FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Weather VIS

Diogo Henrique de Almeida Silva Pereira

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Maria Teresa Magalhães da Silva Pinto de Andrade

July 12, 2021

Weather VIS

Diogo Henrique de Almeida Silva Pereira

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. João Correia Lopes
External Examiner: Prof. José Moreira
Supervisor: Prof. Maria Teresa Andrade

July 12, 2021

Abstract

As of late, weather prediction has become increasingly challenging as a result of climate changes. To address this predictive challenge, a large number of weather stations have been collecting data over the years for a better understanding of the real climate and micro-climates present in planet Earth. The information gathered ranges from variables such as temperature, precipitation, pressure, humidity, and wind intensity. By using this information climate experts can uncover trends in the weather, which are essential to the predicative formula in order to more accurately portrait the future weather. The stations that collect data operate all over the globe and at multiple heights, most are located at ground level with an altitude matching the Earth's topography and are built using metal structures that support all the sensors. There are also weather balloons that gather data directly at the lower atmosphere level and weather buoys which float at ocean level passively collecting data.

Currently, weather stations only collect and transmit data using specific communication protocols. This specialized functionality means that they don't offer a common application programming interface (API) to display the recorded values and historical values, making the available data hard to read and visualize. Adding to this problem, stations can sometimes fail or be poorly calibrated, which increases the amount of misinformation. This problem not only impacts field experts but also daily users, who resort to other means to obtain weather report and forecast information, such as: news channels, radio broadcasts, web services, etc.

This master's thesis aims to create a visualization tool that agglomerates the data from multi-network weather station's reports and displays it in a user friendly manner. The data has been gathered from open-source databases and have been displayed over a global interactive map. To facilitate this process, a geographical information system (GIS) has been used, allowing the platform to more accurately display information related to surface meteorological variables over the specific location where they were recorded. To help on the conception process, similar existing applications here taken into consideration, these applications acquire both public and private station data and provide that information to its users, displaying it on a map specific to the user's location. Some features that help this project stand out from the state of the art, is the extensive available historical data and the use of the Earth's terrain in the display of the weather phenomena, this approach is relevant due to the difference in altitude directly affecting the weather. To achieve this all the data has been converted to sea level, using the respective formulas, and then calculated according to the terrain altitude in the location of that station's reading.

Keywords: Weather report, Surface meteorological elements, Geographical information system.

Resumo

Ultimamente, a previsão do tempo tornou-se cada vez mais desafiante devido a mudanças climáticas. Para combater este desafio, um elevado número de estações meteorológicas tem recolhido dados ao longo dos anos, o que provínciona um conhecimento mais aprofundado do clima e dos microclimas presentes no planeta Terra. Os dados recolhidos remetem a variáveis como temperatura, precipitação, pressão, humidade e intensidade do vento. Utilizando estas informações os especialistas do clima conseguem descobrir tendências no tempo. Estas são essenciais para retratar com mais precisão o tempo futuro. As estações que recolhem dados operam pelo globo e em várias altitudes. A maioria está localizada no nível da terra e são compostas por estruturas de metal que suportam múltiplos sensores, também existem balões meteorológicos que recolhem dados diretamente no nível inferior da atmosfera e boias meteorológicas que flutuam no oceano e recolhem dados passivamente.

Atualmente, as estações meteorológicas apenas recolhem e transmitem dados utilizando protocolos específicos de comunicação. Essa funcionalidade especializada significa que as estações não oferecem uma interface comum para mostrar os valores registados e valores históricos, o que tornando os dados disponíveis difíceis de visualizar e interpretar. Para acrescentar a este problema, as estações podem falhar ocasionalmente ou estar mal calibradas, o que aumenta a quantidade de informação incorreta. Este problema não só influencia os especialistas deste campo, como também os utilizadores diários, que recorrem a outros meios para obter informações meteorológicas tais como: canais de notícias, emissões de rádio, serviços web, etc.

Esta dissertação de mestrado tem como objetivo criar uma ferramenta que junta os dados de múltiplos relatórios de estações meteorológicas e mostra-os de forma acessível e fácil de navegar. Estes foram retirados de bases de dados de públicas e são colocados num mapa global interativo. Para facilitar este processo, um sistema de informações geográficas foi utilizado, isto permite à plataforma exibir com maior precisão as informações remetentes às variáveis meteorológicas no local específico onde foram registadas. Para auxiliar no processo de conceção, foram consideradas aplicações semelhantes já existentes. Estas adquirem dados de estações públicas e privadas e fornecem essa informação aos seus utilizadores através de um mapa específico para localização do onde se encontra o mesmo utilizador. As características que salientam este projeto do estado da arte existente são: a quantidade extensiva de dados históricos disponíveis e a utilização do terreno da Terra na visualização dos fenómenos meteorológicos. Esta abordagem é relevante devido à diferença de altitude afetar diretamente o clima. Para alcançar esta meta todos os dados foram convertidos para o nível do mar, utilizando fórmulas respetivas, e depois foram calculados de acordo com a altitude do terreno no local da leitura.

Palavras-chave: Boletim meteorológico, Elementos meteorológicos de superfície, Sistema de informação geográfica

Acknowledgements

I would like to express my gratitude towards everyone that helped make this work possible.

Firstly, to Professor Maria Teresa Andrade, my supervisor, for her dedication, organization and helpfulness. Her support and comprehensiveness were always remarkable and unparalleled, be it regarding work or any other matters.

Secondly, to Engineer Vitor Teixeira for enduring all of the endless meetings and issues we ran into together while working on this project.

Finally, to my friends, for providing an escape when the work seemed overwhelming, to my family, for always believing in me, and to my girlfriend for always being there for me and picking me back up when I was down.

Diogo Henrique de Almeida Silva Pereira

"It is the mark of an educated mind, to entertain a thought without accepting it."

Aristotle

Contents

List of Figures

List of Tables

Abbreviations

API	Application Programming Interface
GIS	Geographical Information System
GSOD	Global Surface Summary of the Day
NOAA	National Oceanic and Atmospheric Administration
SYNOP	Surface Synoptic Observations
WMO	World Meteorological Organization
3D	Three-Dimensional
ºC	Degree(s) Celsius
ISH	Integrated Surface Hours
USAF	United States Air Force
NCDC	National Climatic Data Center
CSV	Comma-Separated Values
ID	Identifier
IDP	Integrated Communication Project
NCO	Central Operations
NCEP	National Center for Environmental Prediction
NESDI	National Environmental Satellite, Data and Information Service
NCDC	National Climate Data Center
OAR	Ocean and Atmospheric Research
ESRL	Earth System Research Laboratory
CSD	Global Systems Division
RTTY	Radioteletype
ICAO	International Civil Aviation Organization
TAF	Terminal Airport Forecast
AGL	Above Ground Level
GTS	Global Telecommunications System
BUFR	Binary Universal Form for the Representation of Meteorological Data
AMDAR	Aircraft Weather Data Relay
VHF	Very High Frequency
WWW	World Wide Web
OPC	Open Geospatial Consortium
netCDF	Network Common Data Form
GUI	Graphical User Interface
CPU	Central Processing Unit
CSV	Comma-separated Values
FEUP	Faculty of Engineering, University of Porto
SWOT	Strengths, Weaknesses, Opportunities and Threats
QoS	Quality of Service

Chapter 1

Introduction

An overview of the present report is explained in this chapter. Firstly, the context and the motivation behind this work. Secondly, the detailing of its goals. Lastly, explaining the methodologies and expected results stemming from its developments.

1.1 Context and Motivation

In recent times, the ongoing climate changes pose an ever-growing difficulty to predict the weather forecast [Scher and Messori, 2019], the major factor being the decrease in the temperature difference between the North Pole and the Equator. To alleviate this arduous task, and achieve a better understanding of the real climate and micro-climates in planet Earth, weather scientists take advantage of a huge number of weather stations, that have been collecting data continuously. However, this alone, is not an solution to the weather forecast problem, as weather stations simply collect data and transmit it in a non user-friendly way and, since they don't offer a common application programming interface (API), precision and calibration results are sometimes dispersed.

This thesis project was proposed by iClimate Adviser (a startup company specialized in providing personalized information about the weather and climate) as a solution to solve the problem of miss information and poor visualization of weather information gathered by the stations. Proposing the development of a weather API that collects all the available data from the existing public weather stations and displays it over a virtual map, making use of the current computational power to solve a real-world situation. Furthermore, the Earth's surface shape and features have a significant impact on its weather and the creation of micro-climates. Keeping that in mind, the visualization of meteorological elements interpolated with a three-dimensional (3D) layer of the terrain's relief, is an interesting development that has been further explored in this thesis.

1.2 Goals

The main goal of this project is to answer the following research question: Is it possible to create a platform which agglomerates weather data from multiple weather stations and interpolates the temperature with the Earth's terrain, outputting all the information in a user friendly way?

The current work aims to develop an application that has access to public meteorological data and displays the desired element, be it temperature, precipitation, humidity, wind strength or atmospheric pressure, with the respective color identifier, over an interactive map. It also interpolates temperature data to form an information grid, that is used in conjunction with the terrain's elevation to calculate more accurate values.

1.3 Methodologies and Expected Results

To decide the path of development of the project, weekly meetings were scheduled with both Professor supervisor Maria Teresa Andrade and Engineer Vitor Teixeira from iClimate Adviser, separately. The meetings with the company served as a direct interaction with the client, where all the desired features where registered and qualified once implemented. The meetings with Professor Teresa were essential to the production of the thesis document and the validation of the technological choices along with the discussion of multiple solutions to the diverse challenges encountered.

The work begins with a detailed study of the literature in existing weather information APIs. Comparing the already existing work developed in his area has kick-start the project by demonstrating what is the convention established and how it can be improved.

After that, the source of the meteorological data must be chosen. An extensive research into the multiple weather communication protocols has been performed to understand their purpose and how they operate. The collection of the available weather report databases has also been essential, in order to weigh the pros and cons of each one, allowing for an educated choice regarding the origin of the used data.

Following that, the geographic information system (GIS) provider must be selected for the map overlay. Certain characteristics must be considered, such as the impact on the real-time performance of the application, available documentation, free and open source, available features, etc., must be evaluated to find the most suitable for the project. It is also important to understand whether these GIS frameworks are compatible with the type of information that needs to be displayed or if there is any unwanted development setbacks in the long run due to deprecation.

Finally, the results have been analyzed and compared with other weather APIs, taking into account how similar the information is to the reality, how fast and responsive the project solution is and how user-friendly the information is displayed.

It is expected that the project is developed successfully, allowing for the display of the selected weather data over a movable map of the surface of planet Earth. The accessibility of the application has also been considered, striving for a seamless human-device interaction, meaning fast response times and a simple and easy to use interface.

1.4 Document Structure

The next sections of this report are structured as follows: Chapter 2 aims to provide some contextualization on the concepts and lexicon of Earth's climate and terrain as well as GIS. Chapter 3 contains the state of the art for the existing developed weather forecast and report applications, GIS desktop visualizers and frameworks, weather communication protocols and available weather and terrain information datasets. Chapter 4 describes the planned implementation steps, the chosen architecture for implementing a solution and the technological choices. Chapter 5 depicts the solution used for implementing the projected architecture. Chapter 6 shows the performed tests on the implemented solution. Finally, Chapter 7 presents a summary of the report, states the conclusions reached and a summary of the possible future work.

Chapter 2

Contextualisation

This chapter will serve to contextualize the reader in regards to the definition of Earth's weather and topography, as well as GISs and the data types they use. Ending with a summary of the chapter to consolidate the ideas mentioned.

2.1 Weather

"The state of the atmosphere with respect to heat or cold, wetness or dryness, calm or storm, clearness or cloudiness." [Merriam-Webster, 2003]. Common weather phenomena include clouds, wind, fog, rain, snow, and sandstorms. Less common events correspond to natural disasters such as hurricanes, tornadoes, ice storms and typhoons. Most weather phenomena occur in the lowest section of the Earth's atmosphere: the troposphere [Glickman, 2000]. It should be noted that weather and climate have different meanings. Weather refers to daily occurrences of atmospheric conditions, while climate refers to the average atmospheric conditions over a long period of time.

The weather is determined by the differences in air temperature, pressure and moisture between different locations. These differences occur because the angle at which sunlight strikes the Earth, varies which the current season, location an time of the day, all due to Earth's rotation around its tilted axis and it's orbit around the Sun. On the surface of the Earth, the temperature usually varies between ±40 degrees ºC. In the following thousands of years, changes in the Earth's orbit will affect the amount of solar energy arriving and how it is distributed at the Earth, thereby affecting long-term climate. Places at higher altitudes are cooler than those at lower altitudes, this is because most atmospheric heating originates from contact with the Earth's surface, while the Sun's radiation is constantly being lost to space. If this was not the case a greenhouse effect would occur, drastically changing the weather [Le Treut et al., 2007].

Another important aspect of the Earth's weather is the ozone layer. This is a region located on the stratosphere that absorbs most of the Sun's ultraviolet radiation, protecting life from its overexposure. But the use of CFC, HCFC and halon gases causes a chemical reaction with the

Ozone breaking it down to Oxygen, causing the depletion of the Ozone layer, which in turn allows for radiation to reach the surface of the Earth [Parson, 2003].

2.1.1 Weather report

"A systematic statement of the existing and usually the predicted meteorological conditions over a particular area." [Merriam-Webster, 2003]. Weather reports contain meteorological information regarding the Earth's weather conditions at current and past times.

A typical weather report displays the high and low temperatures of the current day accompanied by the average temperature of that day. It also compares how much higher or lower the results are regarding the norm, which is based on values recorded thorough the years.

2.1.2 Weather forecasting

"To predict weather conditions on the basis of correlated meteorological observations." [Merriam-Webster, 2003]. Weather forecasting is an application of science and technology that can predict the future weather conditions. It uses past and present weather data to create trends and bases future weather events in the analysis of these trends. Due to not being an exact science, risk and uncertainty are central to forecasting and prediction, usually the degree of uncertainty is mentioned in forecasts. To minimize uncertainty the used data must be up to date in order for the forecast to be as accurate as possible. In some cases the data used to predict the variable of interest is itself forecast [French, 2016].

2.2 Topography

"The art or practice of graphic delineation in detail usually on maps or charts of natural and man-made features of a place or region especially in a way to show their relative positions and elevations." [Merriam-Webster, 2003].

Topography is the study of the Earth's surface characteristics. It lays the underlying foundation of a landscape, highlighting the existing valleys, mountains, craters and rivers on the surface, it may also incorporate artificial features such as cities, administrative boundaries, landmarks, parks, transportation, buildings and hydrography, this being a science that studies the bottom of the ocean.

