



FACULDADE QI BRASIL

## PROJETO DE SISTEMA PARA INTEGRAÇÃO DA API JIRA COM WEB SERVICE

Gdo. Rafael Marques Vargas

Esp. Poliane Daniele Evangelista Rego

Me. Silvio Cesar Viegas

### RESUMO

Neste projeto vamos desenvolver Web Service para integrar API Jira com objetivo de exibir informações de ISSUES ao cliente.

Estas informações no Jira são internas, para desenvolvedores e pessoal de suporte. Utilizaremos como linguagem de programação a linguagem JavaScript e Node.js. Para montar este WebService vamos construir o servidor com Express utilizando Axios para as requisições assíncronas.

Para exibir as informações vamos utilizar Nodemon que auxiliará no autoreload da aplicação.

Esta ideia surgiu pois não disponibilizamos usuário Jira ao cliente, desta forma, com esta integração selecionamos quais informações compartilhar com cliente mantidas no Jira.

Palavras Chaves: Jira, JavaScript, Web Service, API, ISSUES

### 1 INTRODUÇÃO

O objetivo deste trabalho é o desenvolvimento de Web Service onde vamos consumir API do Jira a fim de recuperar informações de tarefas. As tarefas que vamos recuperar tratam-se de tarefas oriundas de projeto da empresa a qual trabalho, desta forma temos o objetivo de apresentar informações das tarefas do projeto ao cliente sem a necessidade de criar usuário ao cliente.

O Jira é um software comercial que permite o gerenciamento de tarefas e organização da demanda a ser realizada em períodos conforme definido em reunião, ou metodologia ágil. Para cada tarefa é possível mensurar tempo para realização através de Storys Points prevendo o tempo que esta tarefa será concluída.

### 2 TEMA

Quando falamos em API Jira, estamos tratando de uma ferramenta que integra os dados do Jira ao Web Service, através desta API vamos buscar as informações da plataforma do Jira e utilizar no Web Service.

TecMundo descreve API como:

API é o acrônimo de Application Programming Interface ou, em português, Interface de Programação de Aplicativos.

Esta interface é o conjunto de padrões de programação que permite a

construção de aplicativos e a sua utilização de maneira não tão evidente para os usuários.

Sendo assim, uma vez que o software do Jira é utilizado pela equipe interna da empresa, as informações tratadas por esta plataforma são privadas. Desta forma, o acesso é por meio de usuário e senha, como não é compartilhado usuário de acesso ao cliente, este projeto tem o intuito de exibir informações selecionadas das tarefas as quais competem ao cliente subordinado conhecer e acompanhar suas demandas.

## 2.1 DELIMITAÇÃO DO TEMA

Esta primeira versão do projeto tem escopo unicamente para exibir tarefas criadas pelo time de suporte no Jira a fim de demonstrar andamento ao cliente selecionado.

## 2.2 PROBLEMA

O Jira é utilizado como software para gerenciamento de tarefas das quais clientes abrem tarefas internas com equipe de suporte, respectivamente, após análise da equipe de suporte é aberto ISSUE, no Jira, para atendimento por equipe de desenvolvimento. Desta forma, o acompanhamento destas tarefas/ISSUES fica exclusivamente repassado através do time de suporte/cliente. De que forma será possível agilizar este repasse de informações ao cliente de forma rotineira?

## 2.3 JUSTIFICATIVA

Através deste projeto será possível reduzir tempo em repasse da informação ao cliente, uma vez que esta informação estará de forma organizada e clara em tempo real disponível ao cliente para consultar.

## 3 OBJETIVO GERAL

Desenvolver solução tecnológica para exibir informações das tarefas no Jira ao cliente de acordo com suas necessidades.

### 3.1 OBJETIVOS ESPECÍFICOS

Web Service é definido como um software na internet que utiliza sistema de mensagens comunicando com outros aplicativos, neste caso API do Jira. Através desta comunicação vamos recuperar informações de tarefas no Jira.

