

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE – UERN
FACULDADE DE CIÊNCIAS EXATAS E NATURAIS – FANAT
DEPARTAMENTO DE INFORMÁTICA – DI

**API MOSSORÓ+: API REST PARA UMA PLATAFORMA VOLTADA À
CENTRALIZAÇÃO DE EVENTOS EDUCACIONAIS**

MOSSORÓ - RN

ABRIL/2021

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE – UERN
FACULDADE DE CIÊNCIAS EXATAS E NATURAIS – FANAT
DEPARTAMENTO DE INFORMÁTICA – DI

LUIZ CARLOS MOURA REGES

**API MOSSORÓ+: API REST PARA UMA PLATAFORMA VOLTADA À
CENTRALIZAÇÃO DE EVENTOS EDUCACIONAIS**

Relatório apresentado ao curso de Ciência da Computação da Universidade do Estado do Rio Grande no Norte como requisito da disciplina de Trabalho de Diplomação, sob a orientação da Profa. Dra. Cícilia Raquel Maia Leite e coorientação de Exlley Clemente dos Santos.

MOSSORÓ - RN

ABRIL/2021

LUIZ CARLOS MOURA REGES

**API MOSSORÓ+: API REST PARA UMA PLATAFORMA VOLTADA À
CENTRALIZAÇÃO DE EVENTOS EDUCACIONAIS**

Relatório apresentado como pré-requisito para obtenção do título de Bacharel em Ciência da Computação da Universidade do Estado do Rio Grande do Norte – UERN, submetida à aprovação da banca examinadora composta pelos seguintes membros:

Aprovado em: XX / XX / XXXX

Banca Examinadora

Profª. D.Sc Cícilia Raquel Maia Leite (Orientadora)
Universidade do Estado do Rio Grande do Norte – UERN

Exlley Clemente dos Santos (Coorientador)
Universidade do Estado do Rio Grande do Norte – UERN

Prof. D.Sc Maximiliano Araújo da Silva Lopes
Universidade do Estado do Rio Grande do Norte – UERN

Profª. M.Sc. Cynthia Moreira Maia
Universidade Federal de Pernambuco – UFPE

SUMÁRIO

INTRODUÇÃO	5
OBJETIVO	5
METODOLOGIA	5
API MOSSORÓ+: API REST PARA UMA PLATAFORMA VOLTADA À CENTRALIZAÇÃO DE EVENTOS EDUCACIONAIS	8
CONSIDERAÇÕES FINAIS	13
REFERÊNCIAS	14

1. INTRODUÇÃO

É indiscutível que a participação de estudantes em eventos acadêmicos (congressos, feiras científicas, etc.) é importante para a sua formação dada a possibilidade de convívio, a troca de ideias e experiências, tudo isso em consonância com os conhecimentos obtidos em sala de aula. Isso porque os eventos acadêmicos/científicos reúnem uma variedade de pessoas interessadas em dividir e adquirir conhecimentos em determinada área (MARCHIORI, *et. al*, 2006).

Dada essa importância, foi proposta a plataforma Web denominada de Mossoró+ com o objetivo de centralizar a divulgação de eventos acadêmicos/científicos na cidade de Mossoró/RN. Assim, visando as especificidades desta plataforma, utiliza-se de uma API (do inglês, *Application Programming Interface*) REST (do inglês, *Representational State Transfer*) com o objetivo de facilitar a comunicação da aplicação *back-end* com diferentes representações da aplicação *front-end*.

2. OBJETIVO

Desenvolvimento de uma API baseada no modelo arquitetural REST e denominada API MOSSORÓ+, capaz de viabilizar a centralização de eventos acadêmicos/científicos e gerenciar as informações geradas no banco de dados.