In topographic maps the terrain's elevation is the distinguishing factor, as such some maps display the it through contour lines, each line limiting a section where all the data inside it have an equal height value. Other maps use a color scheme to highlight the differences in the terrain (i.e. the brighter a place is the higher the terrain is on that location).

2.2.1 Digital Elevation Models

A digital elevation model (DEM) is a 3D computer graphic representation of elevation data used to represent the surface of the Earth. They are the most common basis for digital production of relief maps, which are maps that render the texture of the terrain. DEMs are the main source used in GISs to model the terrain and can be represented as a raster or as a vector-based triangular irregular network, which a network containing triangle meshes to represent a surface.

2.3 Geographic Information System

Geographic information system (GIS) is an analytical tool that provides the ability to analyze spatial and geographic data. GISs are used in a variety of technologies and applications related to the following: business, management, planning, insurance, transportation/logistics, telecommunications, and engineering [Maliene et al., 2011]. GIS uses the location as a "key index variable" in order to relate available information. This means that for every location there will always be a date and time of occurrence and geographical coordinates.

GIS applications are computer-based programs that allow users to store and edit spatial and non-spatial data, it then displays the data as a map. The user can create interactive queries to specify the desired specified output (i.e. Zooming in on a map) [Clarke, 1986]. Existing examples of programs that use GISs as well as GIS development frameworks will be mentioned in the following Chapter 3.

2.3.1 GIS uncertainties

The accuracy of GIS depends on the source of its data and how it is encoded to reference it. Land surveyors have been able to use GPS-derived locations to provide a high level of location accuracy [Committee et al., 1998]. However, high resolution digital terrain and aerial images are not the only sources of data. For example, paper maps were widely used before the introduction of powerful computers and web technologies, which changed the quality, practicability and expectations of GISs.

When developing a digital terrain database for GIS, topographic maps are the main source of information, while aerial photography and satellite imagery are additional sources for collecting data and identifying attributes that can be mapped. The scale of the map and the map projection, which is the way the globe is flattened into a plane, are very important aspects. This is because information content mainly depends on the scale set and resulting locatability of the map's representations.

Quantitative analysis of maps brings accuracy issues into focus. The equipment used for GIS measurements is much more accurate than the machines used for conventional map analysis. All geographic data is inherently inaccurate, and these inaccuracies will propagate in GIS operations in unpredictable ways [Couclelis, 2003].

2.3.2 Raster Data

A raster consists of a matrix of cells organized into rows and columns where each cell contains a value representing information, much like an image. Examples of raster data used in GISs are: digital aerial photographs, imagery from satellites, digital pictures and scanned maps.

There are two types of raster values: discrete rasters, which have distinct themes or categories (i.e. one grid cell represents a soil type), and non-discrete rasters, which are grid cells with gradual changing data (i.e. elevation mapping).

Since raster data is basically an image it its very easy to understand its concept, making the process of identifying specific positions very simple due to the cells having a numbered position. Algebra operations with raster data are quick and easy to perform. Overall, quantitative analysis is intuitive with discrete or non-discrete rasters.

Despite this, the result can be pixelated due to the cell size contributing to the graphics quality, which makes linear features and paths difficult to display. In addition, raster datasets can become very large because they record the value of each pixel in the image and as the resolution increases the number of pixels increase. Conversely, the size of the cell will decrease which comes at the cost of processing speed and data storage.

2.3.3 Vector Data

Data in this format consists of lines and polygons comprised of individual points stored as coordinate pairs that indicate a physical location in the world.

Because vector data has vertices and paths the graphical output is more easily discerned and since data isn't dependent on grid size the geographic accuracy is higher compared to raster data. However, continuous data is poorly stored and to display it as a vector, it would require substantial generalization. Even tho topology is useful for vector data, it is often processing intensive since vector manipulation algorithms are complex.

2.4 Summary

This section explains what are considered the weather phenomena, the difference between weather report and weather forecast, the latter refers to the predicted of the weather and the other to the current and past weather. It also explains what is the terrain topography and what exactly is a GIS framework, how accurate can they be and the types of data they can use.

Chapter 3

State of the Art

This chapter describes the researched state of the art relating to existing weather data visualization APIs, GIS desktop visualizers and GIS software development frameworks, available weather datasets and finally, weather communication protocols. Ending with a summary of the chapter to consolidate the ideas mentioned.

3.1 Weather data APIs

3.1.1 IMPA

The Portuguese Institute of the Sea and the Atmosphere (Instituto Português do Mar e da Atmosfera, IPMA) is the meteorological, atmospheric, sea and seismic national organization of Portugal.The current IPMA was originally created in 1944 as the National Meteorological Service (Serviço Metereológico Nacional, SMN).

It combines the services of the formerly existing meteorological departments (civil aviation, army, military aviation, navy and agriculture), regional meteorology in the Azores Bureau and some university meteorological institutes.

IPMA displays current, past and forecast information in mainland Portugal, Azores islands and Madeira islands on their website [1]. Global information is also available, but at a very reduced quantity, due to using only open-source stations. This is expected as IMPA is a Portuguese association that is responsible for Portugal's national earth and atmosphere domains.

IMPA's website allows users to view common weather information in a map, using Leaftet (Chapter 3.3.1) to provide GIS tools. It also provides the option to consult specific station's data in the form of graphs to see how the weather progresses, in that area, through time. Detailed temperature and precipitation grids are also available and they appear to use topography as a way to obtain more accurate information.

[1]Instituto Português do Mar e da Atmosfera - `https://www.ipma.pt/pt/index.html`" - Last seen in 2021-02-25

3.1.2 Nullschool Earth

Nullschool's "earth" is a project which has the objective to visualize global weather conditions. These are updated every three hours by using forecast calculated with supercomputers. The project makes use of multiple different map scales: a simplified, larger scale to display animations and a more detailed, smaller scale for static display.

Its weather Data is provided by the Global Forecast System (GFS) and the geographic data is provided by Natural Earth (Chapter 3.4.3) but requires formatting in TopoJSON format for the API to work [Beccario, 2016].

3.1.3 Windy

Windy is a private Czech company, created by Ivo Lukačovič in November 2014, that provides current, historical and forecast weather information worldwide [2]. The information is displayed using animations on a map using a GIS framework developed by them. Initially, it focused only on the animation of wind, and now provides other basic meteorological parameters such as temperature, pressure, relative humidity, cloud cover and other more advanced data. Wind animation is based on the aforementioned open-source project of Cameron Beccario: "earth" [Beccario, 2016].

3.1.4 Weather Underground

Weather Underground is private subsidiary company, founded in 1995 by Jeff Masters, owned by The Weather Company, which originated from the famous International Business Machines Corporation (IBM) [Holloway, 2016]. It offers a weather service that displays current weather conditions, by using the WunderMap feature available in their website [3]. They gather information from the National Weather Service (NWS) and from over 250000 personal weather stations. This data is then displayed using a GIS tool on an interactive map. It also provides historical and current station data to a multitude of newspapers and weather reports, however, the access to this data is only available through a paid subscription.

3.1.5 OpenWeatherMap

OpenWeatherMap is an online service provided by the company OpenWeather, founded in 2012 by Denn Ukolov and Olga Ukolova. It provides current and historical global weather data and forecasting services, all displayed using their developed API, which also uses a GIS and its available on their website [4]. They use convolutional machine learning to associate data from multiple sources: weather satellites, ground radar stations, airport weather stations, remote sensing satellites, automated weather stations and METAR, explained on Section 3.5.3.

The current weather data can be accessed freely for any desired project just by making a query to their website. Historical data however is paid with a subscription.

[2] Home @ Windy Community - `https://community.windy.com//`" - Last seen in 2021-02-25
[3] OpenWeatherMap - `https://openweathermap.org/`" - Last seen in 2021-02-25
[4] Weather Underground Website - `https://www.wunderground.com/`" - Last seen in 2021-02-25

3.1.6 Fogos.pt

Fogos.pt is a Portuguese open-source project [5] which maps occurrences of existing and possible fires in Portugal. In their map, which uses Leaflet (researched on Section 3.3.1) to display data with GIS precision, users can view the exact location of a fire, the time it started and the type of area it affects (i.e. agricultural field, forest, etc.). The website also presents the amount and type of resources used to combat the fire, the timeline of the situation and the meteorological data of the affected area (i.e. mean, max and min temperatures, wind speed, humidity, etc.). All this data is provided by the Portuguese Civil Protection (Proteção Civil Portuguesa).

The UI lets the user choose to see a color grid of the temperature, pressure, wind, precipitation and/or clouds, these are layers are provided by Mapbox.

3.1.7 Dark Sky

Dark sky is a free application, bought by Apple in March 2020 [Business, 2020], that allows user to view the current and historical and forecast weather data in desktop and mobile environments. This data ranges from: temperature, clouds, precipitation, wind speed, ultra violet indexes, pressure and ozone. All current information is displayed on the map which uses OpenLayers (Chapter 3.3.2) for GIS accuracy. The user can consult historical and forecast data with a ten day interval from today. It is also possible to consult older historical data, but it can only be viewed through graphs.

3.2 GIS desktop visualizers

3.2.1 Google Earth

Google Earth is probably the most popular developed GIS. It is a computer program, previously named Keyhole EarthViewer. This software renders a 3D representation of Earth based primarily on satellite imagery. The program draws a map of the earth by superimposing satellite images, aerial photography and GIS data on the 3D earth, so that users can view the city and scenery from all angles. Users can explore the Earth by entering an address or coordinates, or even by using the keyboard or mouse to explore the Earth. The users can also use the touch screen or stylus to download the program on their smartphone or tablet. Personalized data can be added to the program using Keyhole markup language which then uploads them through various resources (i.e. forums or blogs). Google Earth can display various images covering the Earth's surface, and it is also a Web Map Service client. Recently, Google revealed that Google Earth now covers 98% of the world and has captured 10 million miles of street view images [Nieva, 2019].

In addition to navigation, Google Earth also provides other globes for the moon and Mars, as well as tools for viewing the night sky and a flight simulator game. Other features allow users to view photos uploaded to Google Map Views of various locations, information provided by Wikipedia of those locations, and street view images. The web-based application also includes

[5] FOGOS.pt Github - https://github.com/fogospt" - Last seen in 2021-02-25

Voyager, a feature that regularly adds in-app tours, usually introduced by scientists and documentary makers.

Google Earth also has very limited weather report display, allowing to display satellite photographs of the clouds on top of its interactive map.

3.2.2 ArcGIS

ArcGIS is a GIS desktop application maintained by the Environmental Systems Research Institute (Esri) and implemented on C++ Programming Language. Released in 1999 and maintained and updated until 2020 [6], it provides an infrastructure for making maps and analyzing their information, editing geographic data, sharing and discovering geographic information.

Esri's conversion to the ArcGIS platform made the third-party add-in software and scripts developed by many users incompatible. Due to changes in scripting functions, functions, operating systems and a large number of system resources required by ArcGIS software, a small number of user groups cannot migrate to ArcGIS [Fee, 2008].

ArcGIS problems include the high price of the product, the use of proprietary formats, and the difficulty of importing and exporting data between Esri and other GIS software [Nasr, 2007].

3.2.3 QGIS

QGIS is a free to use and open source cross-platform desktop GIS application based on the programming languages: C++, Python and Qt, which supports viewing, editing and analyzing geospatial data [7]. It supports a variety of raster, vector and grid data formats, and uses a plug-in architecture to easily add new support.

They started development in July 2002 and are still maintaining the software [qgis, 2021]. The application's goal is to provide common functions and features while making GIS easy to use. The initial intention of the project was to provide a GIS data viewer and they clearly achieved it since many people have used it for daily GIS data viewing and editing needs.

3.2.4 PostGIS

PostGIS is an open-source software plugin that facilitates the use of geographic objects on the PostgreSQL relational database. PostGIS follows the specification of the Open Geospatial Consortium (OGC) and is implemented as an external extension of PostgreSQL [8] which is based on C programming language.

3.2.5 Smallworld

Smallworld is a proprietary GIS software product provided by General Electric Digital. This product has gradually grown into a global market leader in utilities and communications GIS,

[6] ArcCIS - `https://desktop.arcgis.com/en/arcmap/`" - Last seen in 2021-02-25
[7] QGIS Official Website - `https://www.qgis.org/en/site/`" - Last seen in 2021-02-25
[8] PostGIS - Spatial and Geographic objects for PostgreSQL - `https://postgis.net/`" - Last seen in 2021-02-25

and has maintained its position until today. Smallworld technology also supports applications in telecommunications and utilities[9].

The company designed and optimized a database technology called Version Management Data Storage (VMDS) to store and analyze complex spatial and topological data, and to support alternative versions of data during long-term transactions.

3.2.6 Global Mapper

Global Mapper is a GIS software package developed by Blue Marble Geographics that provides geospatial professionals with an extensive array of spatial data processing tools [10]. Global Mapper handles both vector, raster, and elevation data, and provides general GIS features such as viewing and conversions. Global Mapper has an active user community with a mailing list and online forums.

3.3 GIS Software development frameworks

3.3.1 Leaflet

Leaflet is an open-source JavaScript (JS) library [11], originally created by Vladimir Agafonkin and released in May 2011, that allows simple GIS functionalities to developers. It is used in most of the API examples researched in this Chapter. It allows users to create a JS "Map" object that displays map data from other sources (i.e. Esri, OpenStreetMap, etc.). The documentation is very accessible and simple the consult, and implementing a functional map can be done quickly by following the start guide.

3.3.2 OpenLayers

OpenLayers is another open-source JS library [12], originally created by the company MetaCarta and released in June 2006. It extremely identical to Leaftet, the notable differences being that OpenLayers has much more lines of code compared to Leaflet [Software, 2014] and its heavier 423 KiloBytes (KB) (Openlayers) vs 123 KB (Leaflet). Despite this it does offer more options than Leaflet and has a larger and more complex API, which requires more time to understand and operate than Leaflet's.

3.3.3 Cesium

Cesium is an open-source library [13], developed in 2011, that can be implemented using JS, Unreal Engine or Ion. It allows for mobile and desktop 2D and 3D maps, it has similar functionalities to

[9] Smallworld GIS - Geospatial Asset Management - https://www.ge.com/digital/applications/smallworld-gis-geospatial-asset-management/" - Last seen in 2021-02-25
[10] Global Mapper - https://www.bluemarblegeo.com/global-mapper/" - Last seen in 2021-02-25
[11] Leaflet Github - https://github.com/Leaflet/Leaflet" - Last seen in 2021-02-25
[12] OpenLayers Github - https://github.com/openlayers/openlayers" - Last seen in 2021-02-25
[13] Cesium Github - https://github.com/CesiumGS/cesium" - Last seen in 2021-02-25

the already described frameworks, but it's documentation is not as clear and easy to consult as the others Cesium Doc.

