



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**IZAQUIEL NUNES DO NASCIMENTO**

**CIDADE SINGULAR: UM APLICATIVO DE DIVULGAÇÃO DE  
ATIVIDADES, BENS E SERVIÇOS CULTURAIS**

**CAMPINA GRANDE - PB**

**2022**

**IZAQUIEL NUNES DO NASCIMENTO**

**CIDADE SINGULAR: UM APLICATIVO DE DIVULGAÇÃO DE  
ATIVIDADES, BENS E SERVIÇOS CULTURAIS**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de Bacharel em  
Ciência da Computação.**

**Orientador: Professor Dr. Marcelo Alves de Barros.**

**CAMPINA GRANDE - PB**

**2022**

**IZAQUIEL NUNES DO NASCIMENTO**

**CIDADE SINGULAR: UM APLICATIVO DE DIVULGAÇÃO DE  
ATIVIDADES, BENS E SERVIÇOS CULTURAIS**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de Bacharel em  
Ciência da Computação.**

**BANCA EXAMINADORA:**

**Professor Dr. Marcelo Alves de Barros  
Orientador – UASC/CEEI/UFCG**

**Professor Dr. Dalton Dario Serey Guerrero  
Examinador – UASC/CEEI/UFCG**

**Professor Tiago Lima Massoni  
Professor da Disciplina TCC – UASC/CEEI/UFCG**

**Trabalho aprovado em: 30 de março de 2022.**

**CAMPINA GRANDE - PB**

## **ABSTRACT**

One of the allies of sustainable development in cities is the Creative Economy, a way of uniting technology, innovation, culture, creativity, and sustainability to boost the economy. UNESCO created the Creative Cities Network precisely to encourage this type of initiative. Cities that join the Network commit to sharing their best practices and developing partnerships to boost the cultural sector. One of the responsibilities is the dissemination and dissemination of cultural activities, goods, and services. Therefore, this work proposes to build a system for the dissemination of cultural attractions in creative cities. The Cidade Singular app was finished with the planned features, such as the interactive map functionality to navigate between the main tourist spots. In addition to the application for visitors, a management application was developed, where specialized curators are added and they register and update the tourist attractions database.

# Cidade Singular: Um aplicativo de divulgação de atividades, bens e serviços culturais

Izaquiel Nunes do Nascimento

Orientador: Marcelo Alves de Barros  
Departamento de Sistemas e Computação  
Universidade Federal de Campina Grande  
Campina Grande, Paraíba, Brasil  
izaquiel.nascimento@ccc.ufcg.edu.br

## RESUMO

Um dos aliados do desenvolvimento sustentável nas cidades é a Economia Criativa, uma forma de unir tecnologia, inovação, cultura, criatividade e sustentabilidade para impulsionar a economia. A UNESCO criou a Rede de Cidades Criativas justamente para incentivar esse tipo de iniciativa. As cidades que aderem à Rede se comprometem a compartilhar suas melhores práticas e desenvolvem parcerias para impulsionar o setor cultural. Uma das responsabilidades é a divulgação e difusão de atividades, bens e serviços culturais. Diante disso, este trabalho se propõe a construir um sistema para divulgação de atrações culturais das cidades criativas. O aplicativo Cidade Singular foi finalizado com as funcionalidades planejadas, como a funcionalidade de mapa interativo para navegar entre os principais pontos turísticos. Além do aplicativo para os visitantes, foi desenvolvido o aplicativo de gestão onde são adicionados curadores especializados e os mesmos cadastram e atualizam a base de dados das atrações turísticas.

## Palavras-Chave

Cidade Criativa, Turismo, Divulgação.

## Repositórios

<https://github.com/CompCult/cidade-singular>

<https://github.com/CompCult/cidade-singular-admin>

<https://github.com/CompCult/cidade-singular-api>

## 1. INTRODUÇÃO

A Rede de Cidades Criativas (UNESCO Creative Cities Network - UCCN) foi criada em 2004 para promover a cooperação entre cidades que identificaram a criatividade como um fator estratégico para o desenvolvimento urbano sustentável. A Rede serve como uma plataforma internacional de intercâmbio e colaboração entre cidades para a implementação dos 17 Objetivos de Desenvolvimento Sustentável da Agenda 2030 das Nações Unidas, notadamente o Objetivo 11 sobre cidades e comunidades sustentáveis. A Rede é composta por 246 cidades de mais de 80 Estados-membros da UNESCO, em sete áreas criativas: artesanato e arte folclórica, design, cinema, gastronomia, literatura, arte midiática e música[1].

