

MeteoTower

Diseño y desarrollo de una aplicación para el monitoreo del clima en torres de telecomunicaciones

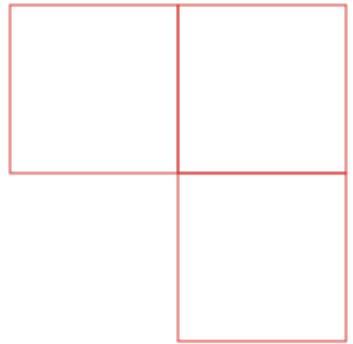
**Santiago Almazán Alarcón
Antonio Pérez Laguna**

22 de junio de 2023



Indice

■ Descripción del proyecto	01
■ Necesidad y objetivos	02
■ Meteorología y Prevención de Riesgos	03
■ Desarrollo del proyecto	04
■ Conclusiones	05
■ Futuras líneas de trabajo	06
■ Backup	07



01

Descripción del proyecto



Propuesta de proyecto

Los requerimientos generales para este proyecto son:

- ✓ Esta aplicación utilizará una API **gratuita** de clima para obtener información en **tiempo real** sobre las condiciones climáticas en la zona donde se encuentran las torres de telecomunicaciones (torre existente en Madrid).
- ✓ La aplicación mostrará una vista en tiempo real de los datos meteorológicos, como la **temperatura**, la velocidad del **viento** y la **precipitación**.
- ✓ Es necesario definir la API de clima utilizar, con la aplicación programada en **Java**, no es necesario utilizar interfaz gráfica, pero sí llamadas periódicas a funciones.
- ✓ La aplicación finalmente permitirá a los técnicos de mantenimiento definir **umbrales de alerta** para ciertas condiciones climáticas, como ráfagas de viento fuertes o lluvias torrenciales, y recibirá notificaciones cuando se superen estos umbrales.
- ✓ Además, la aplicación generará un **historial** de las condiciones climáticas y las alertas recibidas, para que los técnicos puedan realizar análisis y mejorar la planificación de mantenimiento. Con esta aplicación, los técnicos de mantenimiento podrán monitorear el clima de manera constante y tomar medidas preventivas en caso de condiciones climáticas adversas.

Necesidad de la aplicación

- **Objetivo** - ofrecer una aplicación capaz de mejorar la calidad y la eficacia del trabajo mediante la monitorización de diferentes variables que afectan al clima.
 - Aumentando la eficiencia de operación de la empresa.
 - Garantizando la seguridad de los trabajadores.
- **Retos** - aspectos a considerar:
 - Elegir una API de clima.
 - Aprender a programar en Java.
 - Pensar como un usuario de la aplicación.
 - Aprender lo básico de Meteorología.
 - Identificar los casos de uso.
 - Revisar legislación laboral en vigor.
 - Analizar los datos meteorológicos de la API.
 - Programar la aplicación.
 - Cumplir de plazos de entrega.
- **Gestión y desarrollo de proyectos** – Gantt, Trello, GitHub, Eclipse IDE.



02

Meteorología y Prevención de Riesgos



Fenómenos meteorológicos adversos

Todo evento meteorológico capaz de producir, directa o indirectamente, daños a las personas y los bienes y que, por tanto, puede producir impactos.

También puede considerarse como tal cualquier fenómeno meteorológico susceptible de alterar la actividad humana de forma significativa en un ámbito espacial determinado.

La peligrosidad de un FMA indica la magnitud esperable de los impactos que podría producir, sin tener en cuenta la exposición y la vulnerabilidad de las personas y los bienes afectados por el fenómeno.



Avisos y alertas

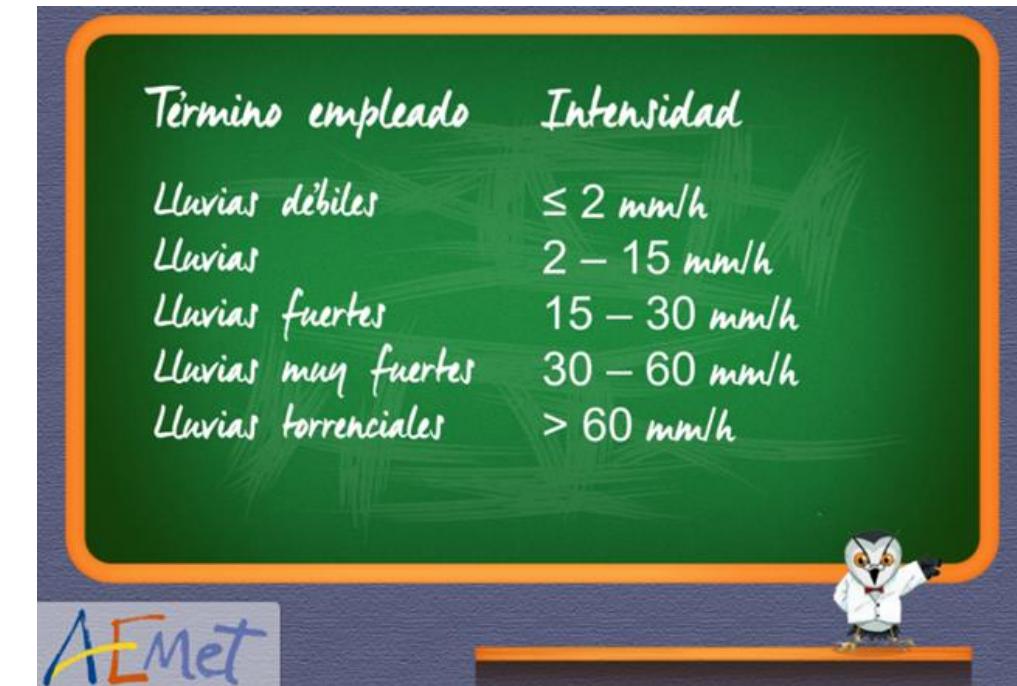
- **Nivel verde** - No existe ningún riesgo meteorológico.
- **Nivel amarillo** - El peligro es bajo.
Recomendación: ESTÉ ATENTO.
- **Nivel naranja** - El peligro es importante.
Recomendación: ESTÉ PREPARADO.
- **Nivel rojo** - El peligro es extraordinario.
Recomendación: Tome medidas preventivas y ACTÚE según las indicaciones de las autoridades.