3.4 Available Weather Datasets

3.4.1 Global Surface Summary of the Day

The Global Surface Summary of the Day (GSOD) is derived from the Integrated Surface Hours (ISH) dataset and is available to the public through the National Oceanic and Atmospheric Administration (NOAA). The ISH dataset includes global data obtained from the United States Air Force (USAF) Climatology Center located in the Federal Climate Complex with National Climatic Data Center (NCDC). Generally, the latest daily summary data can be obtained 1-2 days after the date and time when the observation value is used in the daily summary.

The global summary of the day data for the surface meteorological elements, are derived from the synoptic/hourly observations contained in USAF DATSAV3 Surface data and Federal Climate Complex Integrated Surface Hourly [Air Force Combat Climatology Center, U.S. Air Force, U.S. Department of Defense et al., 2003]. When the date data summary is obtained, at least 4 observations of a day must be displayed. Since the data is converted to constant units, there may be some rounding errors compared to the originally reported value. The daily average is based on the working hours of the workstation. Daily extreme values and totals will only appear when the data reported by the weather station is sufficient to provide valid values. Therefore, the frequency of occurrence of these three elements will be lower than the other values.

The online data files range from 1929 to today with over 9000 stations' data available. The daily elements included in the dataset are:

- Mean temperature (Fahrenheit)
- Mean dew point (Fahrenheit)
- Mean sea level pressure (Millibars)
- Mean station pressure (Millibars)
- Mean visibility (Miles)
- Mean wind speed (Knots)
- Maximum sustained wind speed (Knots)
- Maximum wind gust (Knots)
- Maximum temperature (Fahrenheit)
- Minimum temperature (Fahrenheit)
- Precipitation amount (Inches)

Table 3.1: GSOD file format [of Commerce, 2021]

FIELD	DESCRIPTION
STATION	Station number (WMO/DATSAV3 possibly combined w/WBAN number)
DATE	Given in mm/dd/yyyy format
LATITUDE	Given in decimated degrees (Southern Hemisphere values are negative)
LONGITUDE	Given in decimated degrees (Western Hemisphere values are negative)
ELEVATION	Given in meters
NAME	Name of station/airport/military base
TEMP	Mean temperature for the day in degrees Fahrenheit to tenths. Missing = 9999.9
TEMP_ATTRIBUTES	Number of observations used in calculating mean temperature.
DEWP	Mean dew point for the day in degrees Fahrenheit to tenths. Missing = 9999.9
DEWP_ATTRIBUTES	Number of observations used in calculating mean dew point.
SLP	Mean sea level pressure for the day in millibars to tenths. Missing = 9999.9
SLP_ATTRIBUTES	Number of observations used in calculating mean sea level pressure.
STP	Mean station pressure for the day in millibars to tenths. Missing = 9999.9
STP_ATTRIBUTES	Number of observations used in calculating mean station pressure.
VISIB	Mean visibility for the day in miles to tenths. Missing = 999.9
VISIB_ATTRIBUTES	Number of observations used in calculating mean visibility.
WDSP	Mean wind speed for the day in knots to tenths. Missing = 999.9
WDSP_ATTRIBUTES	Number of observations used in calculating mean wind speed.
MXSPD	Maximum sustained wind speed reported for the day in knots to tenths. Missing = 999
GUST	Maximum wind gust reported for the day in knots to tenths. Missing = 999.9
MAX	Maximum temperature reported during the day in Fahrenheit to tenths. Missing = 9999.9
MAX_ATTRIBUTES	Blank indicates maximum temperature was taken from the explicit maximum temperature report and not from the 'hourly' data. indicates maximum temperature was derived from the hourly data.
MIN	Minimum temperature reported during the day in Fahrenheit to tenths. Missing = 9999.9
MIN_ATTRIBUTES	Blank indicates minimum temperature was taken from the explicit minimum temperature report and not from the 'hourly' data. indicates minimum temperature was derived from the hourly data.
PRCP	Total precipitation reported during the day in inches and hundredths; will usually not end with the midnight observation. "0" indicates no measurable precipitation (includes a trace). Missing = 99.99
PRCP_ATTRIBUTES	A = 1 report of 6-hour precipitation amount. B = Summation of 2 reports of 6-hour precipitation amount. C = Summation of 3 reports of 6-hour precipitation amount. D = Summation of 4 reports of 6-hour precipitation amount. E = 1 report of 12-hour precipitation amount. F = Summation of 2 reports of 12-hour precipitation amount. G = 1 report of 24-hour precipitation amount. H = Station reported '0' as the amount for the day, but also reported at least one occurrence of precipitation in hourly observations. This could indicate a trace occurred, but should be considered as incomplete data for the day. I = Station did not report any precipitation data for the day and did not report any occurrences of precipitation in its hourly observations. It's still possible that precipitation occurred but was not reported.
SNDP	Snow depth in inches to tenths. It is the last report for the day if reported more than once. Missing = 999.9
FRSHTT	Indicators (1 = yes, 0 = no/not reported) for the occurrence during the day of: Fog ('F' - 1st digit). Rain or Drizzle ('R' - 2nd digit). Snow or Ice Pellets ('S' - 3rd digit). Hail ('H' - 4th digit). Thunder ('T' - 5th digit). Tornado or Funnel Cloud ('T' - 6th digit).

- Snow depth (Inches)
- Indicator for occurrence of: Fog, Rain or Drizzle, Snow or Ice Pellets, Hail, Thunder, Tornado/Funnel Cloud. (Binary)

3.4.1.1 GSOD database format

The dataset groups each year's meteorological information in a folder, inside each folder are comma-separated values (CSV) files containing the data of a weather station, the files are named after the station Identifier (ID), which is a string of ten digits. The structure of the CSV can be consulted on Table 3.1

3.4.2 MADIS

MADIS is a meteorological observational database and data delivery system that provides observations that cover the globe.

MADIS obtains data from NOAA data sources and non-NOAA providers, decodes the data, and then encodes all observation data into a common format with uniform observation units and time stamps. It then performs quality inspections and stores the integrated data set with a series of flags that indicate the results of various quality control (QC) inspections. MADIS provides users with several ways to access data to meet their needs. Users can request data from July 2001 (that is, the first public use of MADIS) to the present [14].

In order to provide these services, MADIS utilizes partnerships with international agencies; federal, state and local agencies; universities; volunteer networks and the private sector in order to integrate and QC observations from their stations with those of NOAA.

As part of the Integrated Communication Project (IDP), MADIS runs on the Central Operations (NCO) of the National Center for Environmental Prediction (NCEP). All MADIS data will be added to the NOAA data archive operated by the National Environmental Satellite, Data and Information Service (NESDIS) National Climate Data Center (NCDC). NOAA Office of Ocean and Atmospheric Research (OAR) Earth System Research Laboratory (ESRL) Global Systems Division (GSD) is providing the current research and development work.

3.4.3 Natural Earth

Natural Earth is free to use map dataset available at 1:10, 1:50, and 1:110 million map scales. Natural Earth's data set contains integrated vector and raster mapping data [15].

Natural Earth tries to solve the problem of finding appropriate data for creating small-scale maps, in a WWW over saturated with confusing and poorly attributed geospatial data. Besides that, the data attributes are highly important for mapmaking. Most data contains embedded feature names, which are ranked by relative importance. Other attributes facilitate faster map production, such as width attributes assigned to river segments for creating tapers.

Although the initial dataset was authored by Tom Patterson and Nathaniel Vaughn Kelso, Natural Earth has broadened to be a collaboration of many volunteers and is now endorsed and promoted by the North American Cartographic Information Society (NACIS) [Kelso, 2018].

3.5 Weather Communication Protocols

3.5.1 CLIMAT

A CLIMAT report consists of climatological information from a unique meteorological station for one specific month [Organization, 2009]. These reports are sent and exchanged through the World Meteorological Organization (WMO) and, in order to make processing by software easier, they follow a restrict syntax. They are also divided into up to five sections, depending on the quality and amount of detail of the information provided and contain information of average pressure

[14] NCEP Meteorological Assimilation Data Ingest System - `https://madis.ncep.noaa.gov/`" - Last seen in 2021-12-03

[15] Natural Earth - `https://www.naturalearthdata.com/`" - Last seen in 2021-12-03

at station level, average air temperature, maximum and minimum air temperature average vapor pressure, total precipitation, and total sunshine [NOAA, 2021]. This information is essential for monitoring the characteristics, changes and variability of the climate.

There are also modified forms of CLIMAT codes i.e. CLIMAT SHIP and CLIMAT TEMP / CLIMAT TEMP SHIP codes. These are used to report monthly climate data collected at marine meteorological ground observatories and terrestrial/marine meteorological meteorological high-altitude observatories, respectively. Generally, the included monthly values are obtained by averaging the observations of one or more daily observations in the corresponding month [Organization, 2017].

The data reported by CLIMAT is widely used in meteorological and climate applications and the continuous improvement in the quality and quantity of CLIMAT reports has improved the production of these products.

3.5.2 Surface Synoptic Observations

Surface Weather Observation (SYNOP) is a digital code used to report weather observations from manual and automatic weather stations. This type of information is collected by more than 7,600 manned and unmanned weather stations and more than 2,500 mobile stations around the world. The reports are then sent, mostly by the German Meteorological Service "Deutscher Wetterdienst", using shortwave and low frequency using Radioteletype (RTTY) every six hours.

The report consists of a set of numbers describing general weather information, such as temperature, visibility and pressure recorded at the weather station. This information can be decoded using open source software such as seaTTY [16], metaf2xml [17] or Fldigi [18] and mainly used for climatic statistics and weather forecasting.

3.5.3 METAR

Meteorological Aerodrome Report (METAR) is another format for reporting weather information that was standardized by the International Civil Aviation Organization (ICAO). Because of this it is mainly used by airplane pilots, however it is also used by meteorologists in weather forecasting.

Raw METAR is the most frequent observation weather data transmission format in the world. This data usually comes from an airport or permanent weather observation station. At most sites, reports are generated either every hour or every half an hour. Nevertheless, if there is a significant change in the location conditions of the staff, a report called special (SPECI) can be released.

Some METARs are coded by automatic weather stations established predominantly in airports and military bases. Having said that, information can come from other locations that still use enhanced observations. These observations are documented by digital sensors, coded in software, and then verified by certified weather observers or forecasters before transmission. Observations can also be carried out and encoded by licensed observers or forecasters.

[16] DX soft - https://www.dxsoft.com/en/products/seatty/" - Last seen in 2021-12-03
[17] metaf2xml website - https://metaf2xml.sourceforge.io" - Last seen in 2021-12-03
[18] Fldigi website - http://www.w1hkj.com/" - Last seen in 2021-12-03

3.5.4 Terminal aerodrome forecast

In the field of meteorology and aviation, Terminal Airport Forecast (TAF) is a format used to report weather forecast information, especially directed to aviation-related usage. It is issued every six hours a day to major civil airports, with information containing the next 24-30 hours range. For military airports and certain civil airports, TAF is issued every three hours, with data containing the next 3 to 30 hours.

They are made by human forecasters on the ground and supplements METAR reports, this accomplished due to both using similar codes. In comparison to METAR and weather forecasts, TAF information sources are much more reduced, but since they are more accurate than digital weather forecasts, because they consider local geographic influences, they are preferred as a complement to METAR when used for aviation.

3.5.5 Binary Universal Form for the Representation of meteorological data

The Binary Universal Form for the Representation of meteorological data (BUFR) is a binary data format created in 1988 and maintained by the WMO. Its goal is to replace dozens of character-based, location-driven weather codes, such as CLIMAT and SYNOP. It is designed to be portable, compact and versatile and can represent any kind of data, as well as its specific temporal/spatial context and any other associated metadata. Despite this, BUFR has some flaws[Caron, 2011] and the use of this format resulted in data "separation" and many formatting errors [Hand, 2016].

3.5.6 Aircraft Meteorological Data Relay

Aircraft Weather Data Relay (AMDAR) was established by the WMO and it takes advantage of existing aircraft onboard sensors, computers and communications systems to collect, process, format and transmit meteorological data to ground stations via satellite or radio links [WMO, 2018].

This data is collected globally through the use of commercial aircraft's navigation system and on-board standard temperature and static pressure probes. [Organization, 2003]. AMDAR transmissions are widely used in forecast models as a complement to radiosonde data, which is data provided by weather balloons.

3.5.7 General Regularly-distributed Information in Binary form

The General Regularly-distributed Information in Binary (GRIB) format is used to store historical and weather forecast data. It is standardized by the Basic Systems Committee of the WMO, numbered GRIB [World Meteorological Organization, 2003]. A GRIB file is an agglomerate of independent records of 2D data where each document exists separately, without reference to other records or the overall structure. Therefore, a collection of GRIB files can be attached to each another, and documents can be separated from a collection. Every GRIB document consists of two sections. The first is the header, which is mandatory and describes the document and the actual binary data itself. The second is the Grid Description Section which includes location, resolution and grid data information.

Even though is a lightweight communication protocol there are some problems with using GRIB [Caron, 2011]

The significant problems are:

- Each record is independent and cannot refer to the expected mode of the GRIB writer;
- There is no infallible way to combine records into a multidimensional array from where documents are derived;
- There is a need to use extrinsic tables to describe the data;
- No reliable place where centers can publish their tables;
- Inconsistent and imprecise techniques of versioning local tables;
- No machine-readable versions of the WMO tables.

3.6 Summary

After the collection of the state of the art correspondent to the project and its careful examination, an informed decision can be taken regarding the platform/source from which the weather information is obtained as well as what GIS development tool is more adequate to display the information on a map. The existing weather APIs served as inspiration and guiding line to this project, the final project results has also been compared to the output of this applications. Weather communication protocols were researched, but they have not been used in the project, due to being extremely low level, compared to the high level nature of this project, despite this they are important to understand as they are the "backbone" of weather information.

Chapter 4

Proposed Solution

In this chapter, there will be an overview of the projected solution's architecture, along with its technological choices. Afterwards, a summary of the chapter.

4.1 Architecture

After some initial meetings with the thesis supervisor professor and iClimate Adviser, it was decided that a desktop website was the best suited architecture to embody the spirit of this project. Right after that, the user requirements were gathered using user stories to describe the features that the website needed. Those user stories can be seen on Table 4.1.

