

Em Busca de uma API REST para um Sistema Acadêmico de Terceiros

Ana Lúcia da Silva Ferreira¹, Jalves Mendonça Nicacio², Glauber Ventura Ferreira³, Greyson Mascarenhas Santos Filho³, Vanderson da Silva Rodrigues¹

¹Instituto de Computação – Universidade Federal de Alagoas (UFAL)
Av. Lourival Melo Mota, s/n Tabuleiro do Martins – 57072-900 – Maceió – AL

²Campus São Miguel dos Campos – Instituto Federal de Alagoas (IFAL)
Rua Visconde de Sinimbu, 32, Centro – 57240-000 – São Miguel dos Campos – AL

³Campus Palmeira dos Índios – Instituto Federal de Alagoas (IFAL)
Av. Alagoas, s/n Palmeira de Fora – 57608-180 – Palmeira dos Índios – AL
{dasilvaferreira.af, greysonmrx, vanderson99rodrigues}@gmail.com,
{jalves.nicacio, glauber.ferreira}@ifal.edu.br

Abstract. *This paper presents the roadmap towards the development of an API for searching a third party Academic System (AS) deployed on Federal Institute of Alagoas. The main motivation for this development is that although academic data is directly available through database queries, due to the complexity of AS, the learning curve is high for developing new projects. Such an API will allow students, teachers, technicians and IT system analysts to develop software that use these academic data, and then, do research, development, and innovation projects, promoting the production of innovative scientific and technological knowledge.*

Resumo. *Neste artigo apresentam-se os caminhos seguidos em busca do desenvolvimento de uma API de consulta de dados no Sistema Acadêmico (SA) de terceiros implantado no Instituto Federal de Alagoas. A principal razão para o desenvolvimento desta API é que, apesar dos dados estarem acessíveis através de consultas diretas ao banco de dados, em virtude da complexidade do SA, a curva de aprendizagem é grande para o desenvolvimento de novos projetos. Essa API permitirá que alunos, professores, técnicos e analistas de TI possam desenvolver software que utilize esses dados e, com isso, executar projetos de Pesquisa, Desenvolvimento e Inovação em diferentes áreas, estimulando a produção de conhecimento científico e tecnológico inovadores.*

1. Introdução

Há alguns atrás o Instituto Federal de Alagoas (IFAL) iniciou o processo de implantação de um Sistema Acadêmico (SA) desenvolvido por terceiros. Dentre as motivações para a contratação e utilização desse sistema, podem ser citadas:

1. **Padronização** de uso em todos os *campi* do IFAL, uma vez que antes do processo de implantação diferentes sistemas que possuíam o mesmo propósito eram utilizados por diferentes *campi*.

2. **Uniformização** de uso nos diferentes **níveis de ensino** ofertados pelo IFAL, visto que anteriormente utilizavam-se sistemas diferentes para cada um dos níveis.

As diversas operações utilizadas pelos usuários diariamente ao utilizarem o SA geram uma imensa quantidade de dados, de diferentes naturezas. Esses dados são de propriedade do IFAL, possuem um imenso valor para a obtenção de novas informações e a instituição poderia utilizá-los como entrada para a realização de projetos de Pesquisa, Desenvolvimento e Inovação – PDI em diferentes áreas. Se todos esses projetos pudessem ser desenvolvidos internamente por pesquisadores do próprio IFAL, através do corpo técnico relacionado à área de TI, haveria uma liberdade maior para analisar os dados de maneira exploratória, fazer pesquisa científica e tecnológica, permitindo o desenvolvimento de novos produtos e serviços inovadores.

É importante salientar que desenvolver projetos de PDI internamente na instituição utilizando os dados dos SA é possível atualmente com a infraestrutura de software implantada. Entretanto, é uma tarefa extremamente custosa e improdutiva, uma vez que as equipes de cada um desses projetos precisariam compreender, dominar e acessar diretamente o modelo de dados relacional do sistema, composto por inúmeras tabelas. Além disso, a equipe de TI do IFAL precisaria definir mecanismos de autenticação e autorização para delimitar o acesso aos dados para cada projeto específico.

Em face dessas dificuldades, surgiu a motivação para o desenvolvimento de *Web Services* [WORLD WIDE WEB CONSORTIUM 2004], mais especificamente através de uma API REST [FIELDING 2000], para possibilitar a consulta de dados de atividades acadêmicas gerenciados pelo SA. Essa API pavimentará o acesso às informações, possibilitando que diversos projetos de PDI com dados acadêmicos sejam desenvolvidos de maneira mais eficiente e produtiva.

2. Decisões Arquiteturais e Tecnológicas

Uma das primeiras decisões que precisaram ser tomadas pela equipe ao iniciar o desenvolvimento da API foi a respeito do formato de arquivo utilizado para disponibilizar os dados acadêmicos. Optou-se pelo formato aberto JSON (*JavaScript Object Notation*) em virtude da sua ampla utilização e consequente suporte ferramental.