Ao aderir à Rede, as cidades se comprometem a compartilhar suas melhores práticas e desenvolver parcerias envolvendo os setores público e privado, bem como a sociedade civil, a fim de:

- fortalecer a criação, produção, distribuição e difusão de atividades, bens e serviços culturais;
- desenvolver pólos de criatividade e inovação e ampliar as oportunidades para criadores e profissionais do setor cultural;
- melhorar o acesso e a participação na vida cultural, em particular para grupos e indivíduos marginalizados ou vulneráveis;
- integrar plenamente a cultura e a criatividade nos planos de desenvolvimento sustentável.[2]

Um dos pontos chave para essa estratégia de desenvolvimento funcionar é a ampla divulgação das atrações culturais para atrair visitantes, tanto turistas quanto os próprios moradores da cidade que muitas vezes desconhecem a riqueza e diversidade cultural da sua própria cidade. Cada prefeitura possui seu meio de divulgação, seja por site próprio, redes sociais ou os demais canais de comunicação, porém pode ser interessante para os visitantes um aplicativo que agregasse todas as atrações culturais.

Perante o exposto, surge o aplicativo Cidade Singular, um guia específico de curadoria de uma Cidade Criativa da UNESCO mostrando cartograficamente a localização dos ativos culturais - aqui chamados de singularidades culturais - mais relevantes e abertos ao público, em especial os espaços de criação, produção e comercialização mais expressivos. Não se trata de um guia turístico tradicional genérico porque a informação georreferenciada por ele oferecida é fruto de uma curadoria especializada sobre os pontos de interesse do público interessado em uma das 7 áreas da Economia Criativa.

## 2. PROJETO DA SOLUÇÃO

Cidade Singular é um aplicativo mobile que busca ser um meio de divulgação das principais atividades, bens e serviços culturais de uma cidade. Na prática, se trata de um aplicativo de cartografia onde os diversos ativos culturais são mostrados como marcadores no mapa da cidade. A solução se divide em dois aplicativos: Gerenciador - aqui são adicionadas as singularidades, curadores e cidades e Cliente - aplicativo disponibilizado para o grande público, onde o mapa é exibido com as principais atrações culturais. Nas próximas seções serão discutidos os principais aspectos relacionados ao processo de desenvolvimento do aplicativo.

## 2.1 Principais funcionalidades

Nesta seção serão mostradas as principais funcionalidades desenvolvidas, tanto no aplicativo gerenciador quanto no aplicativo do visitante.

### 2.1.1 Visitante

Este é o aplicativo principal da solução, onde as pessoas interessadas nas atividades culturais da cidade poderão visualizar no mapa os principais pontos de visitação. As funcionalidades são:

- Tela inicial com as informações da cidade e possibilidade de trocar a cidade em questão
- Mapa da cidade com os pontos culturais. O usuário pode clicar em um ponto e será exibida uma janela com mais informações
- Tela de detalhes de uma singularidade com as informações de fotos, endereço, descrição e horário de visitação
- Tela com a identificação dos curadores da cidade em cada área da economia criativa

### 2.1.1 Gerenciador

No aplicativo gerenciador é necessário realizar login e registro, pois as funcionalidades requerem autenticação e autorização. Existem dois tipos de usuário que podem acessar o aplicativo: os pontos focais e os curadores.

Os pontos focais são responsáveis por gerenciar os curadores da cidade. As funcionalidades disponíveis para o ponto focal são:

- Listar todas as singularidades da cidade
- Remover uma singularidade
- Adicionar curadores para as sete áreas de atuação