Figure 4.1 displays the general architecture of the project using an information flow. One thing to notice is that the figure only shows the information flow and does not take into account the communication between the different components that was required to achieve it, that information will be explained in more detail in the following Chapter.

Taking the user requirements into account, to satisfy them successfully this website has a backend component and a frontend component. The backend is composed of a database where all the meteorological data is stored and an executable with the respective queries and commands to interact with the stored information. The frontend is be the website itself, containing the an interative map with the displayed weather information, queried from the database, superimposed.

It should also be noted that the project's should be highly scalable. This means that, in future development, it should process multiple types of data, from multiple sources, displaying them in a plethora ways, requiring only some minor changes to the code.

4.2 Technological Choices

Having decided on the layout of the project, the selection of what technologies to use in it begins. Taking into consideration the technologies investigated in the state of the art (SOTA) section, an-

Table 4.1: User Stories

Title	**User Story**	**Priority**
Map movement	As a user, I want to control the zoom and position of the map, so that I can view any desired region of Earth	High
Date control	As a user, I want to change the date, so that I can view the weather information information specific to that time	High
Station Data	As a user, I want to view weather station data in the form a colored marker, so that I can visualize it better	High
Info Selector	As a user, I want to change the type of information displayed, so that I can view other types of meteorological data	High
Temperature Grid	As a user, I want to view the mean temperature as a color grid, so that I can view information in locations where there are no station data	High
View Topography	As a user, I want to view the topography shader displayed on the map, so that I can better visualize how it influences the weather information	Medium
Station Information	As a user, I want to view information relative to a station by hovering over it, so that I can access all the information of a specific station	Medium
Legend Info	As a user, I want to view a legend regarding the information displayed, so that I can better understand it	Low
Map Type	As a user, I want to change the map type, so that I can chose to see the map with a more realistic appearance	Low
Chart Display	As a user, I want to view weather information displayed in a chart, so that I can view the development of that information during a selected time period	Low

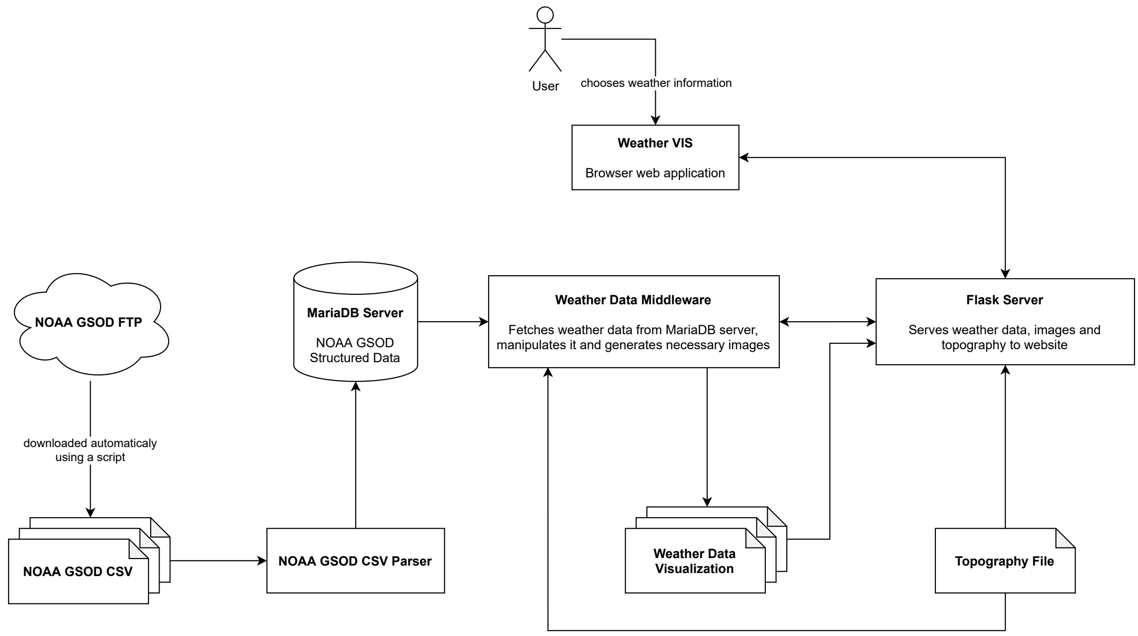


Figure 4.1: Information Flow

other in-depth research, directed towards aspects not directly related with the theme of the project, was conducted to create an available plethora of choices from which an informed decision could be made.

Starting with the source of the weather data to display on the map. The important factors the database must contain, gathered from initial meetings with the company, are free and available timed historical records, from stations located all over the globe, with data relative to the following meteorological aspects: atmospheric temperature, amount of precipitation, wind speed, atmospheric pressure and humidity percentage. As mentioned on the SOTA segment the available datasets are: NOAA's GSOD, MADIS, Natural Earth, Weather Underground and OpenWeather. Taking into account the already acquired information about the aforementioned datasets, one shall be chosen to cater the project needs.

Since Weather Underground and OpenWeather are both private and payed databases, they were immediately discarded as options.

Natural Earth was also considered, but, as it only contains raster and vector mapping data, requiring specific protocols to decrepit and make use of the data the way the project intends, creating undesired complexity, it was also discarded for now.

Leaving the two strongest candidates: NOAA's GSOD and MADIS, which where were weighted and discussed, in initial meetings, to balance the pros and cons of using each one. Being almost identical, because MADIS is an integration of NOAA data sources with non-NOAA providers, as demonstrated on the SOTA segment. Both have a wide array of stations all over the globe with data from the year 1929 until now. They not also register all the information required for the project, but have additional data that can be useful and used in further development.

One of the differences is the way the information is presented. MADIS database are stored using netCDF files, which is a community standard for sharing scientific data and requires external libraries to be read and interpreted. GSOD has numbered CSV files corresponding to the IDs of the stations. A very simple and straightforward way storing the information, facilitating the work of reading and storing the information. The use of CSV files makes it so that it is possible to query them with any text editor, in order to compare the original file information with the stored one.

Another difference is the level of access to the available information. MADIS has restrictions on the use of some of the non-NOAA data and requires a data application form to download the data, while the NOAA data has no restrictions whatsoever. Said restricted information corresponds to: Aircraft Based, Hydrological Surface, Meteorological Surface and Radiometer information, all of them being redundant for the projects main goal. It should also be noted that both GSOD and MADIS have a three day delay on the available information, meaning data from 8th of May will only be available on the 11th.

Because of those differences, NOAA's GSOD was the final choice to develop upon this project, as it has the most simple and easy to understand way of information storage, requiring only a developed parser to automate the decryption and transfer process of the information. By choosing only one source instead of multiple, allows the program to be as simple as possible. It should be noted that open-source stations can record unreliable information and sometimes not transmit any data in specific days.

The next step is to decide what programming languages to develop the project. Many where considered, the more notable being: C++, C, C Sharp (C#), Python, Java, JavaScript and Go. The factors that were taken into account were: easy of use, documentation, available features and compatibility with the project. Since the project is divided into two different sections, each one has been developed with a different programming paradigms.

For the backend, which will be delve into more detail in the next subsection, Python was the preferred choice, due to its very easy-to-understand nature, very complete set of libraries which are instrumental in the development of the project and the previous knowledge and experience using this language, acquired from multiple projects developed for academic course units.

For the frontend, which will also be discussed more in-depth in its respective subsection, JavaScript (JS) was the final choice. This was due to the fact that, JS has an innate compatibility with web development applications by having a lot a lot of libraries that can be used to add functionalities to a website. In the case of this project, most of the GIS development frameworks use JS as a main language.

All the writing of the project code was done and edited on the text editor: Visual Studio Code (vscode). The only other considered option being Sublime Text. The choice was made due to the previous familiarity of using vscode.

All the work was done and tested using the same hardware and using Windows 10 (W10) operating system (OS). A Asus ZenBook UX430UNR laptop which has the following hardware specifications: CPU - Intel QuadCore i7-8550U (1.8GHz), Integrated GPU - Intel UHD Graphics

620, External GPU - NVIDIA GeForce MX150 2GB GDDR5 VRAM, Storage - 256GB SSD, RAM - 16GB LPDDR3.

4.2.1 Backend

The backend component of this project serves as a means to read and store the data, making all numerical operations necessary and also providing the desired data to the frontend, where it is displayed to the user.

As mentioned above the selected programming language was a specific version of Python3. This version was chosen has it has access to very useful libraries that allows the program to read CSV data files, store them in the database, process requests from the frontend, interpolate data and manipulate images. All these aspects are crucial for the project. Another very strong candidate, that meets all the requirements, was Go. The advantage it has over Python is: it does not have the dependencies problem. This means, when the program is executed in a a new environment, there is no need to install all the required packages in order for it to work. But, due to the easy-to-use nature of Python and the lack of knowledge in using Go, Python was the preferred choice.

To simplify and organize the development using Python packages, a Python-environment distribution was chosen. The choice was the bare-bones version of Anaconda: Miniconda. Which creates python environments where the version can be chosen and containing only the desired packages for development.

Next challenge was to create the database. For this, a Unified Modeling Language (UML) class diagram was created to aid the construction of the layout of the relational database (Figure 4.2).

To create this UML an in-depth consultation of the documentation of GSOD files was done using this link GSOD documentation. The classes and attributes where chosen to best store the all the weather aspects. Has seen on the diagram, each station is unique. They are identified by their IDs and have their respective coordinates, name and country associated

With the main classes of the database defined, a management system has been chosen to abstract the interaction with the data, providing also data safety, concurrent access, fault tolerance, integrity and scalability. The chosen system was a fork of the popular MySQL named MariaDB both using Structured Query Language (SQL). It was chosen over MySQL due to having better query performance overall, better testing and bug fixes which are not found in MySQL [Bartholomew, 2012]. NoSQL databases options were also studied, such as MongoDB and Apache Cassandra[], which are non-relational. They offer some benefits over SQL implementations [Stonebraker, 2010], but force the architecture of the database to be redesigned, coupling this with having low knowledge of using NoSQL tools, they were discarded. Since MariaDB operates using SQL, all the queries to the database where developed using SQL statements and syntax.

To handle the administration of the MariaDB, phpMyAdmin was used. This decision takes into consideration that phpMyAdmin has very low security, due to only being required a user + password combo to access the database. But since, it is used for development, and not production, the risk is negligible. This administration tool provides a intuitive web interface that allows for multiple MySQL features such as dropping and creating tables, creating complex queries, sorting the data, etc.

With this, all the technological choices relating to the database are made. To organize its dependencies the Docker software was chosen. It works similar to Anaconda, as it creates a virtual "container" that is specific to the selected project. It also allows it to run on most OSs. Since running the docker on W10 requires virtualization to the Linux kernel [Merkel, 2014], it adds complexity to the program. To bypass this challenge it was run directly on a Linux console using Windows Subsystem for Linux (WSL).

To communicate with the frontend section of the project, the Python web framework Flask was used. Since Node.js and Express are both JavaScript based frameworks they were not considered for this project. The other Python option considered was Bottle, but due to their similarity [Taneja and Gupta, 2014], Flask was the final choice, due to having previous knowledge in using it.

As mentioned on the projects goals, the project interpolates registered temperatures with the topography of the map. In order to do this, a DEM must be chosen to access elevation data. A small research was made in regards to existing DEMs. Information gathered on GTOPO30 [Miliaresis and Argialas, 1999], ASTER [Baldridge et al., 2009], SRTM [Van Zyl, 2001], Earth2014 [Hirt and Rexer, 2015] and GMTED2010 [Danielson and Gesch, 2011], but the client specifically asked for GTOPO30 to be used on this project.

4.2.2 Frontend

The frontend component of this project serves has the output of the information to the user in the form of a website. It contains all the website essential technologies such as: HyperText Markup Language (HTML), which defines the meaning and structure of the web content. Cascading Style Sheets (CSS), which is a style sheet language used to define the way the website looks. And JS, which can make network requests for resources, trigger events and react to them. The use of JS frameworks such as Angular and React, was debated. They abstract and simplify the creation of the User Interface (UI), among other things. Due to the simplicity of the website's UI the use of frameworks is redundant.

Almost all of the aforementioned user requirements will be displayed in the frontend section, notably the map overlay, which leads to the next technological choice: the GIS framework. The available choices already researched are the GIS desktop application: ArcGIS, Global Mapper, Smallworld, PostGIS, QGIS and Google Earth, and the GIS development frameworks: Leaflet, OpenLayers, OpenStreetMap, GeoBase and Cesium. Since the GIS desktop application are standalone programs, most being payed, they were not considered for this project. This decision allows for more freedom in development, has the coding process is not constricted to those programs functioning behavior. While not being chosen, they served as inspiration and guiding lines for the development of the GIS map. Then the natural progression is to use the much more flexible and lightweight GIS frameworks. These consist on JS helper libraries that allow the use of GIS properties such as mapping the Earth's terrain on a website, allowing the creation of markers using a real world geographic coordinate system (GCS), displaying custom layers on top of a map, etc.

All the mentioned frameworks were considered and the final solution was Leaflet. This is due to it having the best documentation, being very easy to understand and develop, having a lot of additional open-source features developed by other users and being used in most of the examples researched for this thesis.

4.3 Summary

This project is separated into two sections, the backend which handles all the weather data and the frontend which displays it in a user-friendly way, allowing the user to request the desired type of data. The source of the data is going to be the GSOD provided by NOAA. For the backend the data is stored in SQL tables using Python and MariaDB. Python also allows for image manipulation and interpolation of results. The frontend has been developed in JS using the Leaflet helper to draw the map and display the information on it.

Having decided on the structure of the project and its the technological choices, the implementation process starts.