Com o objetivo de padronizar a estrutura dos arquivos JSON trocados entre a API e as aplicações cliente, assim como padronizar o mecanismo utilizado para criação, atualização, exclusão, relações, filtragem, ordenação e paginação de recursos, decidiu-se seguir a especificação JSON API [JSON API 2013]. Aderindo a esta especificação, o desenvolvimento da API é facilitado pois a equipe pode se concentrar diretamente nas entidades disponibilizadas pela própria API, uma vez que várias demandas comuns a toda API já são resolvidas pela própria especificação.

Dentre as diversas implementações disponíveis da especificação JSON API, considerando que a API do SA está sendo implementada utilizando o arcabouço Java/Spring Boot [SPRING 2005], após estudos e testes optou-se pela utilização da biblioteca Katharsis [READ THE DOCS 2017], uma vez que ela provê nativamente integração com o Spring Boot para receber/responder requisições de acordo com a especificação JSON API.

Já com relação à camada de dados, o SA utiliza o banco de dados relacional PostgreSQL. O esquema do banco de dados do SA foi então estudado e sete tabelas foram selecionadas inicialmente para consulta: *pessoa*, *discente*, *curso*, *componente_curricular*, *situacao_matricula*, *matricula_componente*, e *componente_curricular_detalhes*. A seleção dessas tabelas se deu em virtude de elas armazenarem dados relativos a uma parte do escopo definido inicialmente para o desenvolvimento da API, a saber: aluno, disciplina e curso. Os dados referentes às notas dos alunos estão em fase de implementação.

Levando em consideração as tabelas citadas anteriormente, realizou-se então a especificação, o projeto e a implementação da API de consulta. Para cada uma das sete tabelas, mapeou-se um recurso REST correspondente. Por exemplo, para a tabela *discente* criou-se uma classe Java com o nome *Discente*, contendo a anotação `@JsonApiResource(type="discentes")`, responsável por representar os dados dos discentes que poderão ser consultados através da API. Através da criação de uma classe de nome *DiscenteResourceRepository*, na qual são implementados os métodos de consulta por discentes, é possível, por exemplo, acessar a seguinte URL através de um método HTTP GET para consultar o discente que possui a matrícula 123456 de acordo com a especificação da JSON API: `/api/discentes/?filter[matricula]=123456`.

Na Figura 1 ilustra-se o conteúdo da resposta, em conformidade com a especificação JSON API, para a requisição da listagem de todos os discentes cadastrados no SA utilizando a API REST. Por questões de sigilo, são apresentados dados de teste. Entretanto, os atributos exibidos no arquivo JSON de resposta condizem com os dados obtidos do banco relacional do SA. A imagem da Figura 1 foi capturada da ferramenta Postman [POSTMAN], utilizada durante o desenvolvimento para realizar consultas e verificar as respostas em formato JSON.



```
1 - {
2   "data": [
3     {
4       "id": "1",
5       "type": "alunos",
6       "attributes": {
7         "matricula": "123",
8         "perEntrada": "2014.1",
9         "nome": "Ana",
10        "nivel": "4",
11        "email": "ana@hotmail.com",
12        "status": "ativo"
13      },
14      "links": {
15        "self": "http://localhost:8080/api/alunos/1"
16      }
17    },
18    {
19      "id": "2",
20      "type": "alunos",
21      "attributes": {
22        "matricula": "321",
23        "perEntrada": "2013.1",
24        "nome": "Ze",
25        "nivel": "3",
26        "email": "ze@hotmail.com",
27        "status": "ativo"
28      },
29      "links": {
30        "self": "http://localhost:8080/api/alunos/2"
31      }
32    }
33  ]
34 }
```

Figura 1. Resposta em formato JSON para a listagem de alunos do SA

Considerando que o objetivo da API é possibilitar a *consulta* de dados, apenas o método HTTP GET (recuperação de recursos) é de fato implementado. Os outros métodos HTTP utilizados na APIs REST (POST para a criação de recursos, PUT/PATCH para atualização e DELETE para exclusão) não são permitidos, fazendo com que um código de erro seja retornado para a aplicação cliente.

3. Prototipagem de uma Aplicação para Validação da API

Com o objetivo de validar a API em desenvolvimento, decidiu-se implementar uma aplicação protótipo, direcionada para os Coordenadores de Curso do IFAL, que permite a visualização das notas de um determinado aluno em um formato de Gráfico de Radar. A ideia é que a partir de uma única visualização gráfica o coordenador possa ter uma visão ampla do desempenho quantitativo do aluno. Na Figura 2 ilustra-se um exemplo deste gráfico para um determinado aluno, com valores fictícios de nota.

