



UNIVERSIDADE  
FEDERAL RURAL  
DE PERNAMBUCO



## Projeto Interdisciplinar para Sistemas de Informação IV

### Projeto:

Especificação: Versão 1.3 de 28 de abril de 2021

Número de integrantes de cada equipe: 1

Data de lançamento: 03 de maio de 2021

### 1. Objetivo

O objetivo deste projeto individual é o desenvolvimento de um sistema distribuído em nuvem, utilizando alguns dos serviços disponíveis no AWS Educate, que sejam capazes de integrar dados meteorológicos de estações do Instituto Nacional de Meteorologia (INMET) existentes no estado de Pernambuco.

Cada uma das estações fornece dados de variáveis meteorológicas, listadas na tabela abaixo. As variáveis meteorológicas horárias serão capturadas através de um [serviço web do próprio INMET implementado em uma API REST](#). Uma tarefa programada utilizando **AWS Lambda** e **EasyCron.com** deverá realizar a chamada ao serviço web do INMET, de hora em hora, filtrar os dados e repassá-los para o serviço **AWS Kinesis Data Streams**. Em seguida, estes dados deverão ser processados pelo **Kinesis Data Analytics**. Os dados processados serão enviados ao **Kinesis Data Firehose** para persistência em um *bucket S3*. Os dados persistidos servirão como referência para o processamento no Kinesis Data Analytics e para a API REST de consulta aos dados que será desenvolvida em outra função **AWS Lambda**.

O processamento consistirá na integração dos dados horários de **pelo menos cinco** das variáveis meteorológicas disponíveis, **de todas as estações meteorológicas de Pernambuco**. Pode-se registrar, por exemplo, o total de precipitação em Pernambuco em um determinado dia ou qual estação apresentou a menor temperatura em todo o estado.

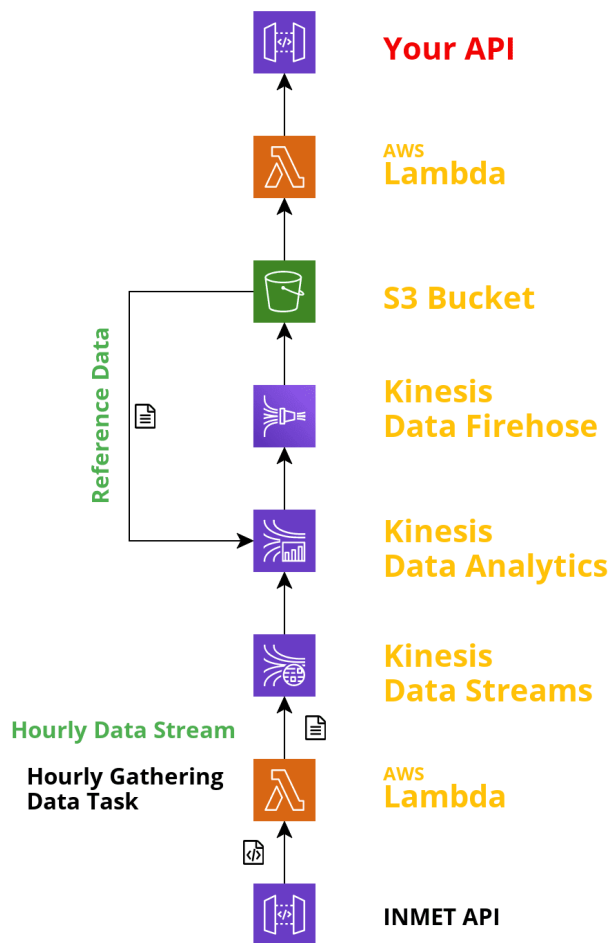


Figura 1 - Arquitetura geral dos serviços em nuvem

A informação integrada deverá considerar a janela de um dia, no entanto, para máximos e mínimos, deverá ser informado nos resultados o horário em que este valor foi observado. Por esta razão, **deverá ser acessada a API do INMET que fornece o dado horário e não a que já apresenta os dados diários consolidados**. Perceba que para computar os valores diários de uma variável meteorológica deverá ser armazenado ao longo do dia o valor respectivo até aquele instante e atualizá-lo quando necessário. Desta forma, a consulta, que veremos a seguir, informará os dados mais recentes.