3. METODOLOGIA

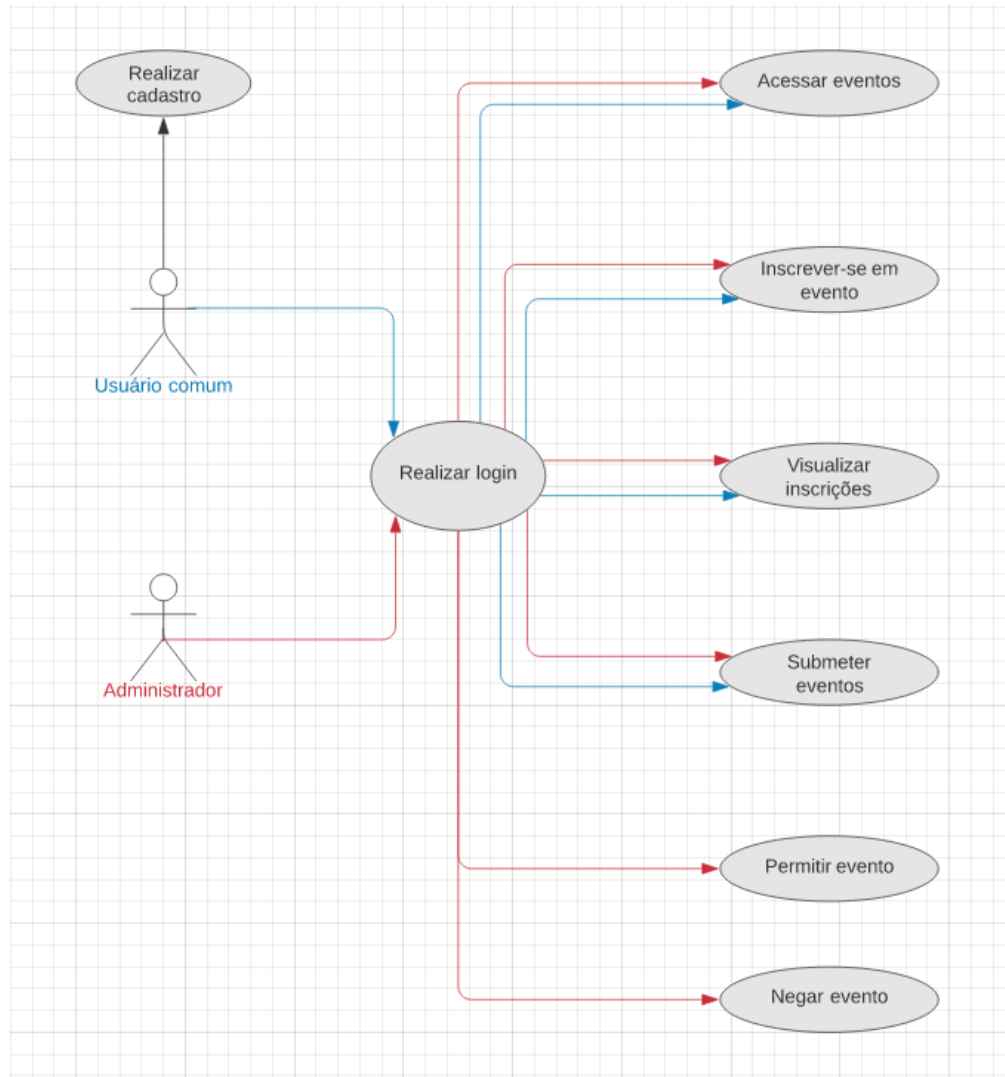
Para a condução desta pesquisa, utilizou-se de uma metodologia dividida em 5 etapas, conduzidas como se segue:

A primeira etapa consistiu na revisão bibliográfica dos principais conceitos desta pesquisa, buscando-se a formulação teórica necessária para a condução das fases subsequentes.

Já a etapa dois tratou do levantamento de requisitos, que se deu apoiado na aplicação de um questionário, em um grupo de estudantes de graduação. O questionário constitui-se de seis questões objetivas e uma questão aberta, e foi aplicado de forma online, durante o período de 26/10/2020 a 04/11/2020, o qual obteve 92 respostas e abordou questões sobre a necessidade de participação de alunos em eventos acadêmicos/educacionais.

Na etapa três, com base nas informações resultantes das etapas anteriores, foi possível modelar e especificar o sistema. O diagrama de casos de uso ilustrado na Figura 1, descreve as principais funções da API.

Figura 1: Diagrama de Caso de Uso.



Fonte: Autoria Própria (2021).