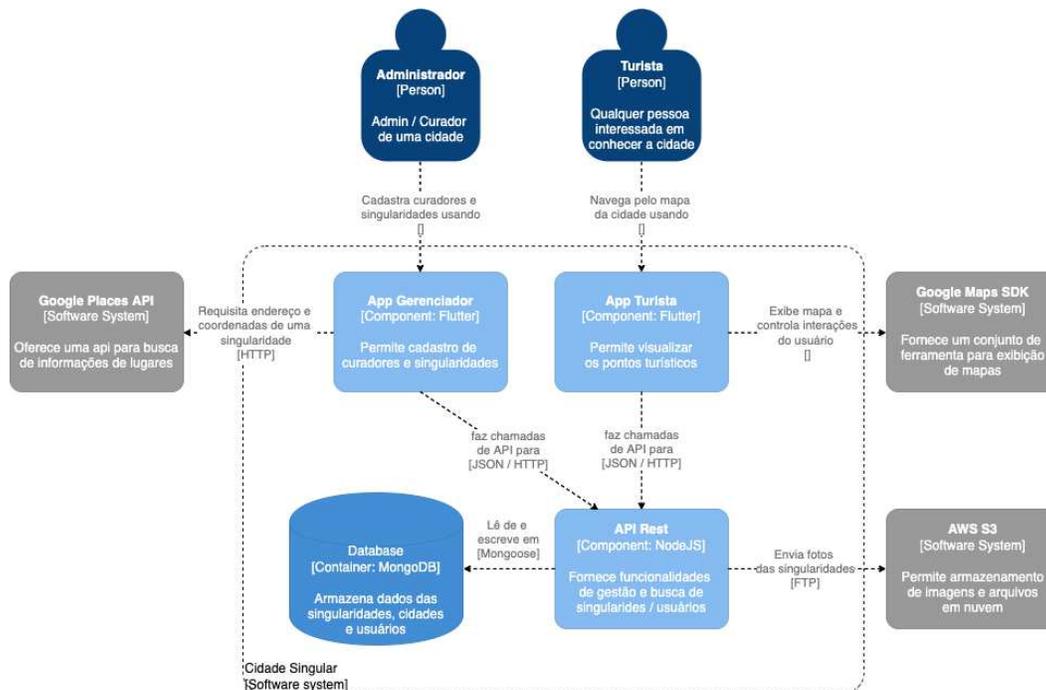
Os curadores são responsáveis por adicionar as singularidades no sistema. Logo, as funcionalidades disponíveis são:

- Adicionar uma singularidade com as informações necessárias como nome, descrição, fotos, horário de visitação e endereço
- Listar e excluir singularidades adicionadas

## 2.2 Arquitetura

Seguindo a divisão mais comum em desenvolvimento de sistemas web, o desenvolvimento do app Cidade Singular seguiu a arquitetura cliente-servidor. O servidor é uma API REST construída em NodeJS e o cliente é composto por dois frontends construídos em Flutter - Um aplicativo de administração para alimentação de dados das singularidades e um aplicativo do visitante para visualização dos ativos culturais da cidade. A figura abaixo mostra uma visão geral de como estes componentes estão dispostos e como interagem entre si. Nas próximas seções será discutido em detalhes cada componente do sistema.

Figura 1. Diagrama de Containers do sistema Cidade Singular



### 2.2.1 API REST

O backend da aplicação foi idealizado para seguir as restrições do estilo de arquitetura REST - sigla em inglês para "Representational State Transfer". Dentre as restrições que a interface de programação de aplicação (API) segue estão: todas as requisições são independentes, isto é, não guardam estado; os recursos são bem identificados e disponibilizados através de métodos HTTP como GET, POST, PUT e DELETE; e o formato dos dados transferidos é JSON (JavaScript Object Notation).

Seguindo o princípio de separação de responsabilidades com o intuito de manter o baixo acoplamento, dividiu-se o código com as seguintes camadas: Models, camada responsável pela modelagem e validação dos dados; Controllers, camada responsável por intermediar as requisições e delegar para os corretos services; Services, responsável por operar a lógica e regras de negócios; Middleware, responsável por interceptar as requisições e adicionar alguma verificação ou validação; e Routes, essa é a camada de comunicação, que provê todas as rotas que são disponibilizadas pela API.

A tecnologia utilizada foi NodeJS, um ambiente de execução JavaScript assíncrono orientado a eventos projetado para desenvolvimento de aplicações escaláveis de rede [4]. Em conjunto com o ambiente NodeJS, foi utilizado o framework Express que fornece uma miríade de métodos utilitários HTTP e middleware para rapidamente criar uma API robusta.