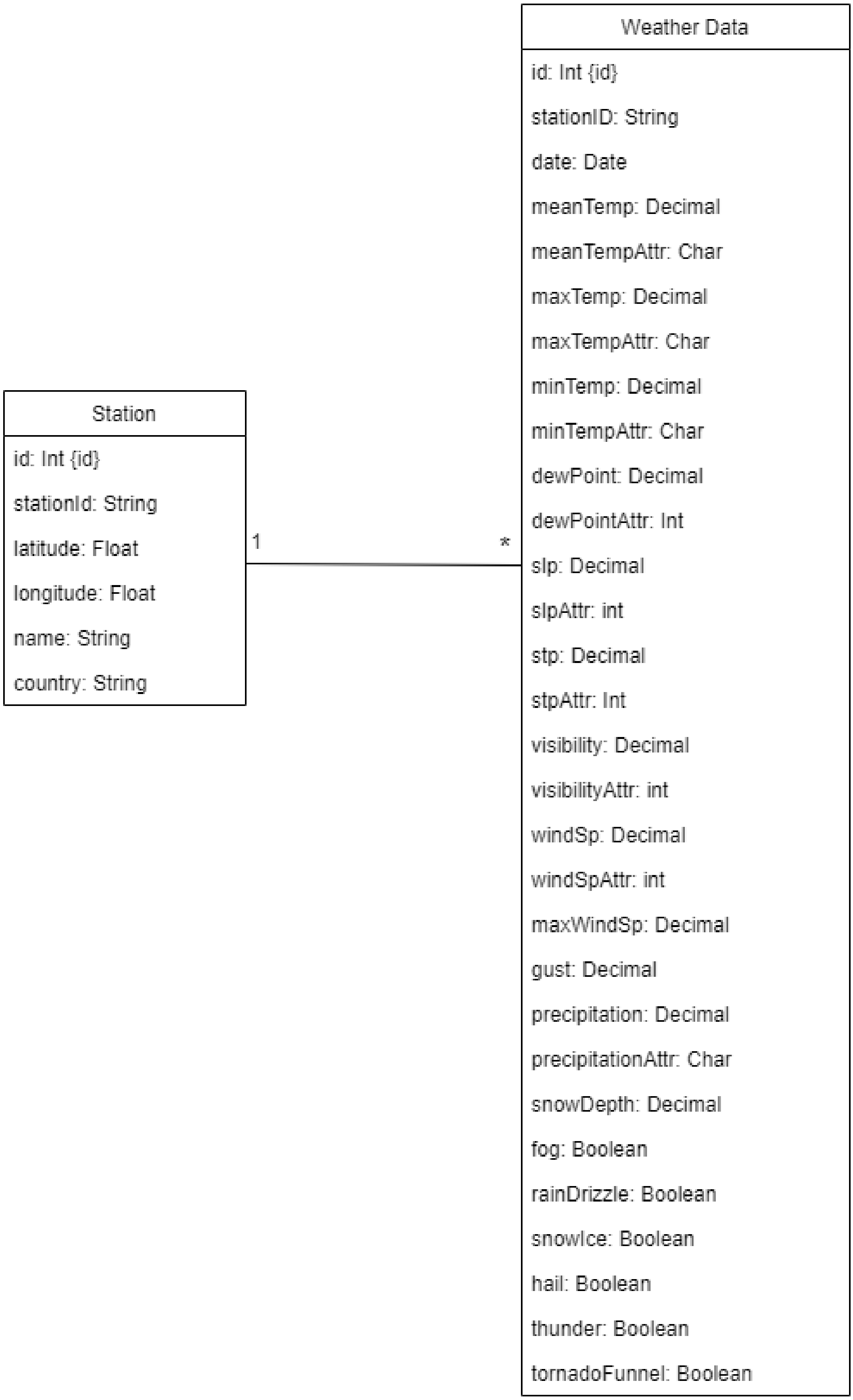


Figure 4.2: UML class diagram

Chapter 5

Implementation

In this chapter, the detailing of the implementation and the considerations that influenced it are described. A summary is presented at the end.

5.1 Initial Process

The work begins with acquiring all the available GSOD data from the NOAA's database. To do so, this GNU wget command was run on a WSL console:

```
$ wget -r -np -R "index.html*" https://www.ncei.noaa.gov/data/global-summary-of-the
    -day/archive/
```

Having all the information available on the local storage, a parser was implemented to read the CSV data. Using the pandas package to read the files and a MySQL connector to transfer the data to the database. A criteria was implemented to filter out stations which had no information regarding their location and elevation, as stations without those values were useless to the project. Only information from valid stations was stored. The process of populating the database took over eight hours.

With the data stored and ready to be accessed, the development of the website starts. Using Flask to host it through Python allows the data to be sent to JS and displayed to the user. In JS, using the Leaflet helper, the map is constructed and displayed using the web browser Google Chrome. The tile information used came from OpenStreetMap and Esri, they provide accurate display of roads, borders and locations and also satellite images.

With this initial setup, the first transfer of information starts. It begins with something simple: transferring the station's location and name and displaying them using markers on the map. In this step, the first challenge of implementation starts. By displaying all 12023 stations at the same time, the website became very slow and cluttered. This can be attributed to the following scenarios: the used hardware is not powerful enough, Google Chrome has information limitations or Leaflet is not prepared to handle this amount of markers on the map. Following this set-back a compromise had to be made. The information displayed would be restricted to the Iberian Peninsula, this change would reduce the number of displayed stations, from the initial value, to 156. To truncate the data, the parser would have to determine which stations were on the new area. After a short research, it was noted that both Portuguese and Spanish stations start with "08" in their IDs, allowing for a simple way to filter stations. The database was drooped and built again, taking around one hour to do so. To test the performance of this plan the same test was conducted. All the stations on the Iberian Peninsula were displayed without any input delay or lag, proving the solution worked.

The next stage was to display information related to station's records. To start simple, each marker would contain the temperature registered during selected day chosen by the user. The marker would also be colored to better display the temperature information. As such, a color scale was developed to encapsulate all the more common temperatures in the Iberian Peninsula. To find the temperature trends a research was conducted leading to this article [Ninyerola et al., 2007], which shows the temperatures oscillate between 36ºC and -12ºC. A scale was built using this information and the result is demonstrated in Figure 5.1.

5.2 Historical Values

This section of implementation is regarding the development of a tool to allow the user to control the date of the displayed data. Possible dates go back to 1929, but during testing, it was observed that very few stations existed in the older years. Since only roughly five valid stations operated during those years, the amount of data was extremely reduced. It was observed that in the year 1973 a "boom" in the number of valid stations happened. Because of this, the time period was limited from 1929-present to 1973-present. This way, only the relevant data would be displayed and used to interpolate and create the temperature grid to "fill in the blanks".

To allow the user to responsively change the information date a time slider was implemented. It was done using an external library, available on this github [dwilhelm89, 2017]. This JS file allows the creating of a time dimension layer inside the leaflet map object, which uses slider divider to control it. With this time slider, arises a new challenge: the amount of information loaded onto the website.

The first naive approach, was to give all the available information on the database, to the website. This was inconceivable has it crashed before it could even display the map. The theory behind this crash is that the amount of information was too big for JS to store and display. To solve this issue a query was developed each time a new day is selected. However, this solution was not optimal, has querying the database every time the slider is moved made the website slow and unresponsive. The next proposed solution was to create a database scope for the website. What this means is, only a select portion of the information is provided to the frontend. This portion is controlled by the time period, which the user controls. The first time scope encapsulated the information from each year. With this JS would have access to all the information regarding the year the user currently selected. This proved to be an improvement, but was not yet optimal has querying the database would take about twenty seconds. To enhance this solution the scope was reduced to weeks. This solution made the queries very fast, but unfortunately increased their amount substantially, making the website unresponsive. Using these two methods the third and final solution was developed. By increasing the scope to all days in a given month, the number of queries was shortened and the time they took to process was acceptable, clocking in at two seconds to complete.

During this time, the displayed information was only relative to the mean temperature recorded by the stations. But it was requested that more information was displayed to the user. To do this the queries to the database where changed to include the missing data. A UI drop-down selector was also implemented containing the following weather information: maximum and minimum temperature, sea level pressure, wind speed, precipitation and relative humidity. This last one was not available directly in the GSOD data, but by doing a quick research a formula was found that allowed for its calculation [Alduchov and Eskridge, 1996].

$$RH = 100 * \frac{e^{\frac{17.625*DP}{243.04+DP}}}{e^{\frac{17.625*MT}{243.04+MT}}} \tag{5.1}$$

RH = Relative Humidity (%)

DP = Dew Point (º)

MT = Mean Temperature (º)

5.3 Temperature interpolation

This is the final step of the implementation, here the algorithm to interpolate the temperature from the stations to form an information grid, and then using the Earth's topography to add precision to the calculations, was implemented. The process started with acquiring the terrain's characteristics. As mentioned on the technological choices section, the chosen model was GTOPO30. This model is available in the EarthExplorer tool, made by the Earth Resources Observation and Science (EROS) center. This tool allows for the selection of the area desired instead of acquiring the global data. Using this feature, the area was limited to the Iberian Peninsula, it was then stored in a 1548 by 950 pixel JPG image, showed in Figure 5.2.

In this image, the brighter a pixel is, the higher the location corresponding to that pixel coordinates is in the real world, and vice versa. The next step was to transform the information on the image into usable data for the project. To do this, the image was loaded into memory using the image manipulation library OpenCV. It makes the process possible by loading every pixel's Red, Green and Blue (RGB) values into a matrix with the number of columns equal to the image height and rows to the width. Since the image is greyscale, the Hue Saturation Value (HSV) color values are preferred to access pixel data, in comparison to the RGB ones. Thankfully, OpenCV has a quick tool to convert image color formats. After doing this, a regression function was found to associate pixel values to real world elevations. For this step, the stations elevations were used in conjunction with the respective pixel value in the image. By calculating their locations on the image using their coordinates, and then comparing the pixel's value with the station's altitude, a set of points were created. These points were then stored in a Microsoft Excel spreadsheet and a regression analysis was conducted to find the function. The result function can be seen on Figure 5.3.

The calculated linear regression provided the function $y = 13.735x - 32.942$, where the y variable represents the elevation in that pixel and the x variable the value of that pixel.

With the topography part ready, the next step was to interpolate station data in order to form a image, to then display on top of the map. The first proposed solution was to create an empty image, using the Python library NumPy. By using the function:

```
numpy.empty(shape, dtype, order, *, like)
```

an empty matrix can be created. A more detailed explanation of the function can be found here: NumPyDoc.

The shape parameter inserted was a tuple containing the image height = 950, image width = 1548 and the number of color channels = 4, this value represents the three RGB channels and the additional alpha channel, which stores the transparency. The other parameter was the dtype = numpy.uint8, which forces the values to be unsigned integers. The remaining parameters where left to their default values.

Having an empty canvas, the next step was to fill it with stations records. But, before proceeding to this step, since the elevation of the terrain is taken into consideration in the final calculation, the temperature values need to be set to sea level. To do this, the adiabatic lapse rate, which is the rate the temperature of an air parcel changes in response to the compression or expansion associated with elevation change, must be found. A search was conducted leading to this book [Hemond and Fechner, 2015], which shows that the variations are different for each layer of the atmosphere. These differences can be seen on Figure 5.4. Since the highest point in the Iberian Peninsula is the Mulhacén mountain in Spain towering at 3.4786 km, all the used data is located in the Troposphere.

The dry adiabatic lapse rate is a constant 9.8 °C per km, but the moist adiabatic lapse rate lies in the range 3.6-5.5 °C per km. Due to not having enough information to determine if the atmosphere, in the given location at a specific time, is wet or dry, the environmental lapse rate (ELR) value must be used. The ELR is the rate of decrease of temperature with the altitude in the stationary atmosphere, at a given time and location. It's variation can be observed in this Figure 5.5. It can observed that the values decrease in a linear way. This allows for more specific measurements of altitude, such as meters. The used ERL value on this project was 0.0064 ºC per meter.

In this stage, the generated image was empty except for the pixels correspondent to station records of temperature, at the sea level. It is worth mentioning that, if multiple stations were positioned in the same pixel, a weighted average would be performed. The next challenge was to fill the rest of the image using the existing data.

The initial approach used OpenCV's image manipulation tools. The dilation function was used in conjunction with Gaussian Blur function. Dilation enlarged the existing station's pixels and the blur smoothed the results so that the image didn't appear pixelated. Both solutions used a five by five matrix containing ones to operate on the image. This matrix is known as an image kernel and controls the weight each pixel value has on the calculation. This solution was extremely fast, but displayed unfavorable results due to pixels with higher color values replacing lower ones in the dilation process, losing precision. Taking this into consideration it wasn't further implemented.

Another more complex approach consisted of going pixel-by-pixel and averaging the two closest stations temperature values, taking into account their distance. This method is know as inverse-distance weighting [Lu and Wong, 2008].

A linear interpolation of this method was thought of at first: consider two stations A and B, and the distances from a certain location to these stations a and b, respectively. To calculate the coefficient for the temperature values of station B, one would use the expression

$$\frac{a}{a+b}$$

seeing that when the distance to station A increases, the more weight should be given to the temperature of station B. Therefore, a linear temperature component interpolation would look

$$temp_L = \frac{b}{a+b} * temp_A + \frac{a}{a+b} * temp_B$$

for a given location L. For a case where the location is collinear with the two stations and the distance between the stations A and B is 10 pixels, meaning $a+b=10$, and for temperature values of $temp_A = 30$ and $temp_B = 5$, the interpolation results can be seen in Figure 5.6a

However, a linear interpolation would cause two problems. The first is that to calculate these distances, it needs the square roots of the squares of the distance differences. The square root is a computationally intensive operation when compared to other basic algebraic operations, and since it was performing those operations for every pixel in the image minus the ones occupied by the stations themselves, results took a long time to produce. The second is that when there are vastly different temperature values for the two closest stations, just a little distance from any station produces a fairly unnatural difference where the other station's temperature value is taken too much into account. When near to a station, that station's temperature value should have a stronger weight than the second closest station's.

To prevent this, the coefficient for the temperature values of station B can be calculated using this

$$\frac{a^2}{a^2+b^2}$$

removing the need for square root operations. For this quadratic-based interpolation, a temperature component is calculated using

$$temp_L = \frac{b^2}{a^2+b^2} * temp_A + \frac{a^2}{a^2+b^2} * temp_B$$

for a given location L. The variation on the temperature values using the same example as before can be seen in Figure 5.6b. The main difference from the linear interpolation is that when near an airport, the values decrease and increase in a smoother way. This is because even though the distance between stations is constant, the denominator on the coefficient now varies and is higher when closer to an station than when in the middle, whereas in the linear interpolation the denominator would be constant.

These two methods were proven obsolete as the following option displayed more accurate results with way faster computation times.

The aforementioned option was to use SciPy interpolation library. This library allows the generation of missing data in multidimensional grids. Since the project intends to create a image from a matrix containing temperature data, the

`scipy.interpolate.griddata(points, values, xi, method, fill_value, rescale)`

funtion was used. A detailed explanation of how this funtion operates is described here SciPy Doc.

The arguments chosen to call this functions are as follow: for the points and values, a python dictionary was previously created using keys as pixels coordinates and values as temperatures, for the xi, which corresponds to the format of the output, a matrix with the size of the image (1548x980) and finally the three available methods were used.