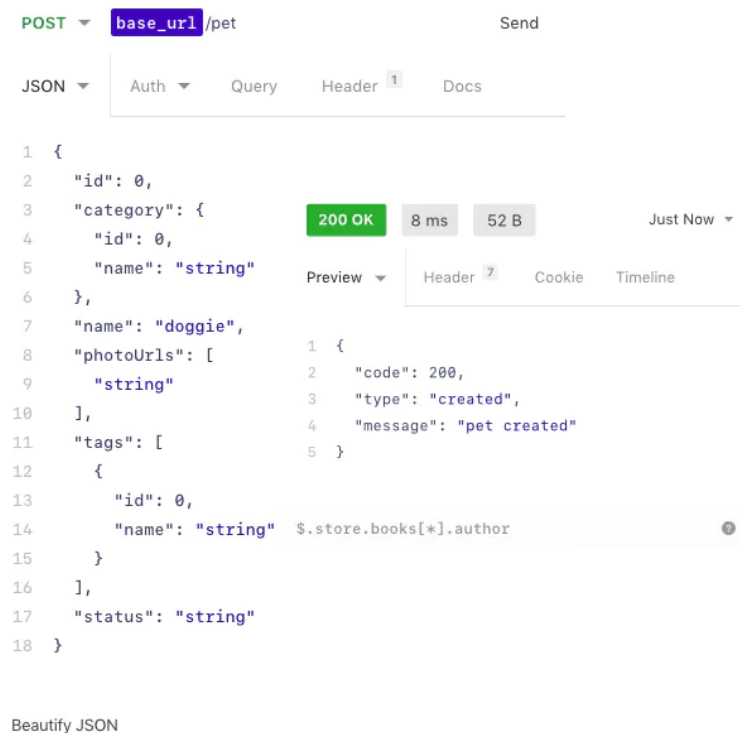
A etapa quatro consistiu no desenvolvimento da API, que se deu utilizando a linguagem JavaScript® (JAVASCRIPT, 2021). Segundo a MDN Web Docs, plataforma colaborativa para desenvolvedores do Mozilla Firefox, JavaScript é uma linguagem leve, interpretada e baseada em objetos com funções de primeira classe, mais conhecida como a linguagem de script para páginas Web.

Utilizou-se também do NodeJS, um ambiente de execução Javascript baseado em eventos assíncronos, projetado para construir aplicativos de rede escalonáveis (OPENJS,

2021); o ExpressJS, um *framework* Web rápido, flexível e minimalista para Node.js (FOUNDATION, 2021); e o MongoDB, um sistema de gerenciamento de banco de dados que suporta naturalmente JSON e possui linguagem de consulta expressiva simples (MONGODB, 2021).

Na etapa cinco, buscando a validação da API, foram realizados testes através do *software* Insomnia, que serve para automatizar testes manuais de APIs. O Insomnia possui uma interface amigável e intuitiva, que oferece facilidade na realização de testes em APIs, como ilustra a Figura 2.

Figura 2: Interface Insomnia.



Fonte: Kong (2021).

Desse modo, a API desenvolvida é intuitiva e de fácil utilização, para que possa ser utilizada no gerenciamento de bases de dados voltadas à centralização de eventos acadêmicos.

4. API MOSSORÓ+: API REST PARA UMA PLATAFORMA VOLTADA À CENTRALIZAÇÃO DE EVENTOS EDUCACIONAIS

Este trabalho apresenta uma *Application Programming Interface* (API) para gerir as regras de negócio de uma plataforma para centralização de eventos educacionais, denominada Mossoró+, desenvolvida por Morais (2020). O sistema desenvolvido pelo autor consiste em uma plataforma Web a qual permite que os usuários acessem listas de eventos acadêmicos disponíveis e, caso desejem, criem um evento, ou realizem a inscrição em um.

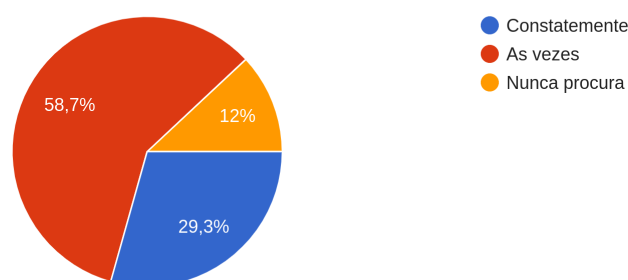
A API foi desenvolvida utilizando do modelo arquitetural REST, do inglês *Representational State Transfer*. O estilo de arquitetura REST é comumente aplicado ao design de APIs para serviços web, este modelo se preocupa com questões como: quando os segmentos de caminho de URI devem ser nomeados com substantivos no plural; o método de solicitação usado para atualizar o estado do recurso; o código de status de resposta HTTP apropriado para um determinado cenário; entre outras (MASSE, 2011).

