "updatedAt": "2019-11-26T11:14:56.575Z"
    },
    "File": null
  }
]
```


**Figura 1. Saída da API na rota /articles. Fonte: Autor.**


```
[
  {
    "id": 1,
    "name": "ERES",
    "year": 2019,
    "url": "xxx",
    "createdAt": "2019-11-26T11:14:56.575Z",
    "updatedAt": "2019-11-26T11:14:56.575Z"
  },
  {
    "id": 2,
    "name": "4C",
    "year": 2019,
    "url": "xxx",
    "createdAt": "2019-11-26T11:15:03.584Z",
    "updatedAt": "2019-11-26T11:15:03.584Z"
  }
]
```


**Figura 2. Saída da API na rota /events. Fonte: Autor.**

```
[
  {
    "id": 5,
    "registration_id": "20151070020434",
    "username": "teste6",
    "password_hash": "$2a$08$KZaZagZW9F1FV/AjEP3jg.YqON8ZdfE3TG9KWmnoz1QjDKkVZ5LYS",
    "firstname": "Teste 1",
    "lastname": "Teste",
    "type": "ALUNO",
    "siape": 1245,
    "lattes": "Bacharelado em Ciências da computação",
    "email": "teste6@outlook.com",
    "birthday": "1997-09-02T00:00:00.000Z",
    "aproved": null,
    "createdAt": "2019-12-04T16:54:50.424Z",
    "updatedAt": "2019-12-04T16:54:50.424Z",
    "file_id": null,
    "UsersStatistic": null,
    "File": null
  },
  {
    "id": 4,
    "registration_id": "20151070020434",
    "username": "teste5",
    "password_hash": "$2a$08$eCiYeQB4CsyXrtuJvakxc.ED0T04wx/OXkEjyuwr58scyXxL8GGx2",
    "firstname": "Teste 1",
    "lastname": "Teste",
    "type": "ALUNO",
    "siape": 1245,
    "lattes": "Bacharelado em Ciências da computação",
    "email": "teste5@outlook.com",
    "birthday": "1997-09-02T00:00:00.000Z",
    "aproved": null,
    "createdAt": "2019-12-04T16:46:52.715Z",
    "updatedAt": "2019-12-04T16:46:52.715Z",
    "file_id": null,
    "UsersStatistic": null,
    "File": null
  },
```


**Figura 3. Saída da API na rota /users. Fonte: Autor.**


```
[
  {
    "id": 1,
    "name": "Eng Software",
    "createdAt": "2019-11-26T11:14:44.800Z",
    "updatedAt": "2019-11-26T11:14:44.800Z"
  },
  {
    "id": 2,
    "name": "BCC",
    "createdAt": "2019-11-26T11:14:48.651Z",
    "updatedAt": "2019-11-26T11:14:48.651Z"
  },
  {
    "id": 3,
    "name": "BSI",
    "createdAt": "2019-11-26T11:14:51.937Z",
    "updatedAt": "2019-11-26T11:14:51.937Z"
  }
]
```


**Figura 4. Saída da API na rota /courses. Fonte: Autor.**

## 5. Conclusão

A finalidade da criação desse serviço web é de que caso, no futuro, outras pessoas, ou até mesmo outros programadores queiram ter acesso aos artigos publicados no repositório, ou a informações de eventos, eles tenham uma forma de conseguir esses dados sem precisar criar suas proprias API's, ou precisem ficar coletando de outras fontes.

Alem de, é claro, formar uma fonte centralizada desses dados, fazendo com que qualquer pesquisa se torne muito mais fácil.

Se tem a convicção de que provavelmente os dados dos artigos, ou até possivelmente os dos eventos serão os mais acessados, porem para fins de desenvolvimento do site do repositório, outros endpoints foram criados para facilitar a vida dos desenvolvedores, e calculos mais complexos não sejam feitos pela maquina do cliente e sim pelo servidor.

## Referências


DevMedia (2007). Introdução ao postgre sql. https://www.devmedia.com.br/introducao-ao-postgresql/6390, dez/2019.

DevMedia (2009). Arquitetura rest: Uma alternativa para construção de serviços web, dez/2019.

PILIOURA, T.; TSALGATIDOU, A. H. S. (2002). Scenarios of using web services in m-commerce. ACM SIGecom Exchanges Volume 3, Issue 4 Winter, Mobile commerce, p. 28-36, Dec.

Rodrigues, L. C. R. (2009). Arquitetura rest. Universidade Federal de Juiz de Fora.