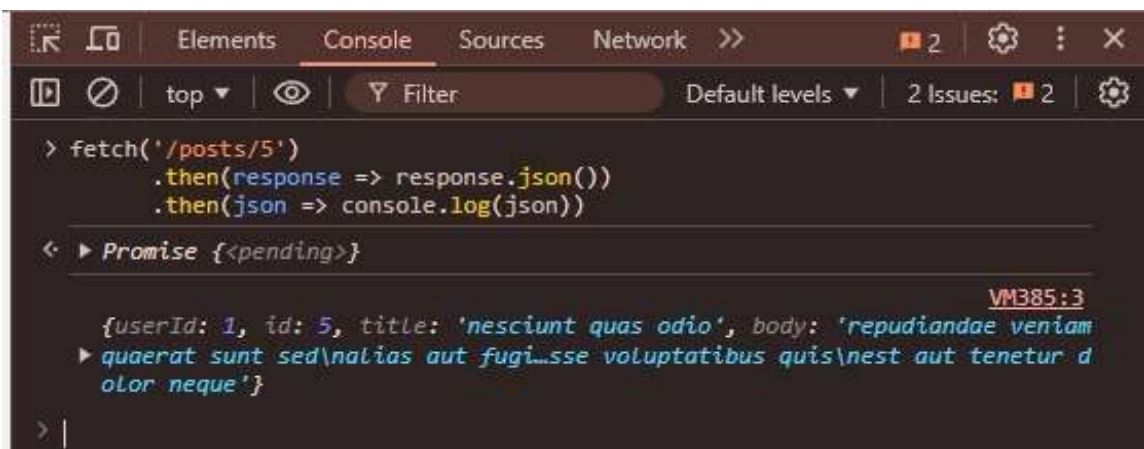


A5- Aplicando Conhecimento

COMPONENTE CURRICULAR:	PROGRAMAÇÃO DE SISTEMAS II
NOME DO ALUNO:	JOÃO PEDRO LIMA LUSTOSA AMORIM
RA:	10289920

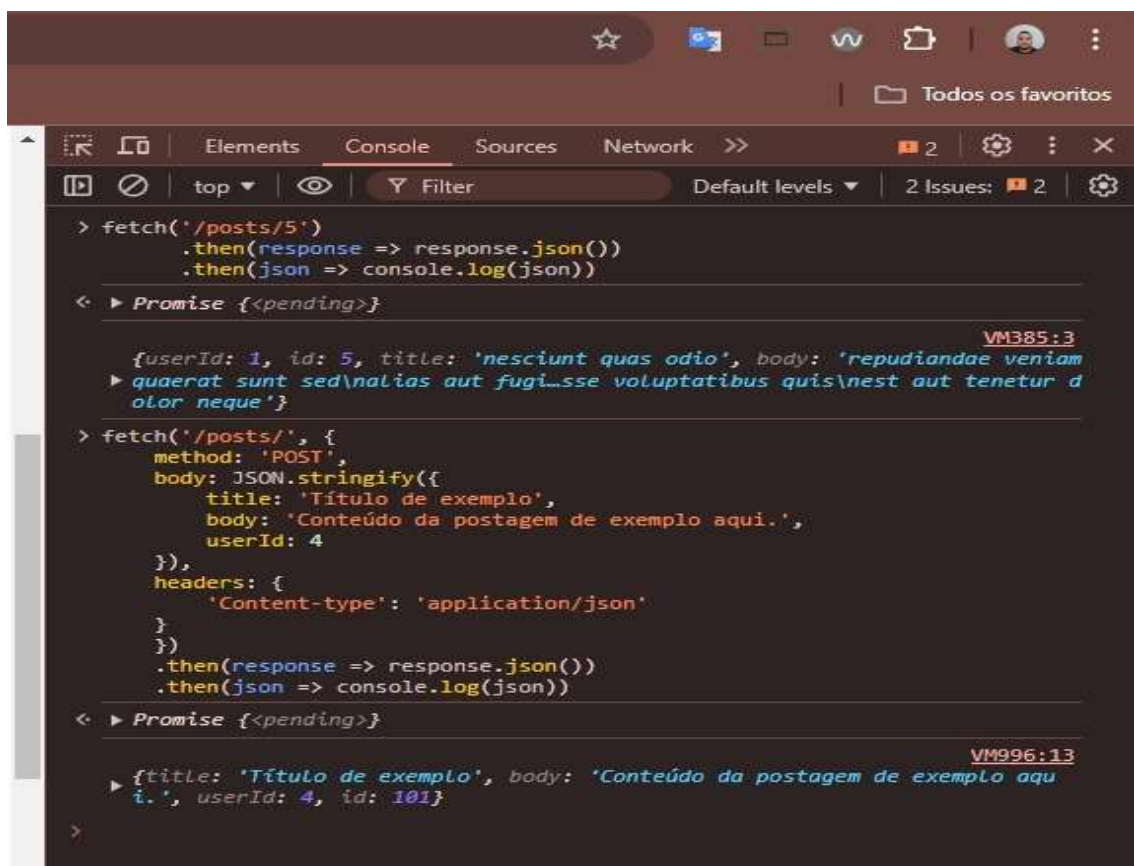


```

> fetch('/posts/5')
  .then(response => response.json())
  .then(json => console.log(json))

< ▶ Promise {<pending>}

VM385:3
{userId: 1, id: 5, title: 'nesciunt quas odio', body: 'repudiandae veniam
▶ quærat sunt sed\aliam aut fugi...sse voluptatibus quis\nest aut tenetur d
olor neque'}
  