Para delimitação das funcionalidades e levantamento inicial de requisitos, nos apoiamos nas respostas obtidas no questionário mencionado na Seção 2.

Quando perguntados sobre a frequência com que buscam por eventos acadêmicos, 81 participantes (88%) responderam que buscam por eventos educacionais às vezes ou constantemente (Figura 3).

Figura 3: Respostas dos participantes quanto a frequência de busca de eventos.

Com qual frequência você procura por eventos acadêmicos/educacionais?
92 respostas

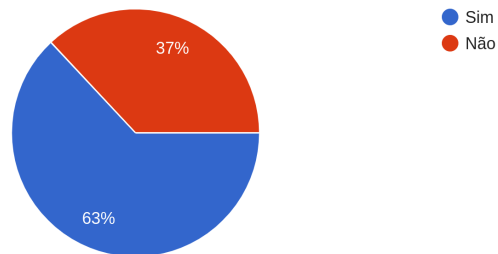


Fonte: Autoria Própria (2020).

Quanto à dificuldade de encontrar eventos, obteve-se os resultados ilustrados na Figura 4. Quando perguntados se já tiveram dificuldades em encontrar eventos educacionais, 58 participantes (63%) responderam que tiveram dificuldades em encontrar eventos.

Figura 4: Respostas dos participantes quanto a dificuldade de localização de eventos.

Caso já tenha tido a necessidade de participar de um evento, teve dificuldade de encontrar?
92 respostas

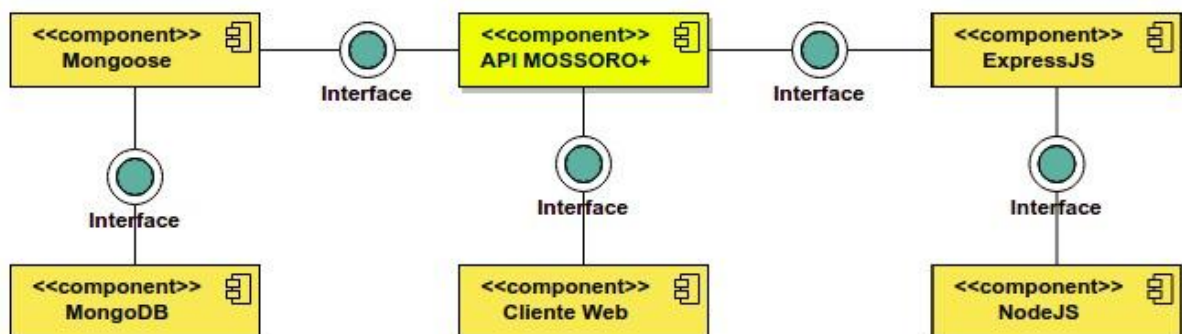


Fonte: Autorial Própria (2020).

Além destas, uma questão aberta foi aplicada a fim de obter dos participantes sugestões de funcionalidades para um sistema para centralização de eventos acadêmicos. Dentre as principais respostas, obteve-se: mostrar eventos próximos da região; linguagem acessível; clareza; e facilidade na inscrição.

O diagrama de componentes ilustrado na Figura 5, apresenta a arquitetura da API, que funciona como intermediadora na oferta de serviços requeridos pelo cliente Web, isto é, o cliente web faz requisições a API, e esta, rodando sob o NodeJS com intermédio do ExpressJS, é responsável por estabelecer as regras de negócio nas consultas a base de dados.

Figura 5: Diagrama de Componentes.



Fonte: Autorial Própria (2021).

Dentre os principais recursos oferecidos pela API, estão:

- A autenticação de usuários;
- Criação, retorno, atualização e destruição de uma instância “Evento”;

- Criação, retorno, atualização e destruição de uma instância “Inscrição”.

Essas funções são fornecidas através de rotas, pontos de acesso disponibilizados por uma API para que a aplicação cliente possa consumir os serviços providos, sendo classificadas em recursos como os principais tópicos disponibilizados (SOARES, 2019). Na API, são enviadas requisições HTTP, que retornam códigos de *status* de acordo com o seu retorno, especificados em Fielding (1999).