Variável meteorológica	Função de agregação	Endpoint	Retorno
PRECIPITAÇÃO TOTAL (mm)	acumulado	/precipitacao	1

RADIAÇÃO GLOBAL (W/m²)	acumulado	/radiacao	1
PRESSÃO ATMOSFÉRICA MÁXIMA (mB)	máximo	/pressmax	2
PRESSÃO ATMOSFÉRICA INSTANTÂNEA (mB)	média	/pressmed	1
PRESSÃO ATMOSFÉRICA MÍNIMA (mB)	mínimo	/pressmin	2
TEMPERATURA MÁXIMA (°C)	máximo	/tempmax	2
TEMPERATURA INSTANTÂNEA (°C)	média	/tempmed	1
TEMPERATURA MÍNIMA (°C)	mínimo	/tempmin	2
PONTO DE ORVALHO MÁXIMO (°C)	máximo	/ptomax	2
PONTO DE ORVALHO INSTANTÂNEA (°C)	média	/ptomed	1
PONTO DE ORVALHO MÍNIMO (°C)	mínimo	/ptomin	2
UMIDADE RELATIVA DO AR MÁXIMA (%)	máximo	/umidmax	2
UMIDADE RELATIVA DO AR INSTANTÂNEA (%)	média	/umidmed	1
UMIDADE RELATIVA DO AR MÍNIMA (%)	mínimo	/umidmin	2
VELOCIDADE DO VENTO INSTANTÂNEA (m/s)	média	/velmed	1
VELOCIDADE DA RAJADA MÁXIMA DE VENTO (m/s)	máximo	/velmax	2

Estes dados integrados diários devem estar disponíveis para serem acessados através de uma API REST utilizando apenas métodos GET. Cada uma das cinco variáveis deverá estar associada a um *endpoint* específico, como mostrado na tabela. O retorno da API será um JSON que é diferente para cada tipo de agregação e deve seguir o esquema mostrado abaixo:

- Retorno TIPO 1 - acumulados e médias:  

```
{ 'VALOR_OBSERVADO': <valor> }
```
- Retorno TIPO 2 - máximos e mínimos:  

```
{ 'CODIGO_ESTACAO': <codigo>, 'NOME_ESTACAO': <nome>,  
  'LATITUDE': <latitude>, 'LONGITUDE': <longitude>,  
  'HORARIO_COLETA': <horario>, 'VALOR_OBSERVADO': <valor> }
```

A escolha das variáveis meteorológicas a serem processadas é uma escolha sua. Qualquer combinação é possível desde que se obedeça ao **mínimo de uso de cinco variáveis meteorológicas e haja pelo menos uma variável para cada tipo de função de agregação.**

## 2. Material de suporte:

A seguir disponibilizamos alguns materiais que podem ajudar e serem utilizados como referência para o desenvolvimento do projeto.

- [Tutoriais práticos da Amazon Web Services](#)
- [ThingsBoard Self-hosted setup using AWS EC2 instance](#)
- [AWS Kinesis Data Analytics Example](#)
- [AWS Cloud Best Practices](#)
- [Well Architected Framework](#)
- [Exemplo de Arquitetura na AWS - Transportation and Logistics Data Lake](#)
- [Exemplo de Arquitetura na AWS - Connected Restaurants using IoT and AI/ML](#)

## 3. O que deverá ser entregue:

- Um vídeo de no máximo 5 minutos relatando a experiência de desenvolvimento do projeto. O vídeo **deverá** conter:

- uma apresentação geral da solução desenvolvida;
- um relato sobre os maiores desafios enfrentados durante o desenvolvimento;
- vantagens e desvantagens da utilização de cada um dos serviços e possíveis alternativas (por exemplo: substituir o Kinesis Data Analytics por um cluster EMR rodando Spark).
- Um link para um repositório com todos os códigos desenvolvidos. e.g. GitHub, GitLab, etc...

#### 4. Dicas

- Não deixe para começar o projeto mais tarde. Comece logo! É impossível fazer o projeto de “virada” ... ele vai lhe tomar alguns dias de trabalho. ;-)
- O projeto é individual mas não é proibido trocar ideia com os colegas e professores na plataforma AVA. O problema enfrentado por um pode ser o mesmo de outros. Portanto, compartilhem as dores. =)
- **Os professores acompanharão o desenvolvimento dos projetos, mas a responsabilidade é sua.** Assim, não acumule dúvidas e utilize o canal de discussão da disciplina. Não espere que os professores “corram atrás” de você.
- Lembre-se da importância do projeto! Certamente a especificação apresentada deixa pontos em aberto. Dúvidas? Pergunte!!!