Os objetivos deste software são divididos da seguinte forma:

- Receber as informações da plataforma do Jira das tarefas por clientes;
- Disponibilizar de forma organizada e permitida as informações através de Web Service.
- Consultar e exibir estas informações pela plataforma cliente que fará conexão com este Web Service.

#### **4 FUNDAMENTAÇÃO TEÓRICA**

Será desenvolvido um Web Service para atender segmento de software Jira. O Jira, é o software utilizado pela empresa para gerenciamento de tarefas pelo time de desenvolvimento. Através desta plataforma é possível gerenciar o esforço desempenhado em cada atividade através dos Story Points empregados em cada tarefa.

Agile School (2022), define Story Point da seguinte forma, “O Story Point é unidade de complexidade de esforço que a gente tem para construir um pedaço de software, por exemplo uma User Story ou qualquer tipo de demanda que você tenha para construir.”.

Esta forma de trabalhar dar-se-á por metodologia ágil, o Scrum. A cada 15 dias, time de suporte trás as tarefas priorizadas pelo cliente para reunião com time de desenvolvimento. Nesta reunião é definido quais tarefas serão atendidas nesta Sprint. Coordenador dos programadores, estabelece limite de Story Point para atendimento na Sprint definido em conjunto aos desenvolvedores a quantidade de esforço que cada tarefa exigirá para seu desenvolvimento. Sendo assim, ao final desta reunião temos a Sprint elaborada com as tarefas para atendimento conforme priorizado com cliente e conforme capacidade do time de desenvolvedores.

Fábio (2018, p. 68), pontua Sprint como evento de duração fixa dentro do Scrum.

Sprint é uma iteração e um evento com duração fixa. Pelas regras do Scrum, devem durar um mês ou menos e possuir uma meta estabelecida com um objetivo claro.

É possível considerar que as Sprint são pequenos projetos com duração de no máximo um mês. Como qualquer projeto, toda Sprint deve servir para realizar algo.

A metodologia Ágil empregada, Scrum, significa uma forma de organizar os processos e pessoas de forma efetiva para o desenvolvimento de aplicações, Fábio (2018, p. 53) explica claramente a seguir:

O Scrum não é um processo, técnica ou método definitivo, e sim um framework dentro do qual podem ser empregados diversos processos ou técnicas, sendo complementado com ferramentas e abordagens para melhor atender às necessidades de produtos, times, ambientes e empresas.

## **5 METODOLOGIA**

Método é a forma que devem ser aplicados diferentes processos para certa finalidade, meios os quais levam ao resultado desejado. Como explica; Amado (2007, p. 27) "Método é a ordem que se deve impor aos diferentes processos necessários para atingir um certo fim ou um resultado desejado. Nas ciências, entende-se como método o conjunto de processos empregados na investigação e na demonstração da verdade."

O método adotado é o indutivo, neste analisamos um caso específico, baseado na experiência da necessidade do cliente por informações.

## **6 PESQUISA APLICADA**

A pesquisa aplicada neste trabalho é a pesquisa qualitativa. Baseado na necessidade do cliente em visualizar as informações necessárias para entendimento e andamento das demandas. Conforme Denzin e Lincoln (2006), a pesquisa qualitativa envolve uma abordagem interpretativa do mundo.

A procura por estas informações é carente por todos os clientes uma vez que são informações importantes para acompanhamento da entrega da tarefa, se haverá desenvolvimento ou se cabe explicação e orientação a respeito do problema apresentado.

## **7 DESENVOLVIMENTO**

Nesta seção, vamos apresentar as técnicas para o desenvolvimento do projeto. Elucidar as tecnologias utilizadas bem como sintetizar os recursos utilizados para compreendermos a abordagem deste desenvolvimento.

Em primeiro momento, será elaborado projeto onde vamos desenvolver uma aplicação Web Service com finalidade de recuperar informações de tarefas classificadas como Erros na plataforma Jira. Estas tarefas são divididas, no Jira, em Tarefas, Erros e Histórias.

Tarefas compreendem a pequenas partes do trabalho a ser realizado, são distintas. Histórias pertencem ao grupo de tarefas direcionadas ao desenvolvimento de novas funcionalidades das quais o projeto atual não foi projetado. Erros, nosso

ponto focal, destaca-se por funcionalidades que deveriam cumprir sua função para qual foram desenvolvidas, porém por algum motivo não apresenta o resultado da forma esperada ou descrita no documento de requisitos.