A autenticação na API é feita com base em JSON Web Token (JWT), uma codificação de token de segurança baseado em *JavaScript Object Notation* (JSON) que permite que informações de identidade e segurança sejam compartilhadas em domínios de segurança (JONES, CAMPBELL E MORTIMORE, 2015). Abaixo, a especificação da rota de autenticação.

POST /auth : Retorna um token de autenticação.

Status de resposta:

- 200: Login feito com sucesso;
- 404: Usuário não encontrado;
- 401: Senha incorreta;
- 500: API encontrou uma situação com a qual não sabe lidar.

A rota Eventos trata das principais funções relacionadas a eventos acadêmicos. Abaixo alguns exemplos de funcionamento da rota.

GET /events : Retorna todos os eventos

Status de resposta:

- 200: Nenhum erro ocorrido;
- 500: API encontrou uma situação com a qual não sabe lidar;

POST /events : Cria um evento

Status de resposta:

- 201: Nenhum erro ocorrido e o recurso foi criado;
- 400: Campos inválidos passados no payload (body);
- 500: API encontrou uma situação com a qual não sabe lidar;

Observação: Um evento sempre é criado com os status "isAccepted" e "isFinished" com valor "false".

PUT /events/:eventId : Atualiza um evento

Status de resposta:

- 200: Nenhum erro ocorrido e o recurso foi criado;
- 400: Campos inválidos passados no payload(body);
- 404: Evento não foi encontrada;
- 500: API encontrou uma situação com a qual não sabe lidar;

Observação: Os status "isAccepted" e "isFinished" não são atualizados.

PATCH /events/:eventId : Modifica os status de um evento

Status de resposta:

- 200: Nenhum erro ocorrido e o recurso foi criado;
- 400: Campos inválidos passados no payload(body);
- 404: Evento não foi encontrada;
- 500: API encontrou uma situação com a qual não sabe lidar.

Observação: somente os booleans "isAccepted" e "isFinished" são aceitos como payload.

DELETE /events/:eventId : Deleta um evento

Status de resposta:

- 200: Nenhum erro ocorrido e o recurso foi criado;
- 404: Evento não foi encontrada;
- 500: API encontrou uma situação com a qual não sabe lidar.

Observação: Um evento que está ocorrendo, ou seja, a data atual está entre o início e o fim do evento, não pode ser excluído.

A rota de inscrições permite ao usuário autenticado realizar ou cancelar sua inscrição em eventos.

POST /subscriptions : Cria uma inscrição num evento

Status de resposta:

- 201: Nenhum erro ocorrido e o recurso foi criado;
- 400: Campos inválidos passados no payload(body) ou evento já iniciado;
- 409: Usuário já está inscrito no evento;
- 404: Usuário ou evento não existem ou evento já encerrado;

- 500: API encontrou uma situação com a qual não sabe lidar;

DELETE /subscriptions/:subscriptionId : Deleta uma inscrição

Status de resposta:

- 200: Nenhum erro ocorrido e o recurso foi criado;

- 400: Evento já está em andamento;

- 404: Evento não foi encontrada;

- 500: API encontrou uma situação com a qual não sabe lidar.

Observação: Uma inscrição num evento que está ocorrendo, ou seja, a data atual está entre o início e o fim do evento, não pode ser excluída.

A Figura 6 apresenta um trecho de código no qual mostra a validação de um objeto JSON na criação de um evento.

Figura 6: Código para exibição de eventos.

```
7 class EventsController {
8   index = async (req, res) => {
9     try {
10      const query = req.query;
11      let filter = {};
12
13      if (Object.keys(query).length) {
14        for (const key in query) {
15          if (query.hasOwnProperty(key)) {
16            filter[key] = query[key];
17          }
18        }
19      }
20
21      const events = await Event.find(filter).populate({
22        path: 'department',
23        populate: { path: 'sector' },
24      });
25
26      return res.status(200).json({ events });
27    } catch (error) {
28      console.error(error);
29
30      return res.status(error.code || 500).json({
31        message: error.message || errors.INTERNAL_SERVER_ERROR,
32      });
33    }
34  };
35 }
```

Fonte: Autoria própria (2021).