Para controlar o acesso das rotas, principalmente as rotas de administração, foi implementado um sistema de autenticação utilizando tokens de acesso com a biblioteca JSON Web Tokens (jwt). Ainda sobre segurança, algumas rotas só podem ser acessadas por usuários com um certo nível de autorização, como na rota de adicionar um curador à cidade que só pode ser operada por administradores. Essa funcionalidade foi implementada no middleware de autorização que faz essa verificação no token que é enviado nas requisições.

Outro ponto que vale a pena destacar aqui é sobre integração com serviços externos, o serviço de envios e-mail e o de armazenamento de arquivos. Nodemailer foi o módulo utilizado para realizar os envios de mensagens para o correio eletrônico dos usuários, inicialmente apenas com instruções para recuperação de senha. O outro serviço externo necessário para o funcionamento da aplicação foi a integração com o Storage S3 da Amazon para o armazenamento de fotos das singularidades, cidades e de perfil dos usuários.

Por fim, os dados foram persistidos utilizando o MongoDB, um banco de dados NoSQL (não relacional). Um módulo facilitador, o mongoose, foi utilizado para conexão, validação e modelagem dos dados.

### 2.2.2 Aplicativo do administrador / curador

A arquitetura seguiu um padrão de divisão em camadas semelhante ao da API, ambas sendo uma adaptação do padrão MVC (Model-View-Controller). Novamente, temos a camada Models responsável pelo formato e validação dos dados; a camada Services, responsável principalmente com a comunicação com os componentes externos como a API; e a camada Views, a mais representativa na estrutura de código do aplicativo. Sobre os controllers temos uma pequena modificação no padrão onde as views possuem algumas funções que assumem o papel dos controllers. Essas funções normalmente ficam em camadas

separadas das views e são chamadas de ViewModel, porém como esse projeto é de pequeno porte optou-se por separar em funções nas views.

Para construção do aplicativo de gestão foi escolhido o framework Flutter, um projeto open source mantido pela Google. Com o Flutter é possível criar aplicativos multiplataforma (Android, IOS e web) compilados nativamente a partir de uma única base de código. Esse framework foi escolhido por ser o mais popular e contar com uma gama de bibliotecas disponibilizadas pela comunidade.

As principais bibliotecas utilizadas foram Shared Preferences, Image Picker, Flutter Modular e Flutter Mobx. Shared Preferences foi utilizada para persistir o login do usuário do app, guardando o token de acesso na memória do celular. Image Picker foi utilizado para permitir que o administrador possa escolher imagens da galeria e colocar como as fotos das singularidades ou escolher uma foto de perfil. Flutter Modular forneceu uma poderosa estrutura de injeção de dependências como também um robusto sistema de rotas. Por último temos o Mobx que é uma biblioteca de gerenciamento de estado que simplifica a conexão dos dados reativos do aplicativo com a interface do usuário.

Sobre os principais comportamentos do sistema, como o aplicativo é acessado tanto por administradores quanto por curadores, foi implementado um sistema que exibe apenas as funcionalidades que o usuário possui autorização de acesso. Por exemplo, apenas os administradores têm acesso à página de gestão de curadores. Outro ponto importante é sobre o cadastro de uma singularidade. Para facilitar o cadastro dos dados, utilizou-se o Google Places API para permitir que o curador possa utilizar um campo de busca para encontrar o ponto turístico que deseja adicionar e assim cadastrar automaticamente as informações como título, endereço e principalmente as coordenadas geográficas da localidade.

### 2.2.2 Aplicativo do visitante

A arquitetura desse frontend seguiu semelhante à apresentada para o aplicativo administrador/curador, inclusive com as mesmas tecnologias e bibliotecas empregadas. A principal diferença é que o aplicativo do administrador tem o foco em cadastro dos dados, já o aplicativo do visitante foca na exibição.

Por se tratar de um projeto de cartografia, foi fundamental buscar uma forma de exibição e navegação dos mapas das cidades. Para tal funcionalidade foi utilizado a biblioteca google\_maps\_flutter, disponibilizada pelo próprio time de desenvolvimento do Flutter. Com as funcionalidades providas pela biblioteca foi implementado um mapa interativo da cidade com a possibilidade de navegar pelo mapa e interagir com os pontos turísticos.

