A aplicação foi desenvolvida utilizando **Ruby 3.0.0** e Framework **Rails: 6.1.3**

No caso optei por usar a extensão do Visual Studio Code chamada **Remote-Containers** para a criação do meu ambiente de desenvolvimento.

**Passos para a criação da aplicação**

Dentro do ambiente com o Ruby e o Rails instalado foram executados os passos abaixo que após validados foram adicionados ao script “**create\_api.sh**”:

**#criação do projeto com o parâmetro –api para não utilizar os recursos html:**

rails new quero\_api --api

**#copia dos arquivos docker file e script entrypoint para uso futuro em ci/cd:**

cp -p Dockerfile teste\_api/ #copia o docker file para uso futuro em ci/cd

cp -p entrypoint.sh teste\_api/ #copia o script entrypoint.sh para uso do docker file

# **executa o scaffold para gerar nosso model e controller da nossa api**

cd quero\_api/

rails g scaffold Sample data:string

**#transforma a classe disponível em um banco de dados relacional**

rails db:migrate #transforma a classe disponivel e transforma em um banco de dados

**#executa o comando no console do rails para criar a chave/valor no nosso banco**

rails runner "Sample.create(data: 'hello world')"

**#baixa a gem que facilita modificar a resposta do json**

bundle add active\_model\_serializers

**#gera o arquivo da nossa classe para alterarmos o retorno do método GET**

rails generate serializer sample

**#substutui o retorno da chave id pela chave data para o retorno do Json**

cd app/serializers/

sed -i 's/id/data/g' sample\_serializer.rb

**#inicia a aplicação na porta 3000**

rails server

**Como criar e testar a aplicação:**

Em um ambiente com Ruby e Rails instalado, executar o script ssh: **./create\_api.sh**

O script irá criar o projeto quero\_api.

Após aguardar a criação, acessar ou enviar o método get para: **http://localhost:3000/samples**

**ps:** no repositório deixei uma api funcional chamada teste\_api

**Caso queira gerar a imagem da aplicação:**

Ir na pasta da aplicação e executar o comando: docker build -t **quero\_api** .