

ATIVIDADES

Exercício 1

Baseado na especificação do início deste documento para o IFgram, modele endpoints, preenchendo a tabela com a modelagem da sua API, bem como a especificação do esquema (Schema) em JSON. Para validar o seu JSON, use algum site de validação, tal como <https://jsonformatter.curiousconcept.com/> e <https://jsonlint.com/>.

Template

Especificação do esquema em JSON

```
{
  "Usuario": { "type":
    "object",
    "properties": {
      "id": {
        "type": "integer"
      },
      "nome": {
        "type": "string"
      },
      "email": { "type":
        "string",
        "format": "email"
      },
    },
  },
}
```

```

    "senha": { "type":
      "string",
      "minLength": 6
    }
  },
  "required": ["nome", "email", "senha"]
}
}

```

Especificação do esquema em JSON

Verbo HTTP	Endpoints (path)	Descrição	Input	Outputs	Cód. Sucesso	Cód. Falha
POST	/usuarios	Criar novo usuário	{ "nome ": "Mario", , "email ": "mario@gmail.com", , "senha": "32654" }	{ "id": 1, "nome": "Mario", "email": "mario@gmail.com" }	201 Created	400 Bad Request
GET	/usuarios	Listar todos os usuários	-	[{ "id": 1, "nome": "Mario", "email": "mario@gmail.com"}]	200 OK	500 Internal Server Error

GET	/usuarios/{id}	Buscar usuário por ID	–	{ "id": 1, "nome": "Mario", "email": "mario@gmail.com" }	200 OK	404 Not Found
PUT	/usuarios/{id}	Atualizar dados de um usuário	{ "nome ": "Roberto", "email ": "roberto@gmail.com" }	{ "id": 1, "nome": "Roberto", "email": "roberto@gmail.com" }	200 OK	400 Bad Request / 404 Not Found

Exercício 2

Implemente o que acabou de especificar em um projeto Java com Spring Web MVC, seguindo o que foi apresentado em sala de aula.