Activación de avisos

Los fenómenos meteorológicos adversos objeto de este proyecto son:

- Lluvia torrencial, cuando la precipitación acumulada supera los 60 mm/h.
- Viento fuerte, cuando la velocidad máxima del viento supera los 50 km/h.
- Temperatura máxima excesiva, cuando supera los 40 ° C.



Número de Beaufort	Velocidad del viento (km/h)	Nudos (millas náuticas/h)	Denominación	Aspecto del mar	Efectos en tierra
7	50 a 61	28 a 33	Frescachón (Viento fuerte)	Mar gruesa, con espuma arrastrada en dirección del viento	Se mueven los árboles grandes, dificultad para caminar contra el viento

3.12 COMUNIDAD AUTÓNOMA DE MADRID

CÓDIGO	NOMBRE DE LA ZONA	PROVINCIA	umbrales		temp. máximas		temp. mínimas		racha máxima		precipitación 12 h		precipitación 1 h		nieve 24 h		
			amilo	nanja	rojo	amilo	nanja	rojo	amilo	nanja	rojo	amilo	nanja	rojo	amilo	nanja	rojo
722801	Sierra de Madrid	Madrid	34	37	40	-6	-10	-14	80	100	140	40	80	120	15	30	60
722802	Metropolitana y Henares	Madrid	36	39	42	-4	-8	-12	70	90	130	40	80	120	15	30	60
722803	Sur, Vegas y Oeste	Madrid	36	39	42	-4	-8	-12	70	90	130	40	80	120	15	30	60

Prevención de riesgos laborales



- **Ley 31/1995 de Prevención de Riesgos Laborales**, la empresa está obligada a informar de la existencia de riesgo grave e inminente y a adoptar las medidas necesarias para que las personas afectadas se protejan, interrumpan la actividad e incluso, si fuera necesario, para que abandonen de inmediato el lugar de trabajo.
- **RD 1215/1997** - formación de trabajo en altura incluirá medidas de seguridad ante condiciones meteorológicas que puedan afectar a la seguridad.
- **RD 614/2001** - establece que los trabajos con riesgo eléctrico se suspenderán en caso de tormenta, lluvia o viento fuertes, nevadas o cualquier otra condición desfavorable.
- **RD 836/2003** - obliga a tener un aviso intermitente cuando la velocidad del viento alcance los 50 km/h y uno continuo a los 70 km/h.
- **VI Convenio General del Sector de la Construcción** - prevé la posibilidad de proponer horarios distintos que permitan evitar las horas de mayor insolación. En caso de no interrumpirse la actividad, se tomarán medidas que reduzcan estos riesgos.

03

Desarrollo de la aplicación





Diagrama de Gantt

HELGA CRISTINA QUIRÓS SANDOVAL





Trello Espacios de trabajo Reciente Marcado Plantillas Crear

PROYECTO INGENIERIA Tablero Power-Up Calendario Seguimiento Get started with TeamGantt Power-Ups Automatización Filtrar Compartir ...

Ideas

Lista de tareas

En proceso

Hecho

Pruebas de la aplicación

Cuadro de mando

Repositorio GitHub MeteoTower

Borrador de anteproyecto

Casos de uso y marco teórico

Trabajos en altura. Planificación de los trabajos.

Borrador Programa para llamar API AEMET.txt

JSON

JAVA

Importante Entrega de Memoria completa

Entregas 2/4

+ Añada una tarjeta

Importante Defensa del proyecto, con presentación

22 de jun. 0/3

Importante Entrega de Memoria completa

23 de jun. 0/3

Santiago Prioridad media Antonio Creación de un interfaz grafico

15 de jun. 1/4

Prioridad media Iconos Estado del cielo

1/2

+ Añada una tarjeta

METEO TOWER

Seleccionar logo de la aplicación

4

Video MeteoTower

3 22 de jun. 1

Selección de ubicación de la torre

1

Antonio Índice de Figuras

23 de jun.

Antonio Índice de Tablas

+ Añada una tarjeta

Buscar Buscar

JC HS SA L

Con vencimiento esta semana en este tablero

Tarjetas con plazo vencido en este tablero

+ Añada una tarjeta





Lino09 / Proyecto-de-Ingenieria

Type to search

Code Issues Pull requests Actions Projects Security Insights

Proyecto-de-Ingenieria Public

Watch 1 Fork 0 Star 0

master 1 branch 0 tags

Go to file Add file <> Code About

No description, website, or topics provided.