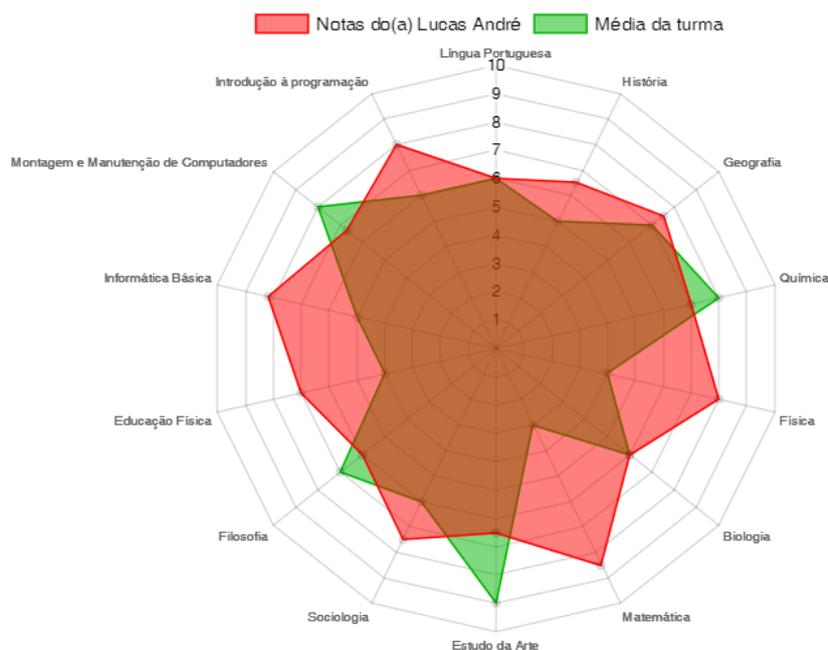


Figura 2. Gráfico de radar ilustrando as médias de um aluno e da sua turma em diferentes disciplinas

Após uma busca pelo aluno utilizando diferentes critérios, a aplicação protótipo exibirá o gráfico de radar, no qual cada disciplina em que o aluno está matriculado será visualizada em um dos seus vértices. Além destas informações, exibe-se também uma curva com as notas do aluno (rosa) e outra curva utilizada para comparação com a média das notas da turma (verde). É possível identificar no gráfico, por meio das situações em que, a partir do centro, a curva da média da turma ultrapassa a curva das notas do aluno, que o aluno obteve um desempenho abaixo da média da turma nas disciplinas de Química, Estudo da Arte, Filosofia e Montagem e Manutenção de Computadores.

O gráfico ilustrado na Figura 2 foi gerado dinamicamente através da aplicação protótipo que está sendo desenvolvida seguindo a estrutura de uma SPA (*Single Page Application*). Para isso, utiliza o arcabouço JavaScript Vue.js [VUE.JS 2014] no lado cliente em conjunto com a biblioteca JavaScript Chart.js [CHART.JS] para efetuar a geração dos gráficos. Até o momento da escrita deste artigo, esta aplicação protótipo ainda não está integrada completamente com a API REST. Por este motivo, o conjunto de dados exibidos no gráfico relativos ao aluno, às disciplinas, e às notas estão inseridos estaticamente na aplicação.

Tão logo a integração entre a aplicação e a API esteja finalizado, o funcionamento ocorrerá da maneira descrita a seguir. Os dados do aluno (nome,

disciplinas, notas) serão recuperados do lado servidor através do envio de arquivos JSON pela API REST; estes dados serão complementados também com uma categorização das disciplinas de acordo as Áreas do Conhecimento utilizadas no ENEM (Matemática e Suas Tecnologias, Ciências Humanas e Suas Tecnologias, etc.), de maneira que o coordenador possa ter uma visão mais clara caso o aluno esteja eventualmente tendo dificuldades em alguma área do conhecimento específica.

4. Considerações Finais e Perspectivas Futuras

Neste artigo apresentaram-se os caminhos tomados em busca da especificação, do projeto e da implementação de uma API de consulta para facilitar o acesso aos dados de atividades acadêmicas gerenciadas pelo SA de terceiros utilizado no IFAL. Pretende-se com a disponibilização desta API que toda a comunidade da instituição seja beneficiada, uma vez que, dependendo da natureza dos projetos de PDI desenvolvidos utilizando a API, pais, alunos, professores, técnicos, gestores acadêmicos, etc. poderão usufruir das informações obtidas.

Além das decisões arquiteturais e tecnológicas tomadas ao longo do desenvolvimento, descreveu-se também o funcionamento da aplicação protótipo que está sendo implementada com o intuito de validar os artefatos de software produzidos pela API. Sua principal funcionalidade é exibir uma visão quantitativa unificada das notas de um determinado aluno para os coordenadores de curso.