5. CONSIDERAÇÕES FINAIS

Este trabalho consistiu no desenvolvimento de uma API para uma plataforma para centralização de eventos acadêmicos e seu código está disponível no GitHub¹.

O comportamento de todos os métodos desenvolvidos na API foram testados e validados por inserção de valores por meio do *software* Insomnia, que permite o envio de requisições e a automatização de testes manuais de APIs (KONG, 2021). Além disso, os objetos JSON requeridos no corpo da requisição são validados, visando a presença de informações fundamentais na realização de funções da API.

Como perspectivas futuras, pode-se citar, além da divulgação da API para a comunidade de desenvolvedores, a inserção de alguns recursos sugeridos pelos estudantes no questionário aplicado, como: a geolocalização de eventos; busca e filtragem por tema/área do evento.

¹ O código da API desenvolvida está disponível no link: <https://bit.ly/32DkDsq>.

REFERÊNCIAS

FIELDING, Roy et al. Hypertext transfer protocol–HTTP/1.1. 1999.

FOUNDATION, Node. Express. 2021. Disponível em: <https://expressjs.com/pt-br/>. Acesso em: abril de 2021.

JONES, Michael; CAMPBELL, Brian; MORTIMORE, Chuck. JSON Web Token (JWT) profile for OAuth 2.0 client authentication and authorization Grants. May-2015. {Online}. Disponível em: <https://tools.ietf.org/html/rfc7523>, 2015. Acesso em: Abril de 2021.

KONG, Inc. Build APIs that work. 2021. Disponível em: <https://insomnia.rest/>. Acesso em: abril de 2021.

MASSE, Mark. REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces. " O'Reilly Media, Inc.", 2011.

MARCHIORI, Patricia Zeni *et al.* Fatores motivacionais da comunidade científica para publicação e divulgação de sua produção em revistas científicas. Seminário Nacional de Bibliotecas Universitárias, v. 14, 2006.

MONGODB, Inc. MongoDB The application data platform. 2021. Disponível em: <https://www.mongodb.com/2>. Acesso em: Abril de 2021.

MORAIS, Sadrak Santiago. **MOSSORÓ+: PLATAFORMA PARA CENTRALIZAÇÃO DE EVENTOS EDUCACIONAIS**. 2020. Monografia (Graduação) - Universidade do Estado do Rio Grande do Norte, Mossoró-RN, 2020.

OPENJS, Foundation. About Node.js. 2021. Disponível em: <https://nodejs.org/en/about/>. Acesso em: Abril de 2021.



INPI INSTITUTO
NACIONAL
DA PROPRIEDADE
INDUSTRIAL
Assinado
Digitalmente

REPÚBLICA FEDERATIVA DO BRASIL
MINISTÉRIO DA ECONOMIA
INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL
DIRETORIA DE PATENTES, PROGRAMAS DE COMPUTADOR E TOPOGRAFIAS DE CIRCUITOS INTEGRADOS

Certificado de Registro de Programa de Computador

Processo Nº: **BR512020002636-2**

O Instituto Nacional da Propriedade Industrial expede o presente certificado de registro de programa de computador, válido por 50 anos a partir de 1º de janeiro subsequente à data de 10/11/2020, em conformidade com o §2º, art. 2º da Lei 9.609, de 19 de Fevereiro de 1998.

Título: MOSSORÓ+: Plataforma para centralização de eventos educacionais

Data de publicação: 10/11/2020

Data de criação: 10/11/2020

Titular(es): UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE - UERN

Autor(es): CICÍLIA RAQUEL MAIA LEITE; VICTOR MARCELO RODRIGUES SILVA; EXLLEY CLEMENTE DOS SANTOS; SADRAK SANTIAGO MORAIS; LUIZ CARLOS MOURA REGES

Linguagem: HTML; JAVA SCRIPT

Campo de aplicação: IF-07

Tipo de programa: GI-01

Algoritmo hash: SHA-512

Resumo digital hash:

DBFA33A46B0033FFAF06D03AB0C3195F8930BE6139998F6CFEEC2C6D65F9FBC17FA6E8EC352D170952872FB500F54A2E8E61CFD584FA4B6D103DFCF52C8AF728

Expedido em: 01/12/2020

Aprovado por:

Helmar Alvares

Chefe da DIPTO - Portaria/INPI/DIRPA Nº 09, de 01 de julho de 2019