## 7.1 JAVASCRIPT

JavaScript é uma linguagem Web, tem característica de linguagem com tipagem dinâmica onde a memória é alocada somente quando define valores para esta variável. Esta abordagem também permite alterar o valor desta variável após utilização apresentando-se então a característica dinâmica e fraca, pois o tipo de valor armazenado, seja número ou texto pode ser alterado sem apresentar erro.

É executada ao lado do cliente, de forma geral no navegador do usuário. Veja, o glossário de desenvolvimento da Mozilla apresenta JavaScript a seguir: “JavaScript (ou "JS") é uma linguagem de programação utilizada principalmente para scripts dinâmicos do lado do cliente em páginas web, podendo também ser utilizada no lado do servidor, usando um interpretador (em inglês: runtime) como o Node.js.”.

## 7.2 NODE.JS

Node.js é um Software que permitir executar os códigos desenvolvidos em JavaScript de forma assíncrona. Esta forma de rodar o código significa processar as tarefas sem bloquear o processamento destas, caso aja alguma outra tarefa em execução ao mesmo tempo. No site Resultados Digitais (09/06/2022) explica: “Comunicação assíncrona é toda comunicação que não ocorre em tempo real. Nela, o interlocutor pode ler ou ouvir a mensagem na hora que desejar e só então respondê-la”, ou seja, o programa executa sem necessidade de aguardar uma tarefa finalizar para prosseguir.

## 7.3 EXPRESS

Express é um framework web estruturada, um conjunto de códigos pronto para o desenvolvimento Web rodando sob o ambiente Node.js. (MDN WEB DOCS).

Com framework Express vamos gerenciar as requisições via HTTP com a URL do domínio do projeto no Jira. Definimos as configurações de conexão da aplicação Web para montar nosso servidor Web, o Web Service.

## 7.4 AXIOS

Através do Axios vamos consumir os dados da API do Jira e exibir na nossa página em JavaScript. No site VUE.JS apresenta Axios da seguinte forma: “Há diversos momentos quando você está desenvolvendo uma aplicação Web que podem necessitar consumir e exibir dados de uma API. Há várias maneiras de se fazer isso, mas a maneira mais popular é usando Axios, um cliente HTTP baseado em Promises.”.

## 7.5 NODEMON

Ferramenta utilizada para apoiar durante o desenvolvimento do projeto, terá a função de reiniciar automaticamente o aplicativo em cada alteração no arquivo. Desta forma ganhamos agilidade ao codar pois não há necessidade de parar para digitar comandos para executar a aplicação uma vez que está em constante auto reload. Remy sintetiza a seguir: “nodemon é uma ferramenta que ajuda a desenvolver aplicativos baseados em Node.js reiniciando automaticamente o aplicativo de nó quando são detectadas alterações de arquivo no diretório.”.

## 8 CRIANDO WEB SERVICE

Nesta seção, vamos apresentar de que forma desenvolvemos nossa API. Durante desenvolvimento observamos que é necessário autenticação na API do Jira bem como autenticação no projeto ao qual queremos retornar as informações desejadas.

Desta forma, adotamos um projeto exemplo de minha autoria, onde neste projeto criei algumas tarefas para retornar no Web Service.

Iniciamos o projeto com a instalação das dependências descritas no desenvolvimento. Importamos as dependências através dos comandos abaixo:

Figura 1: importação de dependências

```
const express = require('express')
const port = 3000
const axios = require('axios')
const cors = require('cors')
const app = express()
const route = express.Router()
```

Fonte: O autor, 2022.