```



```

> fetch('/posts/5')
  .then(response => response.json())
  .then(json => console.log(json))

< ▶ Promise {<pending>}

VM385:3
{userId: 1, id: 5, title: 'nesciunt quas odio', body: 'repudiandae veniam
▶ quærat sunt sed\aliam aut fugi...sse voluptatibus quis\nest aut tenetur d
olor neque'}

> fetch('/posts/', {
  method: 'POST',
  body: JSON.stringify({
    title: 'Título de exemplo',
    body: 'Conteúdo da postagem de exemplo aqui.',
    userId: 4
  }),
  headers: {
    'Content-type': 'application/json'
  }
})
  .then(response => response.json())
  .then(json => console.log(json))

< ▶ Promise {<pending>}

VM996:13
{title: 'Título de exemplo', body: 'Conteúdo da postagem de exemplo aqu
i.', userId: 4, id: 101}
  
```



The screenshot shows a web browser's developer console with the following content:

```
> fetch('/posts/5')
  .then(response => response.json())
  .then(json => console.log(json))

< ▶ Promise {<pending>}

VM385:3
{userId: 1, id: 5, title: 'nesciunt quas odio', body: 'repudiandae veniam
▶ quaerat sunt sed\alias aut fugi...sse voluptatibus quis\nest aut tenetur d
olor neque'}
```

```
> fetch('/posts/', {
  method: 'POST',
  body: JSON.stringify({
    title: 'Título de exemplo',
    body: 'Conteúdo da postagem de exemplo aqui.',
    userId: 4
  }),
  headers: {
    'Content-type': 'application/json'
  }
})
  .then(response => response.json())
  .then(json => console.log(json))

< ▶ Promise {<pending>}

VM996:13
▶ {title: 'Título de exemplo', body: 'Conteúdo da postagem de exemplo aqu
i.', userId: 4, id: 101}
```

```
> fetch('/posts/5', {
  method: 'PUT',
  body: JSON.stringify({
    id: 5,
    title: 'Título modificado',
    body: 'Conteúdo também modificado.',
    userId: 2
  }),
  headers: {
    'Content-type': 'application/json'
  }
})
  .then(response => response.json())
  .then(json => console.log(json))

< ▶ Promise {<pending>}

VM1043:14
▶ {id: 5, title: 'Título modificado', body: 'Conteúdo também modificado.',
userId: 2}
```

```

> fetch('/posts/5')
  .then(response => response.json())
  .then(json => console.log(json))

< ▶ Promise {<pending>}

VM385:3
  {userId: 1, id: 5, title: 'nesciunt quas odio', body: 'repudiandae veniam
  ▶ queraat sunt sed\nalias aut fugi...sse voluptatibus quis\nest aut tenetur d
  olor neque'}

> fetch('/posts/', {
  method: 'POST',
  body: JSON.stringify({
    title: 'Título de exemplo',
    body: 'Conteúdo da postagem de exemplo aqui.',
    userId: 4
  }),
  headers: {
    'Content-type': 'application/json'
  }
})
  .then(response => response.json())
  .then(json => console.log(json))

< ▶ Promise {<pending>}

VM996:13
  ▶ {title: 'Título de exemplo', body: 'Conteúdo da postagem de exemplo aqu
  i.', userId: 4, id: 101}

> fetch('/posts/5', {
  method: 'PUT',
  body: JSON.stringify({
    id: 5,
    title: 'Título modificado',
    body: 'Conteúdo também modificado.',
    userId: 2
  }),
  headers: {
    'Content-type': 'application/json'
  }
})
  .then(response => response.json())
  .then(json => console.log(json))

< ▶ Promise {<pending>}

VM1043:14
  ▶ {id: 5, title: 'Título modificado', body: 'Conteúdo também modificado.',
  userId: 2}

> fetch('/posts/5', {
  method: 'DELETE'
})
  .then(response => console.log(response))

< ▶ Promise {<pending>}

VM1353:4
  ▶ Response {type: 'basic', url: 'https://jsonplaceholder.typicode.com/post
  s/5', redirected: false, status: 200, ok: true, ...}

```

```

> fetch('https://jsonplaceholder.typicode.com/users/4')
  .then(data => {
    console.log('Dados do usuário:', data);
  })
< ▶ Promise {<pending>}

Dados do usuário: VM1396:3
Response {type: 'basic', url: 'https://jsonplaceholder.typicode.com/users/4', redirected: false, status: 200, ok: true, ...}
  body: (...)
  bodyUsed: false
  headers: Headers
    ▶ [[Prototype]]: Headers
    ok: true
    redirected: false
    status: 200
    statusText: ""
    type: "basic"
    url: "https://jsonplaceholder.typicode.com/users/4"
  [[Prototype]]: Response
    ▶ arrayBuffer: f arrayBuffer()
    ▶ blob: f blob()
    body: (...)
    bodyUsed: (...)
    ▶ clone: f clone()
    ▶ formData: f formData()
    headers: (...)
    ▶ json: f json()
    ok: (...)
    redirected: (...)
    status: (...)
    statusText: (...)
    ▶ text: f text()
    type: (...)
    url: (...)
    ▶ constructor: f Response()
    ▶ error: f error()
    ▶ json: f json()
    ▶ redirect: f redirect()
      arguments: null
      caller: null
      length: 0
      name: "Response"
    ▶ prototype: Response {...}
    ▶ [[Prototype]]: f ()
    ▶ [[Scopes]]: Scopes[0]
    Symbol(Symbol.toStringTag): "Response"
    ▶ get body: f body()
    ▶ get bodyUsed: f bodyUsed()
    ▶ get headers: f headers()
    ▶ get ok: f ok()
    ▶ get redirected: f redirected()
    ▶ get status: f status()
    ▶ get statusText: f statusText()
    ▶ get type: f type()
  
```