Activity 0 stars 1 watching 0 forks Report repository

Releases No releases published Create a new release

Packages No packages published Publish your first package

Contributors 2

Lino09 AntonioPerezUEM

Lin09 PI - Proyecto Meteo Tower completo borrador v3 20230614 6747056 5 days ago 45 commits

src	borrado linea	2 months ago
1.0	Update 1.0	last month
1.1	Create 1.1	last month
1.2(montaña)	Create 1.2(montaña)	last month
2	Update 2	last month
2.0	Update 2.0	last month
2.1	Update 2.1	last month
3.1	Create 3.1	last month
3.2.	Create 3.2.	last month
4	Create 4	3 weeks ago
4.1.	Create 4.1.	3 weeks ago
4.3	Create 4.3	3 weeks ago
4.4	Create 4.4	3 weeks ago
4.5	Rename MeteoTower to 4.5	2 weeks ago
5.0	Update 5.0	2 weeks ago
5.1	Create 5.1	2 weeks ago



eclipse-workspace - MeteoTower/src/MeteoTower/MeteoTower.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Projects X Package Explorer X MeteoTower.java X

```
1 package MeteoTower;
2
3 import java.io.IOException;
4
5 public class MeteoTower {
6
7     //Se declaran tres variables estaticas que representan los umbrales de aviso para temperaturas
8     static double avisoTemperaturaAlta = 40; //C
9     static double avisoVientoFuerte = 50;// km/h
10    static double avisoLluviaTorrencial = 60;//mm/h
11
12    public static void main(String[] args) {
13
14        //Se crea un objeto Scanner para leer la entrada del usuario y se muestra un mensaje
15        Scanner entrada = new Scanner(System.in);
16
17        int opcion;
18
19        //Se utiliza un bucle do-while para mostrar continuamente el menu hasta que el usuario elija salir
20        do {
21            //Se crea el objeto date para conocer la fecha y hora actuales
22            Date fechaActual = new Date();
23
24            //Se muestra la fecha y hora actual en el terminal
25            System.out.println("Fecha y hora actual: " + fechaActual);
26
27            //Se muestra el menu principal
28            System.out.println("----- MENÚ PRINCIPAL -----");
29            System.out.println("1. Observacion Ultimas 24h");
30            System.out.println("2. Prediccion a 7 dias");
31            System.out.println("3. Prediccion a 48 h");
32            System.out.println("4. Historico Climatologico");
33            System.out.println("5. Salir");
34
35            //Se pide al usuario que introduzca una opcion
36            System.out.print("Introduzca una opcion:");
37
38            //Se lee la opcion introducida por el usuario
39            opcion = entrada.nextInt();
40
41            //Se ejecuta la opcion seleccionada
42            switch (opcion) {
43                case 1:
44                    System.out.println("Observacion Ultimas 24h");
45                    break;
46                case 2:
47                    System.out.println("Prediccion a 7 dias");
48                    break;
49                case 3:
50                    System.out.println("Prediccion a 48 h");
51                    break;
52                case 4:
53                    System.out.println("Historico Climatologico");
54                    break;
55                case 5:
56                    System.out.println("Salir");
57                    break;
58                default:
59                    System.out.println("Opcion no valida");
60            }
61
62        } while (opcion != 5);
63
64    }
65}
```

Console X

```
metetower [Java Application] C:\Users\anton\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe (19 jun 2023 20:20:45) [pid: 12900]
----- METEO TOWER -----
----- PUERTO DE NAVACERRADA, lat 40,793056 °N lon -4,010556° E -----
----- Información elaborada por la Agencia Estatal de Meteorología -----
----- Mon Jun 19 20:20:46 CEST 2023 -----
----- MENÚ PRINCIPAL -----
1. Observacion Ultimas 24h
2. Prediccion a 7 dias
3. Prediccion a 48 h
4. Historico Climatologico
5. Salir
Introduzca una opcion:
```

27°C Mayorm. nublado Búsqueda 21:20 19/06/2023



**Diseño y desarrollo de una aplicación para el monitoreo
del clima en torres de telecomunicaciones**

Funciones principales de MeteoTower

- Datos meteorológicos de las últimas 24 horas
- Predicción a 7 días
- Predicción a 48 horas
- Histórico de observaciones de días anteriores
- Activación de avisos
 - Ráfagas de viento fuerte
 - Lluvias torrenciales
 - Temperatura elevada

```
-----  
----- METEO TOWER -----  
--- PUERTO DE NAVACERRADA, lat 40,793056 °N lon -4,010556° E ---  
--- Información elaborada por la Agencia Estatal de Meteorología ---  
----- Wed Jun 14 18:49:11 CEST 2023 -----  
----- MENÚ PRINCIPAL -----  
1. Observacion Ultimas 24h  
2. Prediccion a 7 dias  
3. Prediccion a 48 h  
4. Historico Climatologico  
5. Salir  
Introduzca una opción:
```



```
-----  
Fecha y Hora: 2023-06-08T03:00:00  
Temperatura máxima: 40.0 * Aviso Temperatura Alta  
Temperatura minima: 9.8  
VMax (Km/h): 80.0 * Aviso Viento Fuerte  
Precipitaciones: 60.0 * Aviso Lluvia Torrencial  
-----
```

¿Por qué elegimos AEMET y JAVA?

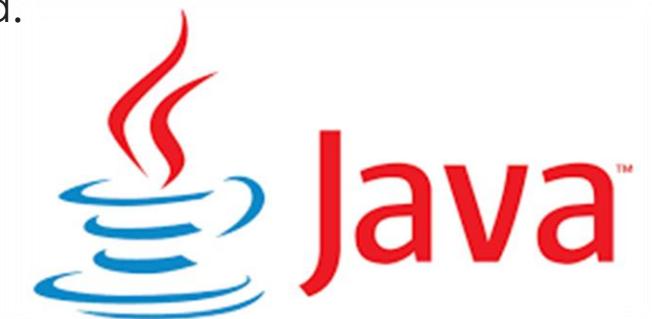


API gratuita, que ofrece:

- los datos meteorológicos en tiempo real de las últimas 24 horas.
- la predicción para los 7 días siguientes.
- el histórico de datos por día para los mismos parámetros.
- con garantía de continuidad.