Para iniciar a implementação utilizamos padrão apresentado na página da Express, Axios de acordo com as imagens abaixo:

Figura 2: Padrão para importar e utilizar Express.

```
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

Fonte: <https://expressjs.com/pt-br/starter/hello-world.html>, 2022

Figura 3: Padrão para importar e utilizar Axios

```
import axios from 'axios';

// Envia uma requisição post
axios({
  method: "post",
  url: "/user/12345",
  data: {
    firstName: "Fred",
    lastName: "Flintstone",
  },
});
```

Fonte: [https://axios-http.com/ptbr/docs/api\\_intro](https://axios-http.com/ptbr/docs/api_intro), 2022.

Agora vamos iniciar criando uma função assíncrona utilizando o express. Router onde tem a função de auxiliar na manipulação na rota da API. Para executar esta função assíncrona utilizamos o await passando a biblioteca Axios que é uma biblioteca de consulta assíncrona para APIs. Neste momento pegamos o cabeçalho, nossos headers, e passo a autenticação que API exige. Repasso URL do Jira do meu

projeto. Logo após envio somente as informações necessárias do arquivo JSON que é retornado nesta consulta. Veja abaixo imagem com este código:

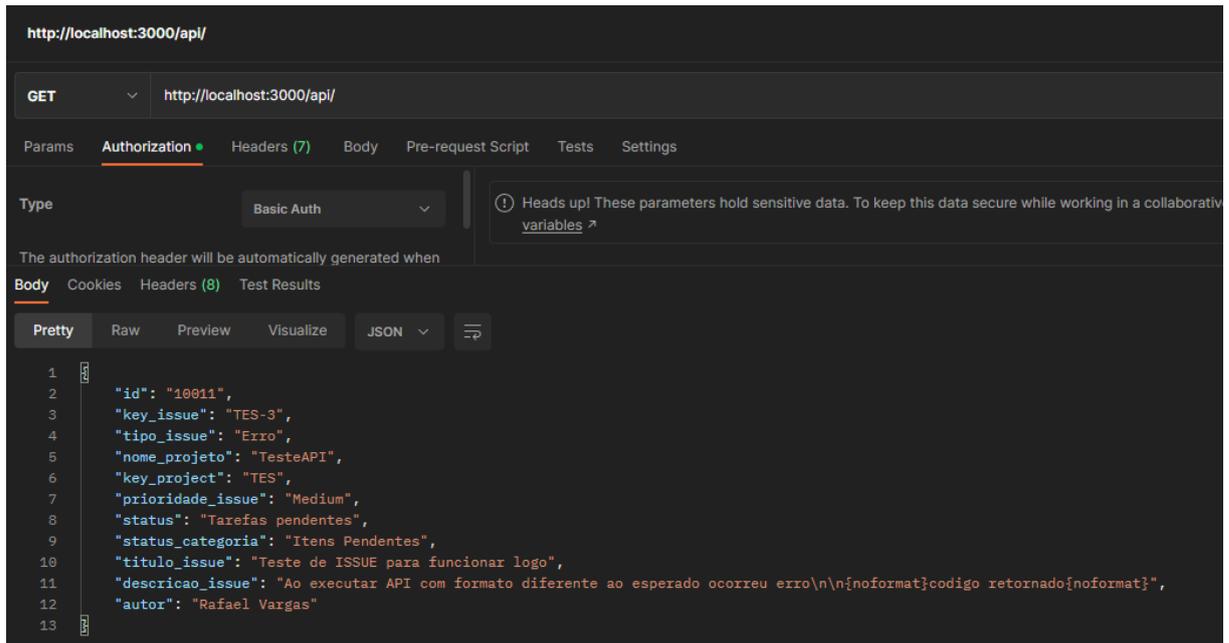
Figura 4: Código do projeto, autenticação e organização dos dados da ISSUE.