Os próximos passos para evolução da API REST dizem respeito à finalização da implementação de consultas de notas dos alunos e posteriormente à inclusão de informações de frequência dos alunos e, em seguida, de consulta de professores. Já com relação à aplicação, pretende-se realizar as seguintes atividades: finalizar a integração com a API, adicionar informações de frequência na visualização gráfica para que se possa verificar a correlação entre notas e frequência, permitindo a identificação de situações que antecipem eventuais evasões.

Uma funcionalidade transversal que precisa necessariamente ser incorporada na API antes da sua efetiva disponibilização para a comunidade acadêmica é um mecanismo de autorização, uma vez que é inconcebível do ponto de vista da segurança da informação expor os dados acadêmicos sem um mecanismo robusto para garantir que apenas as pessoas autorizadas possam acessá-los. Um estudo preliminar já foi realizado acerca do arcabouço OAuth 2.0 [ELOY 2017], com um foco especial no *grant type Authorization Code*, que deve ser utilizado na API como mecanismo de autorização. Além disso, o protocolo OAuth 2.0 “obriga que todas as requisições sejam feitas utilizando uma camada de transporte segura através de *Transport Layer Security* (TLS) ou *Secure Socket Layer* (SSL)” [ELOY 2017].

Com relação à segurança da informação por parte dos desenvolvedores de software que utilizarão a API, pretende-se disponibilizar um ambiente de homologação com uma base de dados de teste para que as aplicações implementadas possam ser testadas antes de serem implantadas no ambiente de produção com os dados reais. Isto evitará que os dados reais sejam coletados de forma indevida. Uma vez em ambiente de produção, apenas os usuários autorizados (e.g. alunos, professores, coordenadores, etc.) mediante login e senha poderão acessar os seus respectivos dados.

Um outro trabalho que poderá ser desenvolvido é um estudo comparativo para analisar o processo de desenvolvimento de uma mesma aplicação de dados acadêmicos

com e sem a utilização da API. Sem a API, os dados seriam acessados através de consultas ao banco de dados. O intuito do estudo é medir os ganhos de produtividade da equipe de desenvolvimento ao utilizar a API em detrimento do acesso direto ao banco.

Por fim, é importante destacar que ao tempo que esta API estiver disponibilizada para uso, diferentes aplicações e projetos poderão ser desenvolvidos, incluindo, mas não se limitando a: painéis de indicadores gerenciais (*dashboards*); mineração de dados educacionais [ROMERO e VENTURA 2010]; ferramentas para auxiliar gestores na tomada de decisão; integração com aplicações educacionais de terceiros, tais como Kahoot [KAHOOT! AS 2013], Socrative [MASTERYCONNECT 2014], Google Classroom [GOOGLE 2014], Gradepen [GRADEPEN CORPORATION 2014]; novas interfaces gráficas para acesso a funcionalidades já existentes; desenvolvimento de novas aplicações.

Referências

- CHART.JS. Open source HTML5 Charts for your website. Disponível em: <<https://www.chartjs.org/>>. Acesso em: 27 jul 2018.
- ELOY, A. OAuth 2.0: Proteja suas aplicações com Spring Security OAuth2. São Paulo: Casa do Código, 2017.
- FIELDING, R. T. Architectural Styles and the Design of Network-based Software Architectures. Irvine, CA, EUA. 2000. Tese de Doutorado.
- GOOGLE. Google Classroom. 2014. Disponível em: <<https://classroom.google.com/>>. Acesso em: 27 jul 2018.
- GRADEPEN CORPORATION. Gradepen. 2014. Disponível em: <<https://gradepen.com/>>. Acesso em: 27 jul 2018.
- JSON API. JSON API - A specification for building APIs in JSON. 2013. Disponível em: <<http://jsonapi.org/>>. Acesso em: 27 jul 2018.
- KAHOOT! Play Kahoot! 2013. Disponível em: <<https://kahoot.it/>>. Acesso em: 27 jul 2018.
- MASTERYCONNECT. Socrative. 2014. Disponível em: <<https://www.socrative.com/>>. Acesso em: 27 jul 2018.
- POSTMAN. API Development Environment. Disponível em: <<https://www.getpostman.com/>>. Acesso em: 27 jul 2018.
- READ THE DOCS. Katharsis json-api documentation. 2017. Disponível em: <<http://katharsis-jsonapi.readthedocs.io/en/latest/>>. Acesso em: 27 jul 2018.
- SPRING. Spring Boot. 2005. Disponível em: <<https://spring.io/>>. Acesso em: 27 jul 2018.
- ROMERO, C.; VENTURA, S. Educational Data Mining: A Review of the State of the Art. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), v. 40, n. 6, p. 601 - 618, Nov 2010.
- VUE.JS. The Progressive JavaScript Framework. 2014. Disponível em: <<https://vuejs.org/>>. Acesso em: 27 jul 2018.