- Lenguaje orientado a objetos, con una curva de aprendizaje muy rápida.
- Multitud de librerías, facilita la creación de aplicaciones distribuidas.
- Interpretado y compilado a la vez, independiente de la arquitectura.
- Robusto, con gestión eficiente de memoria y de alto rendimiento.



MeteoTower: Código



Importación de paquetes y declaración de umbrales de aviso.

```
import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.time.LocalDateTime;
import java.time.ZoneId;
import java.time.ZoneOffset;
import java.time.ZonedDateTime;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;
import java.util.Date;
import org.json.JSONArray;
import org.json.JSONObject;

public class MeteoTower {

    //Se declaran tres variables estaticas que representan los umbrales de aviso para temperatura alta, viento fuerte y lluvia torrencial
    static double avisoTemperaturaAlta = 40; //C
    static double avisoVientoFuerte = 50;// km/h
    static double avisoLluviaTorrencial = 60;//mm/h
```

MeteoTower: Código

Menú principal de MeteoTower.

```
//Se utiliza un bucle do-while para mostrar continuamente el menu hasta que el usuario elija la opcion
do {
    //Se crea el objeto date para conocer la fecha y hora actuales
    Date date = new Date();

    System.out.println("-----");
    System.out.println("----- METEO TOWER -----");
    System.out.println("-----");
    System.out.println("----- PUERTO DE NAVACERRADA, lat 40,793056 °N lon -4,010556° E -----");
    System.out.println("-----");
    System.out.println("----- Información elaborada por la Agencia Estatal de Meteorología -----");
    System.out.println("-----");
    System.out.println("----- " +date+ " -----");
    System.out.println("-----");
    System.out.println("----- MENÚ PRINCIPAL -----");
    System.out.println("1. Observacion Ultimas 24h");
    System.out.println("2. Prediccion a 7 dias");
    System.out.println("3. Prediccion a 48 h");
    System.out.println("4. Historico Climatologico");
    System.out.println("5. Salir");
    System.out.print("Introduzca una opción: ");
    opcion = entrada.nextInt();

    //Se utiliza una declaracion switch para ejecutar diferentes metodos segun la opcion seleccionada por el usuario
    switch (opcion) {
        case 1:
            ObservacionUltimas24h();
            break;
        case 2:
            Prediccion7dias();
            break;
        case 3:
            Prediccion48h();
            break;
        case 4:
            Historico();
            break;
        case 5:
            System.out.println("-----");
            System.out.println("----- METEO TOWER -----");
            System.out.println("-----");
            break;
        default:
            System.out.println("Opción inválida. Intente nuevamente.");
    }
} while (opcion != 5);
```



MeteoTower: Código

Lectura de ficheros JSON.

```
public static void ObservacionUltimas24h() {  
    //Se define una variable "respuesta" para almacenar la respuesta de la API.  
    String respuesta = "";  
  
    try {  
  
        //Se crea un objeto URL con la URL de la API que se va a llamar.  
        URL url = new URL("https://opendata.aemet.es/opendata/api/observacion/convencional/datos/estacion/2462?api_key=eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJsaW5vYWxtyXphb;FAZ21b  
  
        //Se crea objeto de conexión  
        HttpURLConnection conexion = (HttpURLConnection) url.openConnection();  
  
        //Método de conexión  
        conexion.setRequestMethod("GET");  
  
        //Conexión  
        conexion.connect();  
  
        //Respuesta de la API  
        int respuestaAPI = conexion.getResponseCode();  
  
        if(respuestaAPI != 200) {  
            System.out.println("Error");  
  
        } else {  
  
            //Se crea un objeto Scanner para leer el flujo de datos de la respuesta de la API  
            Scanner p = new Scanner(url.openStream());  
  
            while(p.hasNext()) {  
  
                //Se lee el contenido de la respuesta linea por linea y se almacena en la variable respuesta.  
                respuesta += p.nextLine();  
            }  
  
            //Se cierra el objeto Scanner  
            p.close();  
  
            new JSONObject(respuesta);  
        }  
    }  
}
```



Fichero JSON.

```
{  
    "descripcion": "exito",  
    "estado": 200,  
    "datos": "https://opendata.aemet.es/opendata/sh/06d41cb3",  
    "metadatos": "https://opendata.aemet.es/opendata/sh/98527307"  
}
```

MeteoTower: Código

Lectura de ficheros JSON.

```
//Se define una variable respuesta2 para almacenar la respuesta de la API
String respuesta2 = "";
try {

    //Se crea un objeto URL con la URL de la API que se va a llamar.
    URL url = new URL("https://opendata.aemet.es/opendata/sh/ic39dd1c");

    //Se crea objeto de conexión
    HttpURLConnection conexion = (HttpURLConnection) url.openConnection();

    // Método de conexión
    conexion.setRequestMethod("GET");

    //Conexión
    conexion.connect();

    //Respuesta de la API
    int respuestaAPI = conexion.getResponseCode();

    if (respuestaAPI != 200) {
        System.out.println("Error");
    } else {

        //Se crea un objeto Scanner para leer el flujo de datos de la respuesta de la API
        Scanner p = new Scanner(url.openStream());
        while (p.hasNext()) {

            //Se lee el contenido de la respuesta linea por linea y se almacena en la variable respuesta.
            respuesta2 += p.nextLine();
        }

        //Se cierra el objeto Scanner
        p.close();
    }
}
```