The nearest method functions similarly to a nearest neighbor algorithm. It uses an auxiliary kd-tree, which is a binary tree with each node being dimension k, for a quick nearest-neighbor lookup [Maneewongvatana and Mount, 2002].

The linear method uses linear barycentric interpolation [Hormann, 2014], which is basically a linear interpolation using barycentric coordinates. These coordinates represent the location of a point in reference to a simplex (same concept as a hyperrectangle, but for a triangle).

The cubic method constructs a piecewise cubic interpolating Bezier polynomial on each triangle, using a Clough-Tocher scheme [Alfeld, 1984]. The gradients necessary for this are estimated using the global algorithm described in [Nielson, 1983] and [RENKA et al., 1984].

The output of this function is then converted into an image using the topography, with the obtained ELR funtion, as a mask. Images are generated for every day in the month selected by the user and then displayed on top of the map for their respective day.

The spanned results from this implementation will be discussed in more detail in the next chapter.

5.4 Summary

The development of this project began with displaying station information to the user on the map. This lead to the challenge of controlling the amount of information available to the website session. After deciding that the information available is best provided in months, the interpolation process began. This process consists on converting the registered temperature to sea level values, interpolating the points to create an information grid and finally use the topography to find the values correspondent to the Earth's altitude. Three interpolation solutions where implemented but one showed much more promising results than the others. This solution has three methods of interpolation and their results will be discussed in the following chapter.

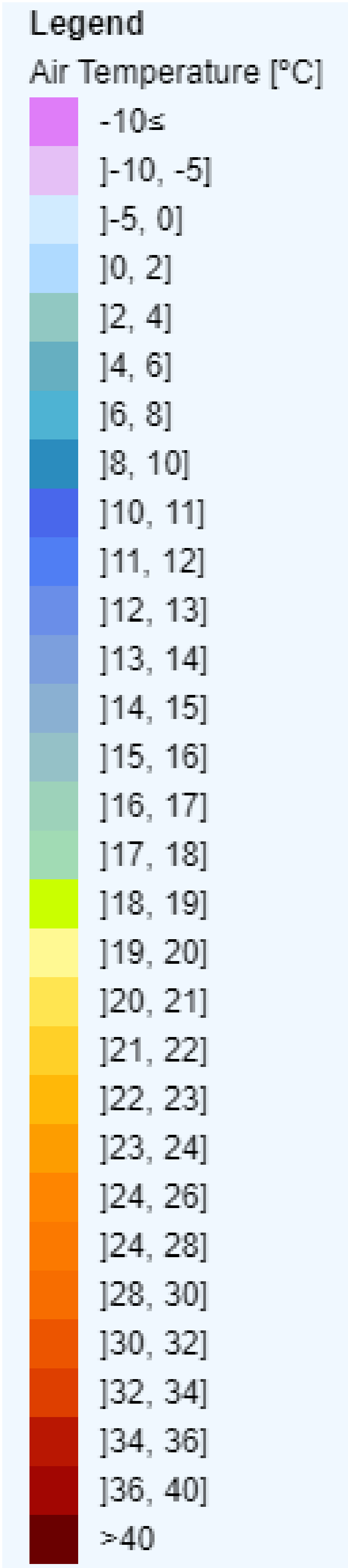


Figure 5.1: Temperature Color Scale

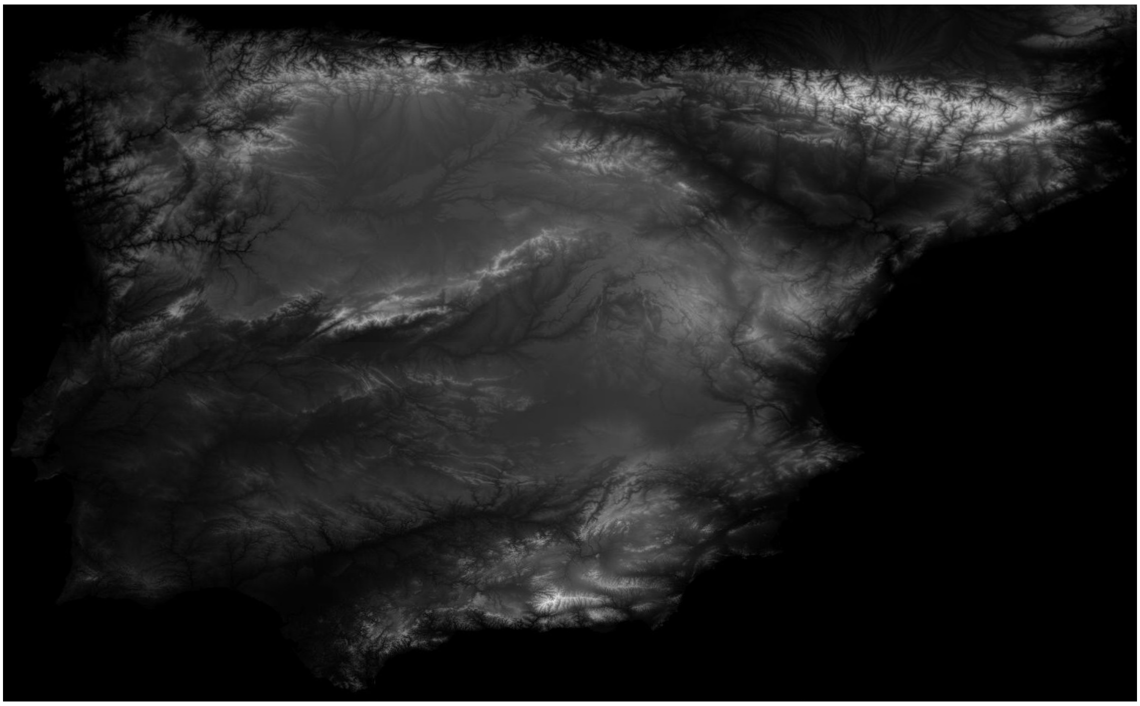


Figure 5.2: GTOPO30 Iberian Peninsula Topography

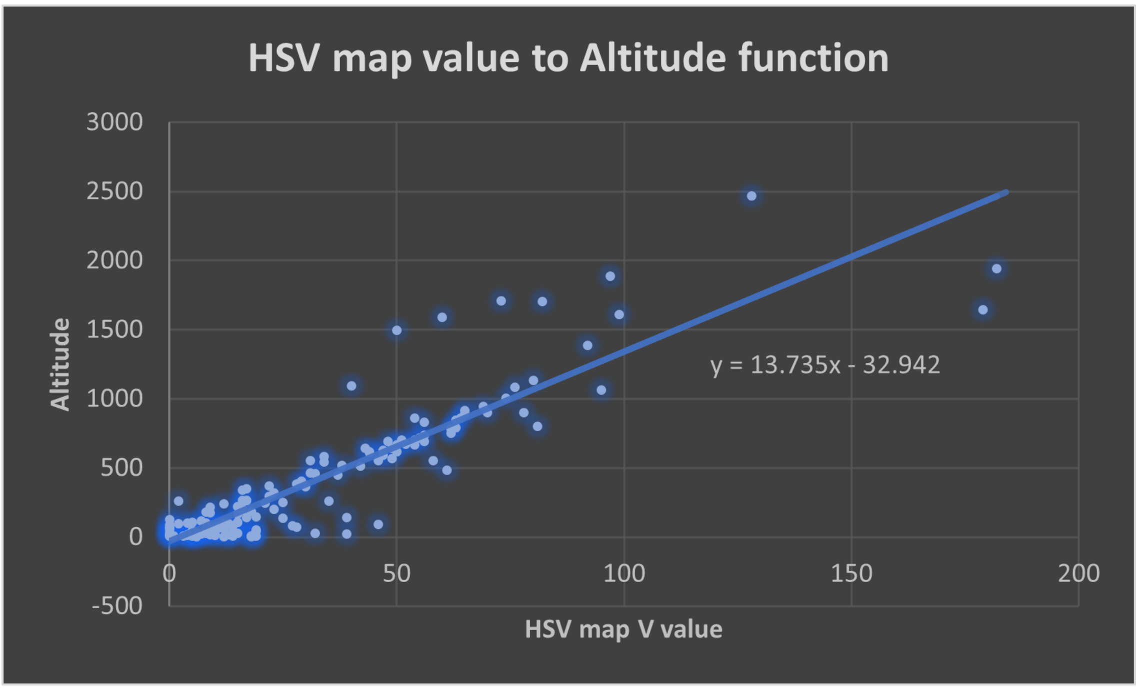


Figure 5.3: HSV map value to Altitude Function

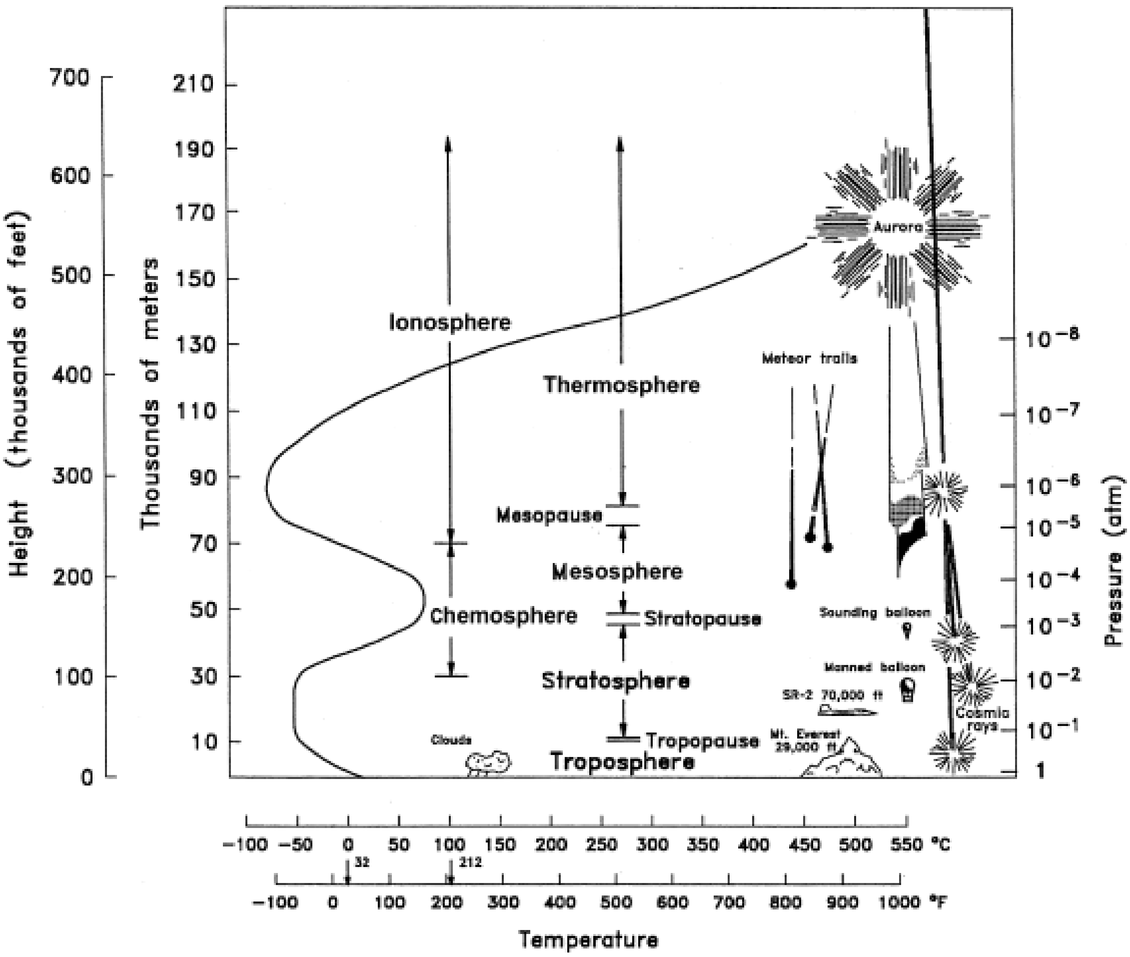


Figure 5.4: Temperature relation with the vertical structure of the atmosphere [Hemond and Fechner, 2015]

