NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

# Arquitectura Para Integração de Sistemas

Aula Pratica 1

# Preparação da Componente Prática

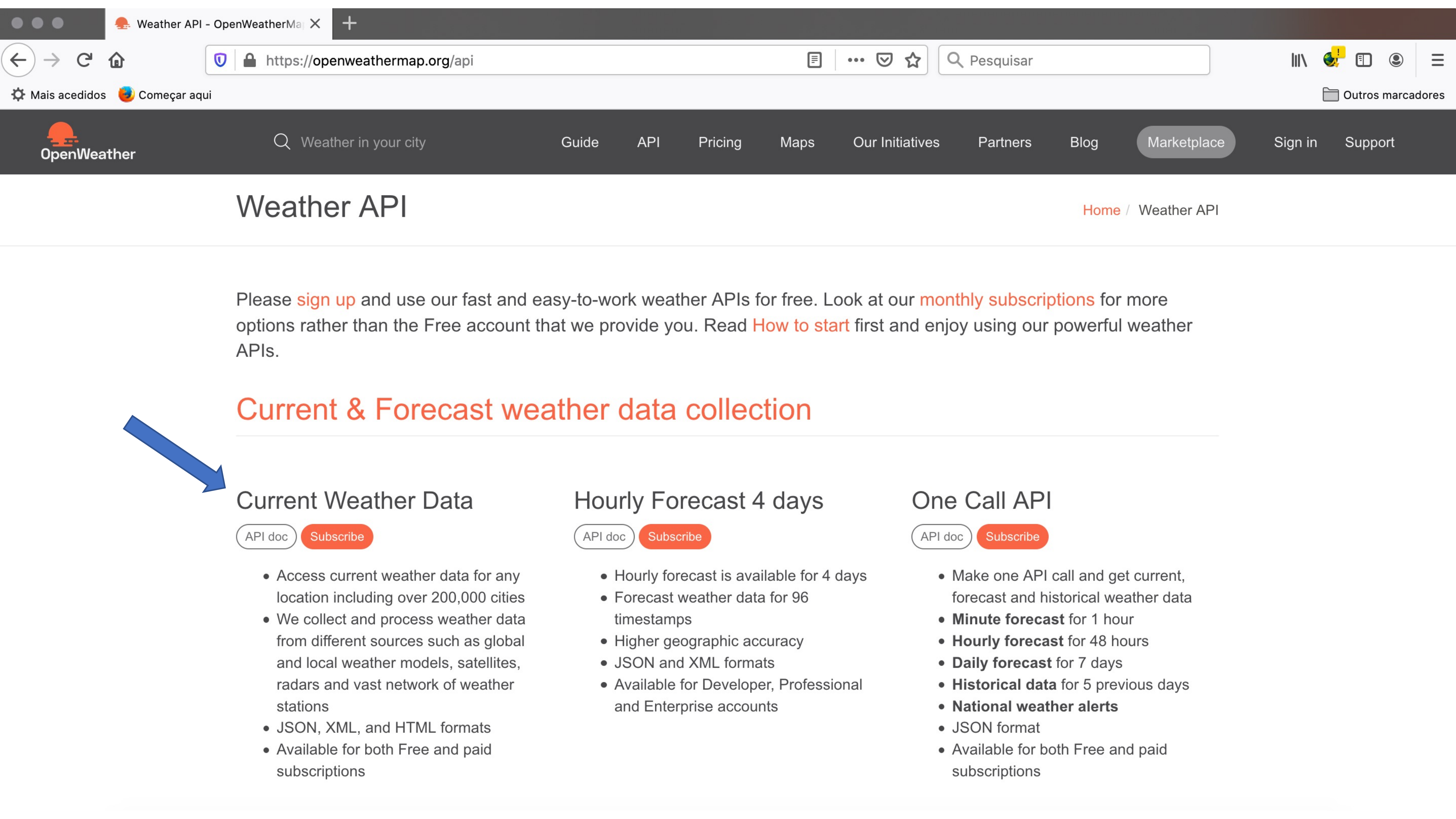
## Trazer de casa o seguinte software pré-instalado