Fichero JSON.

```
[ {
    "idema" : "2462",
    "lon" : -4.010556,
    "fint" : "2023-06-08T14:00:00",
    "prec" : 0.0,
    "alt" : 1894.0,
    "vmax" : 14.1,
    "vv" : 6.8,
    "dv" : 196.0,
    "lat" : 40.793056,
    "dmax" : 207.0,
    "ubi" : "NAVACERRADA PUERTO",
    "pres" : 808.5,
    "hr" : 100.0,
    "stdv" : 2.8,
    "ts" : 10.3,
    "tamin" : 9.3,
    "ta" : 9.3,
    "tamax" : 10.0,
    "tpr" : 9.3,
    "vis" : 0.1,
    "stddv" : 16.0,
    "inso" : 0.0,
    "tss5cm" : 11.5,
    "vmaxu" : 26.6,
    "dvu" : 203.0,
    "pacutp" : 0.13,
    "vvu" : 6.8,
    "stdvvu" : 1.6,
    "stddvu" : 15.0,
    "dmaxu" : 203.0,
    "tss20cm" : 11.5,
    "geo850" : {
        "value" : 1469.9,
        "present" : true
    },
}
```



MeteoTower: Código

Obtención de los datos a partir de las JSONArray y JSONObject y conversión de hora UTC a local.

```
// Se crea un Array para modelar un dato concreto
JSONArray jsonArray = new JSONArray(respuesta2.toString());
int numElementos = jsonArray.length();

// Obtenemos la zona horaria de España
ZoneId zonaHorariaEspana = ZoneId.of("Europe/Madrid");
DateTimeFormatter formatoFechaHora = DateTimeFormatter.ofPattern("yyyy-MM-dd'T'HH:mm:ss");

// Se crea un array FechaHoras para almacenar las fechas y horas formateadas.
String[] FechaHoras = new String[numElementos];

// Se recorre el array JSON y se extrae la fecha y hora en formato UTC.
for (int i = 0; i < numElementos; i++) {
    JSONObject jsonObject = jsonArray.getJSONObject(i);
    String fechaHoraUTC = jsonObject.getString("fint");

    // Se convierte la fecha y hora de UTC a la zona horaria de España y se formatea.
    LocalDateTime fechaHoraLocal = LocalDateTime.parse(fechaHoraUTC, formatoFechaHora);
    ZonedDateTime fechaHoraZonaHorariaEspana = fechaHoraLocal.atZone(ZoneOffset.UTC)
        .withZoneSameInstant(zonaHorariaEspana);
    String fechaHoraFormatada = fechaHoraZonaHorariaEspana.format(formatoFechaHora);

    // Se guarda la fecha y hora formateada en el array FechaHoras
    FechaHoras[i] = fechaHoraFormatada;
}
```



MeteoTower: Código

```
//Se crea un array temperaturasMaximas para almacenar las temperaturas maximas de las ultimas 24h
double[] temperaturasMaximas = new double[numElementos];
for (int i = 0; i < numElementos; i++) {
    JSONObject jsonObject = jsonArray.getJSONObject(i);
    double temperaturaMaxima = jsonObject.getDouble("tamax");
    temperaturasMaximas[i] = temperaturaMaxima;
}

//Se crea un array temperaturasMinimas para almacenar las temperaturas minimas de las ultimas 24h
double[] temperaturasMinimas = new double[numElementos];

for (int i = 0; i < numElementos; i++) {
    JSONObject jsonObject = jsonArray.getJSONObject(i);
    double temperaturaMinima = jsonObject.getDouble("tamin");
    temperaturasMinimas[i] = temperaturaMinima;
}

//Se crea un array de precipitacionLluvia para almacenar las precipitaciones de las ultimas 24h
double[] PrecipitacionLluvia = new double[numElementos];

for (int i = 0; i < numElementos; i++) {
    JSONObject jsonObject = jsonArray.getJSONObject(i);
    double Precipitacion = jsonObject.getDouble("prec");
    PrecipitacionLluvia[i] = Precipitacion;
}

//Se crea un array de VelocidadMaxima para almacenar las velocidades maximas de las ultimas 24h
double[] VelocidadMaxima = new double[numElementos];

for (int i = 0; i < numElementos; i++) {
    JSONObject jsonObject = jsonArray.getJSONObject(i);
    double Velocidad = 3.6*jsonObject.getDouble("vmax");
    VelocidadMaxima[i] = Velocidad;
}
```



Obtención de los datos a partir de las
JSONArray y JSONObject.

MeteoTower: Código

```
// Se crea un JSONObject predicción para acceder al JSONObject predicciones
JSONObject predicción = jsonObject.getJSONObject("predicción");

// Se crea un JSONArray días para acceder al JSONArray día
JSONArray días = predicción.getJSONArray("día");

// Recorremos el JSONArray días
for (int i = 0; i < días.length(); i++) {
    JSONObject día = días.getJSONObject(i);

    //Obtenemos la fecha de la predicción a 48h
    String fecha = día.getString("fecha");
    System.out.println("Fecha: " + fecha);
```

Recorrido de varias JSONArray hasta llegar a la deseada.

Conversión "," en ".".