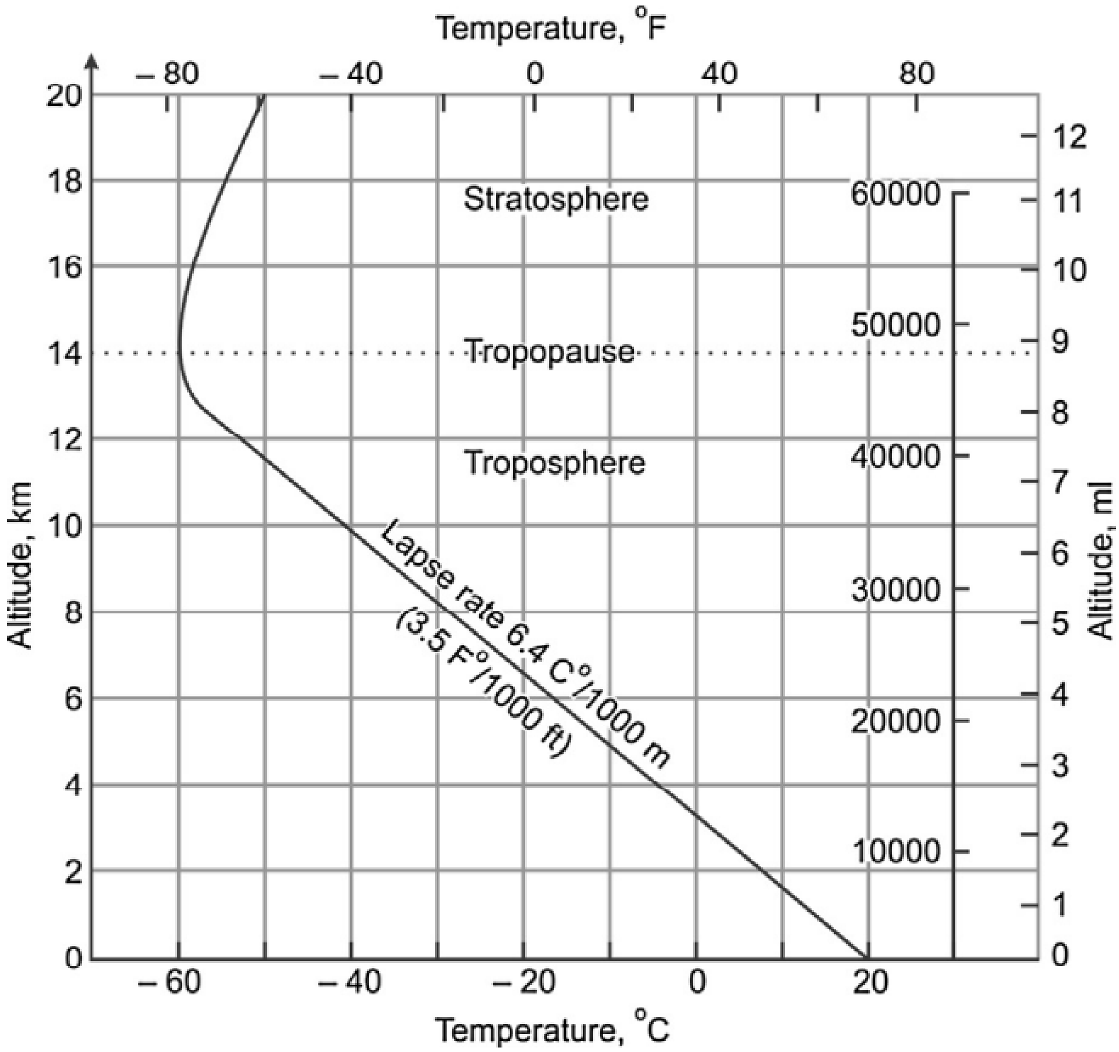
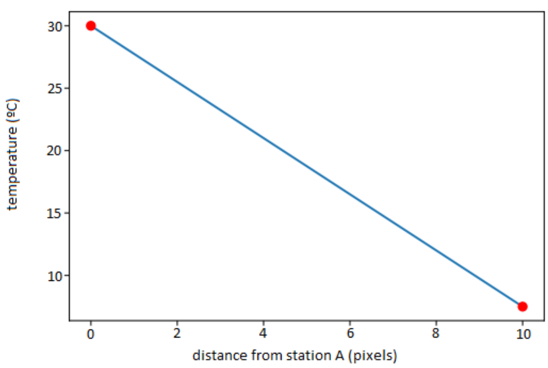
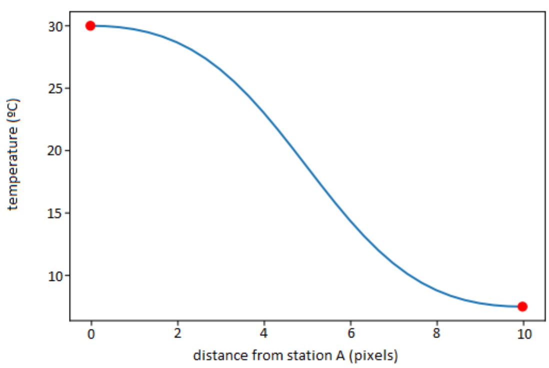


Figure 5.5: Lapse Rate [Muralikrishna and Manickam, 2017]



(a) Linear



(b) Quadratic

Figure 5.6: Temperature Interpolation Examples

Chapter 6

Project Results

In this section, the obtained experimental results will be displayed and analyzed. Following with a display of the final solution and a guide on it's uses. Ending with a small summary to consolidate the discussed ideas.

6.1 Experimented Interpolation Methods

The methods provided by SciPy will be evaluated in this section. The test to verify how accurate the results are is: after applying the topography mask, the color of the pixels corresponding the stations locations, has to have the same color on the neighboring pixels. The amount of time each solution takes to create one image has been taken into account, this is important because the images are going to generated "on the fly". A comparison has been made between the displayed results and other weather applications, to see how innovative this solution is and if it represents the values correctly. All the data used corresponds to the 1st of June 2021.

6.1.1 Nearest Method

Nearest method takes 2 seconds to generate one image (ex: Figure 6.1), the results are not very promising as there are very sudden changes in the temperatures giving it a "mosaic" look, this result is expected as the algorithm only takes into account its neighborhood, resulting into agglomerations of values. By applying the the topography shader, seen on Figure 6.2, an outline of the Peninsula can be noted and the highest places are observable due to the temperature decrease. Still the "mosaic" effect is still visible, making this solution not very valuable.

6.1.2 Linear Method

This method is a very notable improvement compared to the nearest one. It takes 5 seconds to generate an image (seen on Figure 6.3). The result looks promising, both have similar color areas,

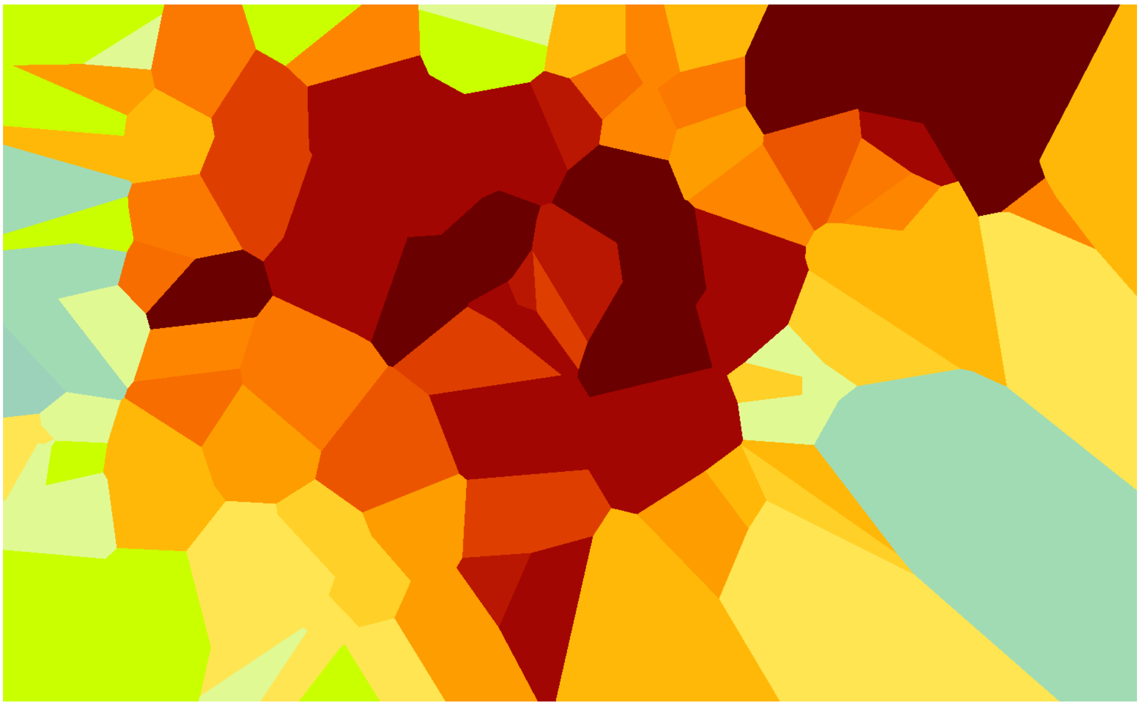


Figure 6.1: Nearest Interpolation Result

but linear has more gradual changes throughout the area. One of the problems is that, due to how it calculates information inwards, values beyond the outside stations are not calculated. When applying the topography values (Figure 6.4) the results are much more clear. It doesn't have the previous agglomerates problem, making the transition between areas much smoother. Despite this, in some areas there is some noticeable pixilation, this is due to the algorithm working with linear functions, which create those visible straight lines.

6.1.3 Cubic Method

The solve the previous issue, the cubic method creates smooth curvatures at the cost of more computational power. This however, is barely noticeable, as the generation of an image takes only 1 second more than linear (result on Figure 6.5). The color areas are similar to the other methods, but there is also some specific color that are only visible using this solution. This can be due to the algorithm working with quadratic functions, solving the problem linear interpolation previously mentioned. With the topography calculation (Figure 6.6), the result is similar to linear, as expected, but there is no noticeable pixilation, making it an improvement overall and the preferred solution.

To validate this proposal the color of the station's marker should have a resemblance to its surroundings, this however is not the case in some stations. An explanation for this can be attributed to the topography linear function generated (Figure 5.3), it can noted that in some points deviate greatly from the function. Another explanation can be that the elevation of some stations do not correspond to the altitude of the terrain in those coordinates, this however, seems unlikely.

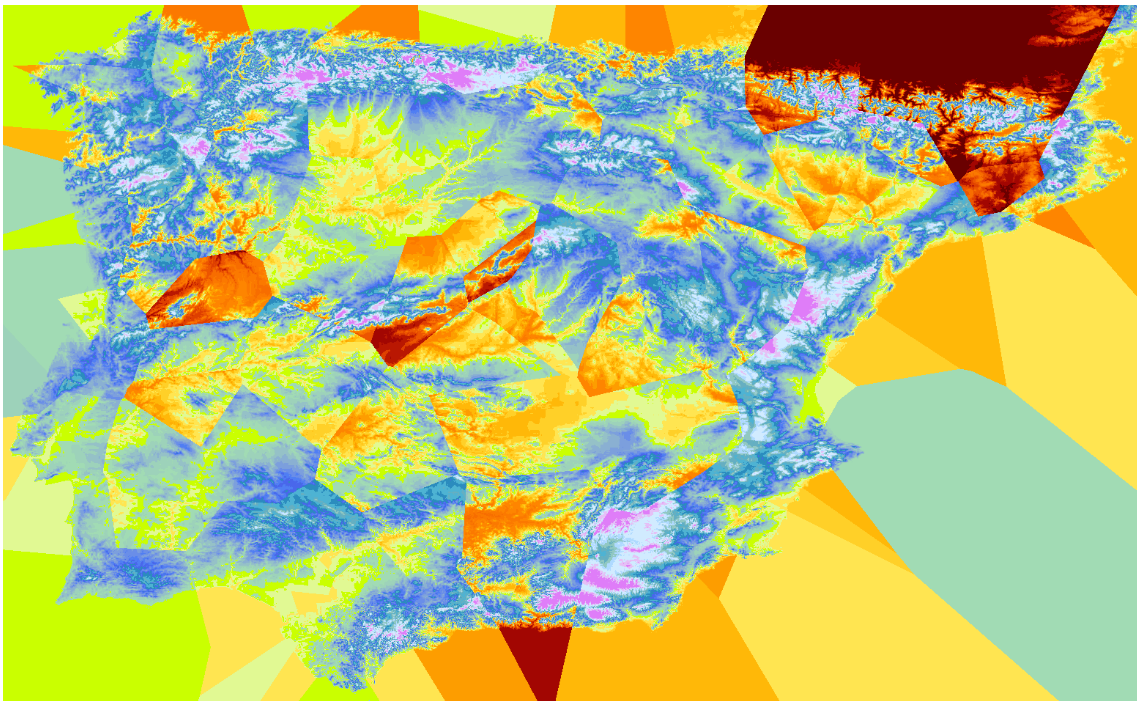


Figure 6.2: nearest Interpolation with Topography Result

Comparing the results with fogos.pt temperature layer (Figure 6.7), this solution seems to have much more detail and clearer values. The same can be said for Dark Sky (Figure 6.8), Windy (Figure 6.9) and OpenWeatherMap (Figure 6.10). It can be extrapolated from these images that these websites give very little or no importance the terrain's altitude influence.

The one website which seems to have similar data compared to this project is IPMA (Figure 6.11). Comparing the results it can be noted that this project gives a lot of weight to the altitude temperature differences, possibly creating false data. This can be attributed to the use of the lapse rate, which doesn't take into account the moisture of the atmosphere. Another aspect to note is that IMPA has access to private paid stations, likely providing more accurate data and information regarding the moisture of the atmosphere. This allows them to calculate the adiabatic lapse rate which is more reliable than the average lapse rate.

6.2 Final Result

As mentioned before the cubic interpolation was used in the final version of the project, but due taking some seconds to complete (as well as the other methods), and since the images are generated when new data is requested by the user, the temperature grid function is not enabled, as it made the website unresponsive. This problem can have multiple solutions, the first one is lowering the image resolution so that less calculations are performed. Another possible solution is to improve the hardware running the program, by having a more powerful CPU or a Cuda GPU, interpolation calculations would very likely be faster.

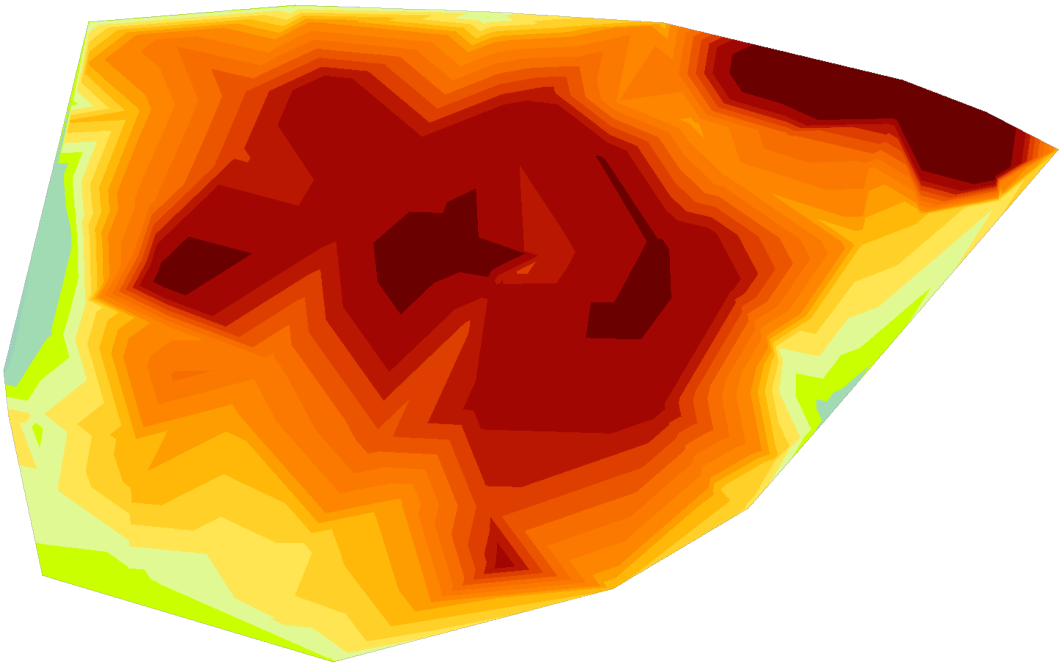


Figure 6.3: Linear Interpolation Result

The final version of this project implemented all the user requirements requested by the client, expect for the chart section where the station data would presented, this was due to time constraints. The website is only available if hosted locally, but it can easily be put online, hosting the database in a server (i.e. Amazon Web Service) and deploying the Flask application using Linux server (i.e. Linode).

The final appearance of this project can be seen on Figure 6.12.