1. Java SE (https://www.oracle.com/pt/java/technologies/javase-downloads.html )
2. Apache Tomcat server (https://tomcat.apache.org/download-90.cgi), de preferência instalem a versão '32-bit/64-bit Windows Service Installer'
3. Para realizar este trabalho é necessário instalar o ACTIVITI BPMN Engine (http://activiti.org/download.html) no vosso computador. O Activiti é uma ferramenta *open-source* de modelação de *workflows* que pode executar processos de negócios descritos na linguagem BPMN 2.0. Para fazerem a instalação sigam as instruções deste site https://www.activiti.org/userguide/, nomeadamente as instruções do capitulo 2.1 "One minute version".
4. Postman (https://www.postman.com/downloads/ ) e instalar
5. Façam o download do "Eclipse IDE for Java EE Developers" usando este link: https://www.eclipse.org/downloads/packages/release/2020-12/r/eclipse-ide-java-developers. Criem um workspace, de preferência na pasta do projeto

# Exercicios Práticos

- Acedam ao Moodle e descarreguem o ficheiro “trabalho_1.pdf”
- No final da aula, é necessária a submissão Individual no Moodle de um documento com as respostas ao exercícios

- Seguir a partir de:
  - #Lab 1: Invocação de API externa. Exemplo: OpenWeatherMap



# Weather API

Please sign up and use our fast and easy-to-work weather APIs for free. Look at our monthly subscriptions for more options rather than the Free account that we provide you. Read How to start first and enjoy using our powerful weather APIs.

## Current & Forecast weather data collection

### Current Weather Data

API doc Subscribe

- Access current weather data for any location including over 200,000 cities
- We collect and process weather data from different sources such as global and local weather models, satellites, radars and vast network of weather stations
- JSON, XML, and HTML formats
- Available for both Free and paid subscriptions

### Hourly Forecast 4 days

API doc Subscribe

- Hourly forecast is available for 4 days
- Forecast weather data for 96 timestamps
- Higher geographic accuracy
- JSON and XML formats
- Available for Developer, Professional and Enterprise accounts

### One Call API

API doc Subscribe

- Make one API call and get current, forecast and historical weather data
- **Minute forecast** for 1 hour
- **Hourly forecast** for 48 hours
- **Daily forecast** for 7 days
- **Historical data** for 5 previous days
- **National weather alerts**
- JSON format
- Available for both Free and paid subscriptions

# Informação de Base

**J**ava**S**cript **O**bject **N**otation

JSON can be used to represent the resources and to send them in http requests or responses.

A JSON object is just a string of characters, in the end, it's just text.

- Although it has JavaScript in its name, this is:
  - just a way to format and represent data,
  - so it can be used in most programming languages

# JSON – JavaScript Object Notation

Let's take the **products** example, imagine that the product models, has a **name**, **price** and the number of **units** in stock. A simple representation can be:

```
{
  "id": 10,
  "title": "Nice Sneakers 2.0",
  "units": 14
}
```

# JSON – JavaScript Object Notation

A JSON object **starts and ends with curly brackets {}**.

The content is a **list of key/value pairs**.

The **keys must have double quotes ""** around them, these are just strings.

# JSON – JavaScript Object Notation

**Possible values**:

- a **string**
- a **number**
- an **object** (JSON object)
- an **array [...]**
- a **boolean (true / false)**
- ***null (different from "undefined")***

https://www.w3schools.com/js/js_json_datatypes.asp

# JSON – JavaScript Object Notation

Example with a **JSON object** as a value:

```
{
  "id": 10,
  "title": "Nice Sneakers 2.0",
  "units": 14,
  "price": {"value": "59.99", "currency": "EUR"}
}
```

# JSON – JavaScript Object Notation

Example with an **Array** as a value:

```
{
  "id": 10,
  "title": "Nice Sneakers 2.0",
  "units": 14,
  "colors": ["red", "green", "blue"]
}
```

# JSON – JavaScript Object Notation

Example with a **boolean (true/false)** and **null**:

```
{
  "id": 10,
  "title": "Nice Sneakers 2.0",
  "units": 14,
  "returnable": true,
  "long_description": null
}
```