Verificar la existencia del dato deseado y pasarlo a "Double".

```
//Accedemos al JSONArray temperatura
JSONArray temperatura = día.getJSONArray("temperatura");

for (int j = 0; j < temperatura.length(); j++) {
    JSONObject temperaturaObject = temperatura.getJSONObject(j);

    // Verificar si existe value

    if (temperaturaObject.has("value")) {
        String temperatural = temperaturaObject.getString("value");
        String horatemperatural = temperaturaObject.getString("periodo");

        //Convertimos la temperatura de una String a un double
        double temperatura2 = Double.parseDouble(temperatural);
```

```
//Creamos un StringArray llamado temperaturasMaximas
String[] temperaturasMaximas = new String[numElementos];

//Creamos un DoubleArray a partir del StringArray llamado temperaturasMaximas
double[] temperaturasMaximasl = new double[numElementos];

//Recorremos el JSONArray
for (int i = 0; i < numElementos; i++) {
    JSONObject jsonObject = jsonArray.getJSONObject(i);
    String temperaturaMaxima = jsonObject.getString("tmax");

    //Convertimos las "," en "."
    temperaturaMaxima = temperaturaMaxima.replace(",", ".");

    //Convertimos la String temperaturaMaxima en un double
    double temperaturaMaximal = Double.parseDouble(temperaturaMaxima);
    temperaturasMaximas[i] = temperaturaMaxima;
    temperaturasMaximasl[i] = temperaturaMaximal;
```

MeteoTower: Código

Mostrar los valores por consola.

```
//Se imprime la observación de las últimas 24 h
System.out.println("Observación últimas 24h");
System.out.println("-----");
for (int i = 0; i < numElementos; i++) {

    System.out.println("Fecha y Hora: " + FechaHoras[i]);

    //Verificamos si supera el umbral máximo para emitir un aviso de temperatura alta
    if(temperaturasMaximas[i] >=avisoTemperaturaAlta) {
        System.out.println("Temperatura máxima: " + temperaturasMaximas[i] + " * Aviso Temperatura Alta");
    }
    else{
        System.out.println("Temperatura máxima: " + temperaturasMaximas[i]);
    }

    System.out.println("Temperatura mínima: " + temperaturasMinimas[i]);

    //Verificamos si supera el umbral máximo para emitir un aviso de viento fuerte
    if(VelocidadMaxima[i] >=avisoVientoFuerte) {

        //Se utiliza el método Math.round() para redondear el número a 2 decimales
        System.out.println ("VMax(Km/h): " + Math.round( VelocidadMaxima[i] * 100) / 100d + " * Aviso Viento Fuerte");
    }
    else{
        System.out.println("VMax(Km/h): " + Math.round( VelocidadMaxima[i] * 100) / 100d);
    }

    //Verificamos si supera el umbral máximo para emitir un aviso de lluvia torrencial
    if(PrecipitacionLluvia[i] >=avisoLluviaTorrencial) {
        System.out.println("Precipitaciones: " + PrecipitacionLluvia[i] + " * Aviso Lluvia Torrencial");
    }
    else{
        System.out.println("Precipitaciones: " + PrecipitacionLluvia[i]);
    }
    System.out.println("-----");
}
```



Pruebas de validación

La aplicación presenta los datos coincidentes con los que se descargan de la web de la AEMET y realiza correctamente la conversión de unidades.

```
-----  
Fecha y Hora: 2023-06-15T16:00:00  
Temperatura máxima: 20.2  
Temperatura mínima: 17.9  
VMax (Km/h): 18.36  
Precipitaciones: 0.0
```

```
[ {  
    "idema" : "2462",  
    "lon" : -4.010556,  
    "fint" : "2023-06-15T14:00:00",  
    "prec" : 0.0,  
    "alt" : 1894.0,  
    "vmax" : 5.1,  
    "vv" : 2.0,  
    "dv" : 337.0,  
    "lat" : 40.793056,  
    "dmax" : 268.0,  
    "ubi" : "NAVACERRADA PUERTO",  
    "pres" : 818.1,  
    "hr" : 52.0,  
    "stdvv" : 1.3,  
    "ts" : 25.3,  
    "tamin" : 17.9,  
    "ta" : 20.0,  
    "tamax" : 20.2,  
    "tpr" : 9.8,  
    "vis" : 20.0,  
    "stddv" : 64.0,  
    "inso" : 46.2,  
    "tss5cm" : 13.5,  
    "vmaxu" : 5.1,  
    "dvu" : 342.0,  
    "pacutp" : 0.0,  
    "vvu" : 2.2,  
    "stdvvu" : 0.9,  
    "stddvu" : 42.0,  
    "dmaxu" : 270.0,  
    "tss20cm" : 13.2,  
    "geo850" : {  
        "value" : 1556.7,  
        "present" : true  
    },  
}
```



Pruebas de validación

```
----- METEO TOWER -----  
--- PUERTO DE NAVACERRADA, lat 40,793056 °N lon -4,010556° E ---  
--- Información elaborada por la Agencia Estatal de Meteorología ---  
----- Fri Jun 16 15:55:35 CEST 2023 -----  
  
--- MENÚ PRINCIPAL ---  
1. Observacion Ultimas 24h  
2. Prediccion a 7 dias  
3. Prediccion a 48 h  
4. Historico Climatologico  
5. Salir  
Introduzca una opción: 70  
Opción inválida. Intente nuevamente.  
  
----- METEO TOWER -----  
--- PUERTO DE NAVACERRADA, lat 40,793056 °N lon -4,010556° E ---  
--- Información elaborada por la Agencia Estatal de Meteorología ---  
----- Fri Jun 16 16:10:23 CEST 2023 -----  
  
--- MENÚ PRINCIPAL ---  
1. Observacion Ultimas 24h  
2. Prediccion a 7 dias  
3. Prediccion a 48 h  
4. Historico Climatologico  
5. Salir  
Introduzca una opción:
```

El menú principal termina el programa seleccionando la opción 5. Salir.