## 3. SISTEMA EM USO

Ao final de três meses de desenvolvimento, foi finalizada a primeira versão do aplicativo Cidade Singular. Os dados mostrados a seguir são das duas cidades piloto para exibição do projeto: Campina Grande e João Pessoa.

As figuras 2 e 3 mostram o resultado final das páginas de adicionar singularidades e curadores no aplicativo administrador. Além dessas páginas, foram finalizadas as páginas de listagem das singularidades e dos curadores, como também a página de perfil com a opção de editar os dados do usuário.

Figura 2. Página de adicionar singularidade (App administrador)

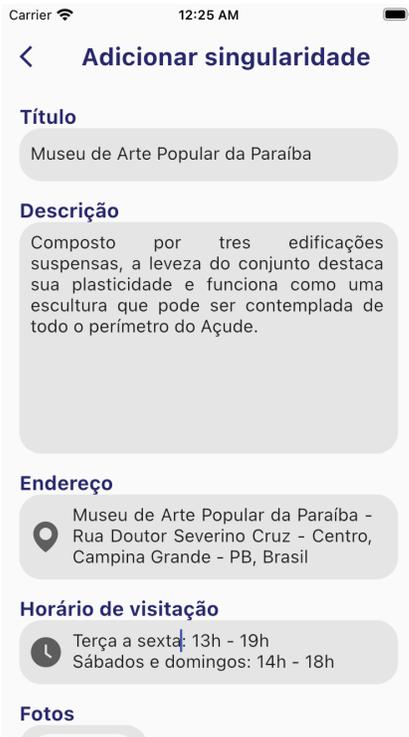


Figura 4. Página inicial do App do visitante com pop-up para trocar de cidade



Figura 2. Página com opções de adicionar um dos sete tipos de curador (App administrador)



Figura 5. Página com mapa de Campina Grande (App visitante)

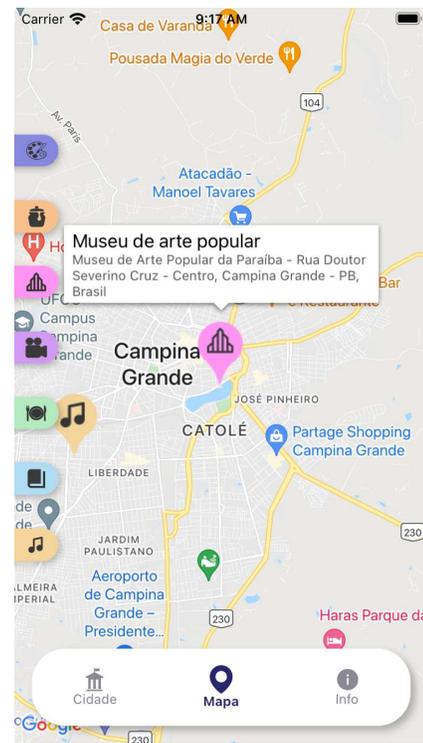


Figura 6. Página com listagem dos curadores (App visitante)

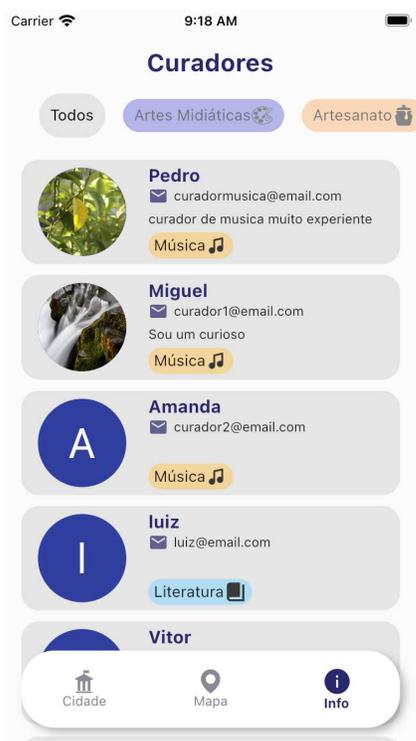
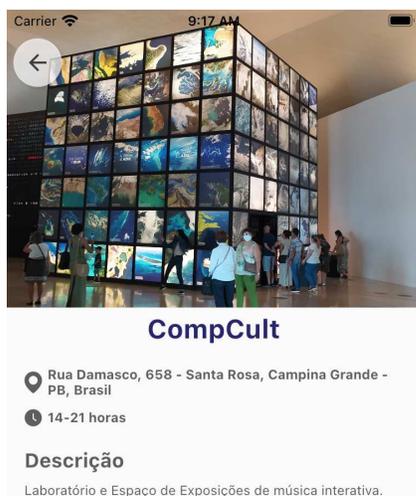