```
route.get('/', async (req, res) => {  
  const response = await axios({  
    method: "get",  
    headers: {  
      req: id='TES-3',  
      authorization: req.headers.authorization  
    },  
    url: `https://estudosjs.atlassian.net/rest/api/2/issue/${id}`,  
  })  
  
  res.send({  
    id: response.data.id,  
    key_issue: response.data.key,  
    tipo_issue: response.data.fields.issuetype.name,  
    nome_projeto: response.data.fields.project.name,  
    key_project: response.data.fields.project.key,  
    prioridade_issue: response.data.fields.priority.name,  
    status: response.data.fields.status.name,  
    status_categoria: response.data.fields.status.statusCategory.name,  
    titulo_issue: response.data.fields.summary,  
    descricao_issue: response.data.fields.description,  
    autor: response.data.fields.creator.displayName  
  })  
})
```

Fonte: O autor, 2022.

Como defini aplicação para rodar na porta 3000, agora no Postman entro com a requisição `http://localhost:3000/api`. Configuro autenticação na aba Authorization e Basic Auth e cliço em enviar:

Figura 5: Utilizando Postman como cliente para testar projeto.



Fonte: O autor, 2022.

Repare que neste momento temos o retorno de todas as informações que configuramos para receber do Jira.

A função do Postman neste exemplo foi para simular a requisição do cliente ao nosso Web Service.

## 9 RESULTADO

O resultado do projeto não atendeu a todas as expectativas, uma vez que para recuperar as informações de ISSUES do Projeto inicial era preciso autenticação e permissão ao acessar projeto fora da instituição. Uma vez que esta autenticação não foi liberada para acesso, desenvolvemos uma solução em cima de um Projeto exemplo onde recuperamos apenas uma Tarefa.

Esta tarefa foi recuperada através de desenvolvimento de Web Service que consiste na consulta à API Jira de determinada ISSUE ou tarefa. Ao recuperar estas informações, organizamos os dados coletando somente o necessário para exibir ao cliente.

Durante o desenvolvimento deste Web Service, separamos algumas responsabilidades para evolução desta API Web Service, que são listadas abaixo:

- Criação de ISSUES através de consulta ao Web Service enviado informações

relevantes.

- Alteração de ISSUE da mesma forma apresentado acima.
- Autenticação através do próprio Web Service.
- Possibilidade de integração com Mobile.

## REFERÊNCIAS

TEC MUNDO. **O que é API?** Disponível em:

<<https://www.tecmundo.com.br/programacao/1807-o-que-e-api-.htm#:~:text=API%20%C3%A9%20o%20acr%C3%B4nimo%20de,t%C3%A3o%20evidente%20para%20os%20usu%C3%A1rios>>. Acesso em: 3 set. 2022.

AGILE SCHOOL. **Story Points: Entenda de uma vez por todas como fazer estimativas.** Disponível em: <<https://agileschool.com.br/story-points-entenda-de-uma-vez-por-todas-como-fazer-estimativas/>>. Acesso em: 3 set. 2022.

CRUZ, Fábio. **Scrum e Agile em Projetos - 2ª Edição:** Conquiste sua certificação e aprenda a usar métodos ágeis no seu dia a dia. São Paulo: Brasport Livros e Multimídia Ltda, 2018.

CERVO, Amado L; BERVIAN, Pedro A; DA SILVA, Roberto. **Metodologia Científica.** 6º ed. São Paulo: Pearson, 2007.

RESULTADOS DIGITAIS. **Comunicação assíncrona: o que é, boas práticas e ferramentas.** Disponível em:

<<https://resultadosdigitais.com.br/marketing/comunicacao-assincrona/#:~:text=Comunica%C3%A7%C3%A3o%20ass%C3%ADncrona%20%C3%A9%20aquela%20que,Um%20exemplo%20%C3%A9%20o%20email.>>. Acesso em: 18 set. 2022.

MDN. **JavaScript.** Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Glossary/JavaScript>>. Acesso em: 18 set. 2022.

MDN WEB DOCS. **Express Web Framework (Node.js/JavaScript).** Disponível em: <[https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express\\_Nodejs](https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs)>. Acesso em: 18 set. 2022.

VUE.JS. **Usando Axios para consumir APIs.** Disponível em: <<https://br.vuejs.org/v2/cookbook/using-axios-to-consume-apis.html>>. Acesso em: 18 set. 2022.

SHARP, REMY. **Nodemon.** Disponível em: <<https://www.npmjs.com/package/nodemon>>. Acesso em: 18 set. 2022.

SHARP, REMY. **Nodemon.** Disponível em: <<https://www.npmjs.com/package/nodemon>>. Acesso em: 18 set. 2022.