La aplicación activa los avisos en caso de superación.

```
-----  
Fecha y Hora: 2023-06-08T03:00:00  
Temperatura máxima: 40.0 * Aviso Temperatura Alta  
Temperatura minima: 9.8  
VMax(Km/h): 80.0 * Aviso Viento Fuerte  
Precipitaciones: 60.0 * Aviso Lluvia Torrencial  
-----
```

Lecciones aprendidas

- Dificultades con la lectura de ficheros JSON
- Confusión entre velocidad del viento y racha (ingeniería inversa)
- Hora local vs hora UTC
- Datos por consola en vez de interfaz gráfica
- Lectura de los datos climatológicos históricos
- Conversión de unidades





MeteoTower: video



Casos de uso

Caso 1

Alarma de **avería urgente** por pérdida de servicio en un radioenlace de Cliente en un emplazamiento con torre que implica trabajo en altura. El **gestor de mantenimiento correctivo** consulta el estado actual (por lluvia, viento o temperatura elevada) para decidir desplazarse a resolver o no.



Caso 2

Planificación a corto plazo de **trabajo de instalación** de equipos de 5G en una torre, previsto para la semana siguiente. El **gestor de despliegue** busca el día más adecuado desde el punto de vista meteorológico, descartando los días con mayor probabilidad de situaciones desfavorables.





Casos de uso

Caso 3

Planificación de **mantenimiento preventivo anual** de la infraestructura de la torre de telecomunicaciones. El **gestor de mantenimiento preventivo** analiza el registro pasado de la meteorología del emplazamiento para determinar la época del año donde hay menor probabilidad histórica de precipitaciones, vientos fuertes o temperaturas elevadas.



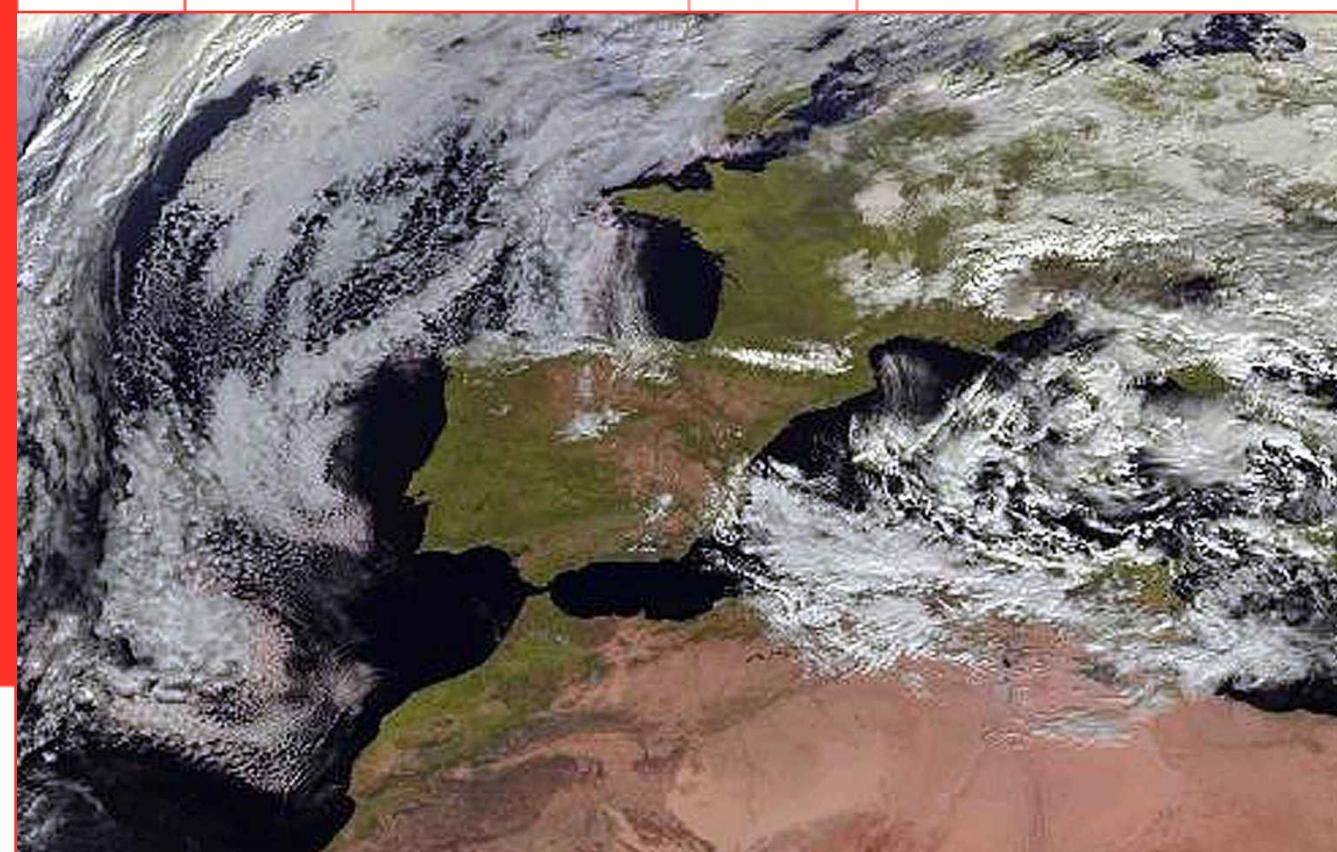
Caso 4

Activación de avisos por temperaturas elevadas, lluvias torrenciales o vientos fuertes, con la entrada en vigor del Plan Nacional de actuaciones preventivas de los efectos de los excesos de temperatura, para que el **gestor responsable de los trabajos** adopte las medidas oportunas de protección, como la reducción o modificación de las horas de desarrollo de la jornada de trabajo prevista.



04

Conclusiones



Resultados del proyecto

Conforme a los requerimientos iniciales:

- ✓ Aplicación programada en **Java**, utilizando una API **gratuita** de clima con información en **tiempo real**.
- ✓ Para el emplazamiento seleccionado, se muestran los siguientes datos:
 - ✓ Datos de observación de las últimas 24 horas.
 - ✓ Predicción a 7 días.
 - ✓ Predicción a 48 horas.
 - ✓ Histórico de datos diarios.
- ✓ Activación de **avisos** por condiciones extremas de **temperatura, viento y lluvia**.
- ✓ Los datos se presentan por **consola**.