Figura 7. Página com detalhe de uma singularidade (App visitante)



No Aplicativo do visitante foram finalizadas a página de detalhes da cidade com opção de trocar a cidade (figura 4), a página de mapa interativo (figura 5) com opção de filtros por categoria e opção de clicar em uma singularidade e ser direcionado para a página de detalhes (figura 7), e a página de informações sobre os curadores da cidade (figura 6).

## 4. EXPERIÊNCIA E LIÇÕES APRENDIDAS

### 4.1 Processo de desenvolvimento

Adotando uma metodologia de desenvolvimento ágil, foram realizadas pequenas entregas ao longo do processo. O orientador desse projeto atuou como Product Owner (PO) dando feedbacks a cada entrega. Com base nos comentários e sugestões chegamos nos resultados exibidos no tópico anterior.

Logo após levantar os requisitos essenciais, o desenvolvedor atuou também como designer fazendo todo o trabalho de prototipagem das telas no software Figma. Apesar de não ser uma responsabilidade do programador fazer o design de aplicações, essa atividade contribuiu para um melhor entendimento das funcionalidades que iriam ser implementadas e também ajudou no planejamento das próximas etapas de desenvolvimento.

A API Rest foi a primeira parte do sistema a ser construída, pois é a especialidade do programador. Dito isso, rapidamente o backend da aplicação foi desenvolvido sem maiores problemas. O maior esforço ficou, então, com os front-ends do Cidade Singular. O aplicativo do administrador foi construído em seguida e foi o que demandou mais tempo, pois envolvia mostrar elementos na tela de acordo com nível de autorização do usuário e também comunicação com a api do google para buscar informações do endereço das singularidades. Já o aplicativo do curador, apesar de parecer mais complexo, muito do código e das páginas se assemelhavam com o do aplicativo do administrador, logo o processo de desenvolvimento foi mais tranquilo.

### 4.2 Principais desafios e limitações

Como o objetivo do sistema é atrair atenção para as singularidades da cidade, a interface do aplicativo deveria ser bonita e moderna. Esse foi o primeiro desafio encontrado: fazer o design do Cidade Singular. O desenvolvedor não possui conhecimentos sólidos na área de interface de usuário e por isso essa foi uma demanda que exigiu bastante tempo e pesquisa por referências.

Outros pontos que exigiram bastante atenção foram as integrações com os serviços da Google. Esse foi o primeiro contato do desenvolvedor com a API Places e também com SDK de mapas.

### 4.3 Trabalhos futuros

Apesar das funcionalidades principais do sistema terem sido finalizadas, algumas melhorias podem ser adicionadas para tornar o aplicativo mais popular, dentre elas:

- Opção do visitante poder avaliar as singularidades;
- Opção de adicionar comentários nas singularidades;
- Além da visualização no mapa, ter a opção de visualização das singularidades em uma lista;
- Página de divulgação de eventos;
- Opção de compartilhamento nas principais redes sociais;

As sugestões acima são relacionadas ao aspecto social, criando uma comunidade ao redor do aplicativo. Esse é um ponto muito importante para a adoção por parte da sociedade da Cidade Singular, pois assim gera mais confiança e engajamento.

## 5. REFERÊNCIAS

- [1] Patrimônio e criatividade. Disponível em: <<https://pt.unesco.org/covid19/cultureresponse>>
- [2] What is the Creative Cities Network ? Disponível em: <<https://en.unesco.org/creative-cities/content/about-us>>
- [3] O que é API REST? Disponível em: <<https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>>
- [4] Sobre Node.js. Disponível em: <<https://nodejs.org/pt-br/about/>>

---

### Sobre o autor:

Izaquiel Nunes do Nascimento é graduando em Ciência da Computação na Universidade Federal de Campina Grande com foco em engenharia de software.