Conclusiones

La empresa que utilice la aplicación **MeteoTower** tendrá las siguientes mejoras:

- Disminución de los desplazamientos infructuosos, debido a las inclemencias meteorológicas.
 - Reducción de las jornadas de trabajo perdidas.
 - Reducción de los kilómetros realizados.
- Aumento de la seguridad de los trabajadores ante fenómenos meteorológicos adversos.
 - Disminución de accidentes trabajando (altura, eléctrico, obras de construcción).
 - Disminución de accidentes “in misión”.



Aprendizaje basado en proyectos

Competencias básicas

- Redactar y desarrollar proyectos en el ámbito de la ingeniería.
- Aplicar elementos básicos de economía y de gestión de recursos humanos, planificación de proyectos, así como de legislación en las telecomunicaciones.
- Capacidad de trabajar en un grupo multidisciplinar y de comunicar.
- Transmitir información, ideas, problemas y soluciones al público.

Competencias transversales

- Valores éticos.
- Trabajo en equipo.
- Análisis y resolución de problemas.
- Adaptación al cambio.



05

Futuras líneas de trabajo



Líneas de trabajo futuras

- Selección de cualquier punto de España, determinando la estación AEMET más cercana.
- Ampliación de los datos mostrados: índice UV, nieve, granizo, ...
- Ampliación de avisos por fenómenos extremos y definición de umbrales por el usuario.
- Notificación vía correo electrónico de avisos por fenómenos extremos.
- Exportación de datos a Excel o fichero csv.
- Presentación de los datos en interfaz gráfica, con iconos del estado del cielo.
- Creación de aplicación para teléfonos móviles, además de la versión web.
- Análisis por IA del histórico climatológico para planificar del mantenimiento preventivo.





**Universidad
Europea**

Preguntas

Ve más allá

06

Backup



Escala de Beaufort

Número de Beaufort	Velocidad del viento (km/h)	Nudos (millas náuticas/h)	Denominación	Aspecto del mar	Efectos en tierra
0	0 a 1	< 1	Calma	Despejado	Calma, el humo asciende verticalmente
1	2 a 5	1 a 3	Ventolina	Pequeñas olas, pero sin espuma	El humo indica la dirección del viento
2	6 a 11	4 a 6	Flojito (Brisa muy débil)	Crestas de apariencia vítreas, sin romper	Se caen las hojas de los árboles, empiezan a moverse los molinos de los campos
3	12 a 19	7 a 10	Flojo (Brisa Ligera)	Pequeñas olas, crestas rompiendo.	Se agitan las hojas, ondulan las banderas
4	20 a 28	11 a 16	Bonancible (Brisa moderada)	Borreguillos numerosos, olas cada vez más largas	Se levanta polvo y papeles, se agitan las copas de los árboles
5	29 a 38	17 a 21	Fresquito (Brisa fresca)	Olas medianas y alargadas, borreguillos muy abundantes	Pequeños movimientos de los árboles, superficie de los lagos ondulada
6	39 a 49	22 a 27	Fresco (Brisa fuerte)	Comienzan a formarse olas grandes, crestas rompiendo, espuma	Se mueven las ramas de los árboles, dificultad para mantener abierto el paraguas
7	50 a 61	28 a 33	Frescachón (Viento fuerte)	Mar gruesa, con espuma arrastrada en dirección del viento	Se mueven los árboles grandes, dificultad para caminar contra el viento
8	62 a 74	34 a 40	Temporal (Viento duro)	Grandes olas rompiendo, franjas de espuma	Se quiebran las copas de los árboles, circulación de personas muy difícil, los vehículos se mueven por sí mismos.
9	75 a 88	41 a 47	Temporal fuerte (Muy duro)	Olas muy grandes, rompiendo. Visibilidad mermada	Daños en árboles, imposible caminar con normalidad. Se empiezan a dañar las construcciones. Arrastre de vehículos.
10	89 a 102	48 a 55	Temporal duro (Temporal)	Olas muy gruesas con crestas empenachadas. Superficie del mar blanca.	Árboles arrancados, daños en la estructura de las construcciones. Daños mayores en objetos a la intemperie.
11	103 a 117	56 a 63	Temporal muy duro (Borrasca)	Olas excepcionalmente grandes, mar completamente blanca, visibilidad muy reducida	Destrucción en todas partes, lluvias muy intensas, inundaciones muy altas. Voladura de personas y de otros muchos objetos.
12	+ 118	+64	Temporal huracanado (Huracán)	Olas excepcionalmente grandes, mar blanca, visibilidad nula	Voladura de vehículos, árboles, casas, techos y personas. Puede generar un huracán o tifón



Trabajos en campo





Trabajos en campo





Trabajos en campo



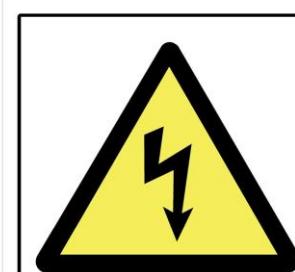
Cartelería PRL



ANTENA
RF



SOLO
PERSONAL
AUTORIZADO



RIESGO
ELECTRICO



USO
OBLIGATORIO
DE CASCO



ES OBLIGATORIO
USAR ARNES
DE SEGURIDAD



MeteoTower



**Diseño y desarrollo de una aplicación para el monitoreo
del clima en torres de telecomunicaciones**



**Universidad
Europea**

Gracias

Ve más allá