The top left controls (Figure 6.13) allow the selection of the type of information displayed (Figure 6.14) and from which month it is displayed (Figure 6.15). The slider controls which day, of the chosen month, is displayed on the map.

Hovering over a marker shows all the information regarding that station (Figure 6.16), for the selected day.

Finally the top right contains the multiple usable map tiles, as well as the topography layer option (Figure 6.17).

6.3 Summary

This results of these work originated from three methods abstracted by the SciPy library, after analyzing them, the preferred method was cubic interpolation. It shows great potential, but also provides some unreliable information, which can be explained by the usage of sub-optimal formulas in the calculation of how the temperature varies with the altitude. By comparing the project's

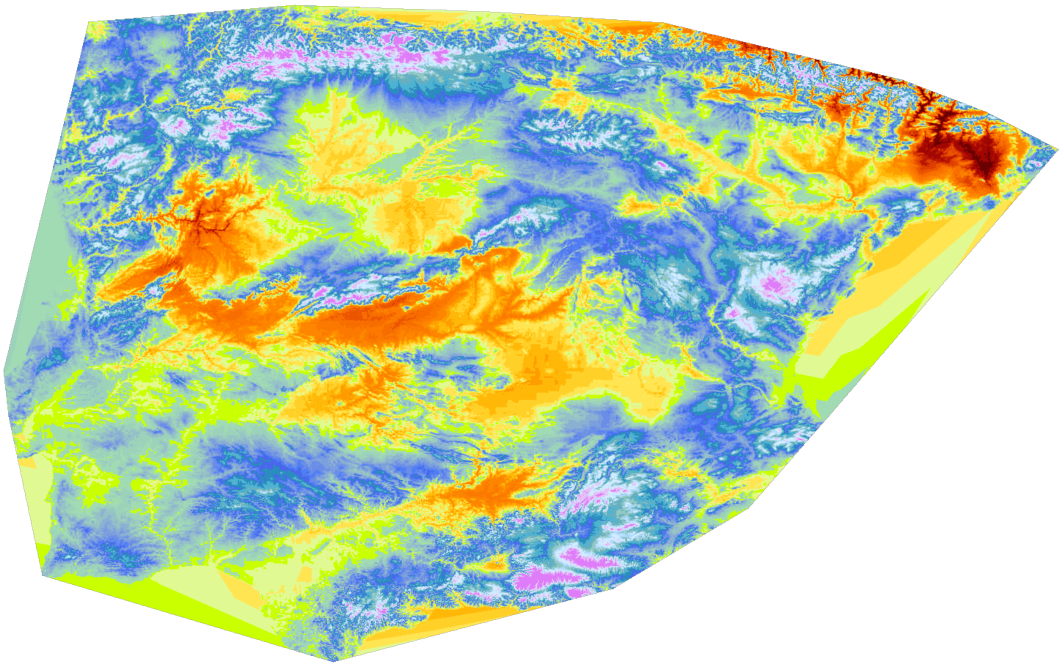


Figure 6.4: Linear Interpolation with Topography Result

result with other existing platforms it can be noted that it presents an innovative solution, which can be further developed on.

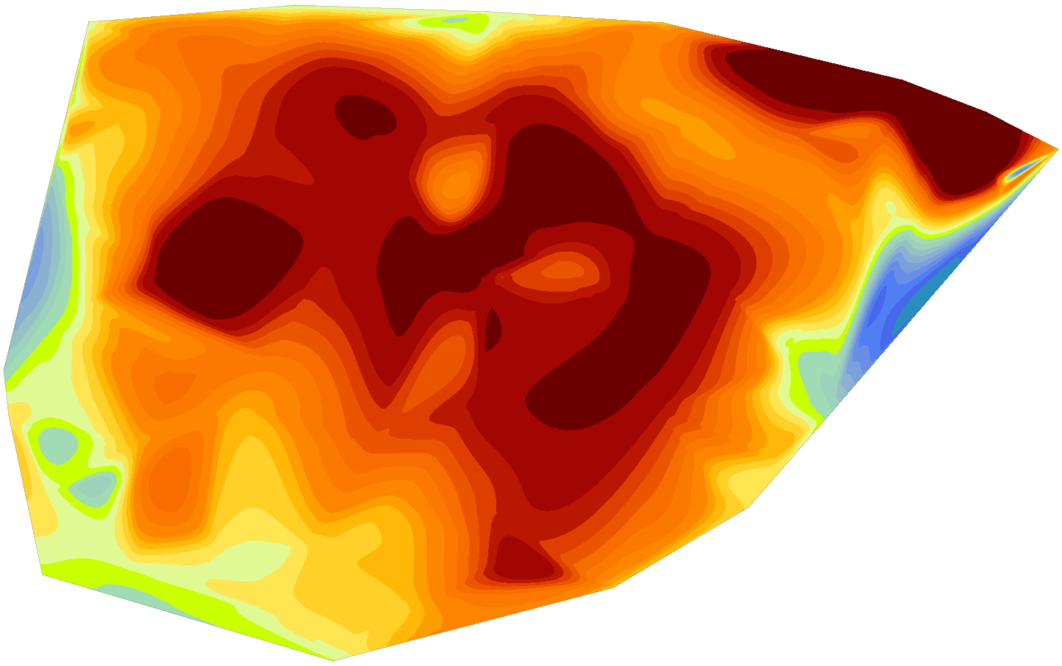


Figure 6.5: Cubic Interpolation Result

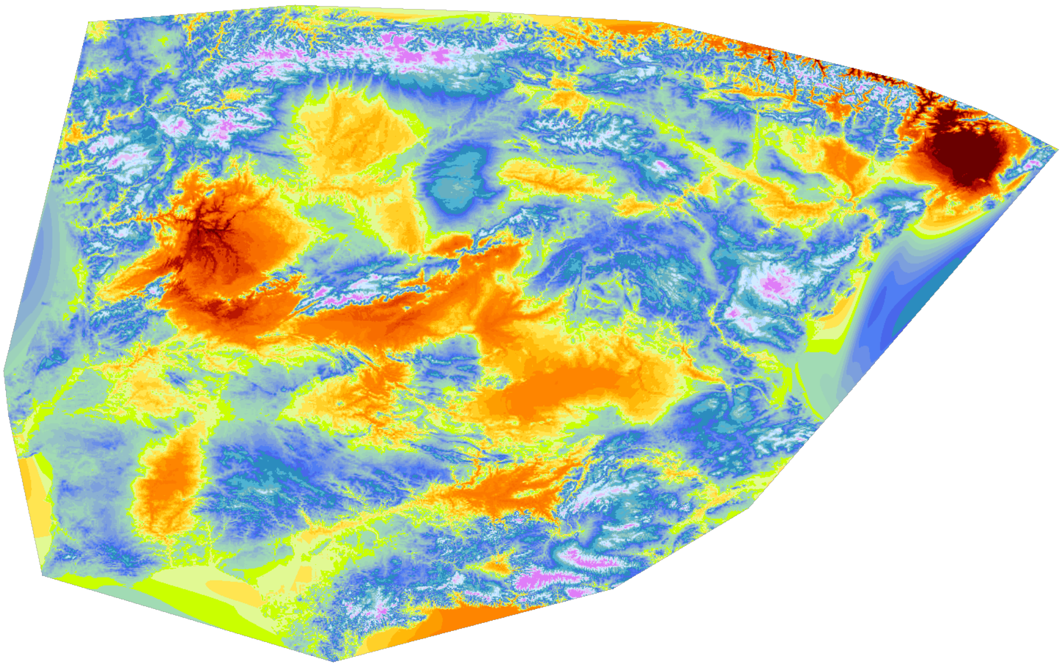


Figure 6.6: Cubic Interpolation with Topography Result

Figure 6.7: Fogos.pt temperature layer [Pina, 2021]

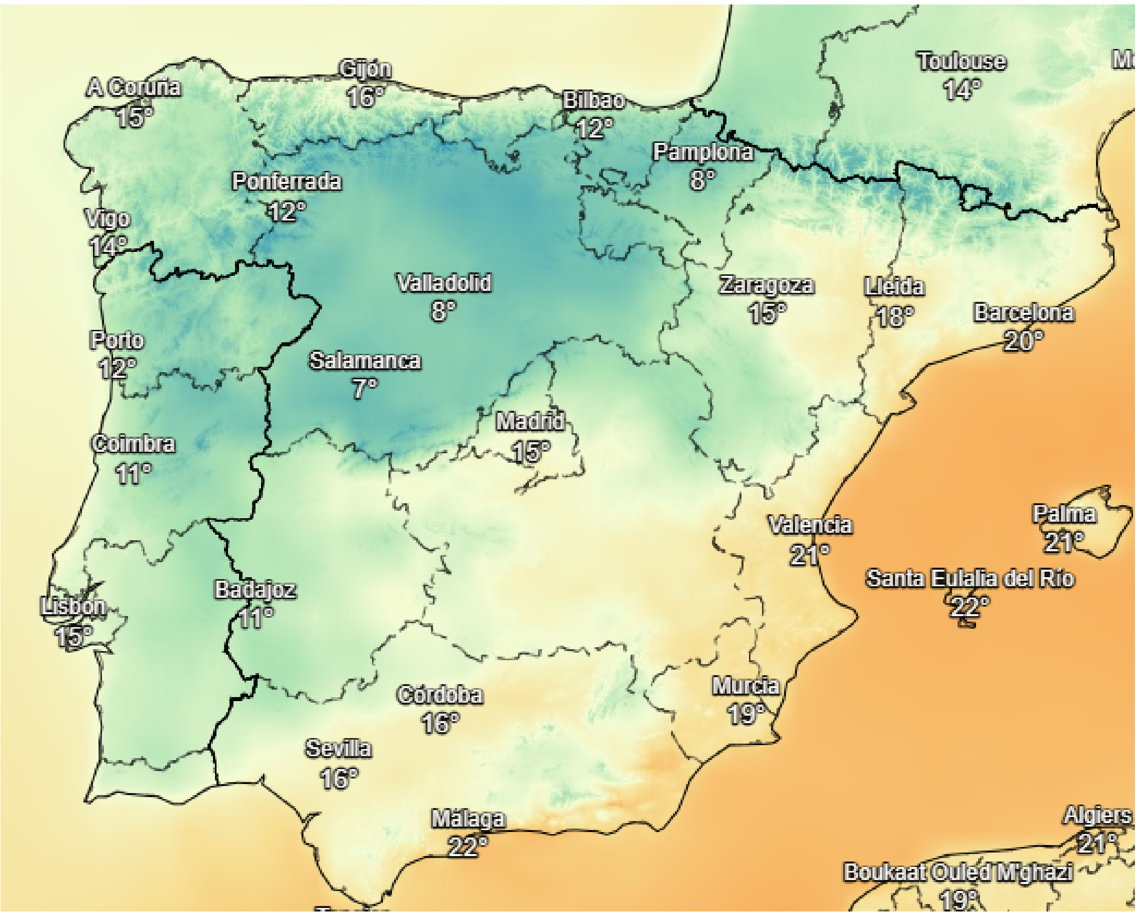


Figure 6.8: Dark Sky temperature layer [Apple, 2021]

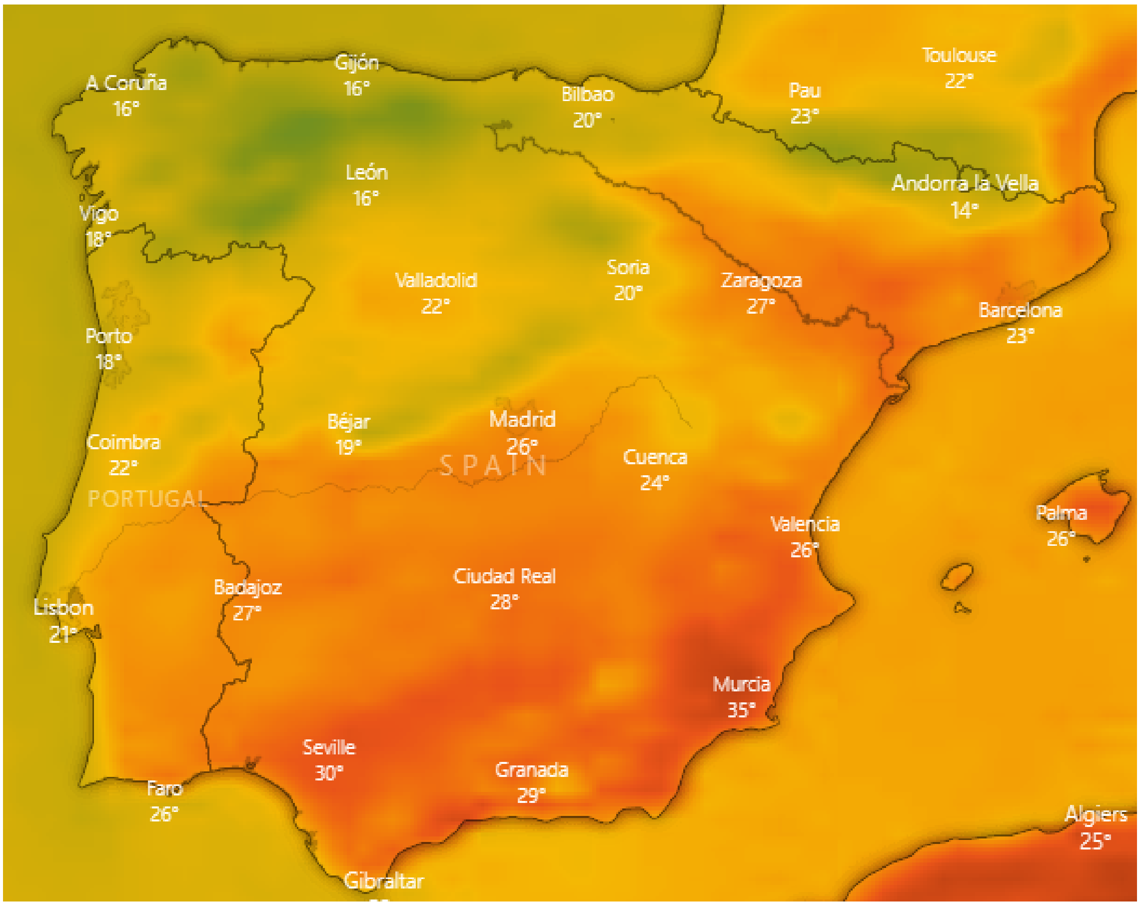


Figure 6.9: Windy temperature layer [Lukačovič, 2014]

Figure 6.10: OpenWeatherMap temperature layer [OpenWeather, 2021]

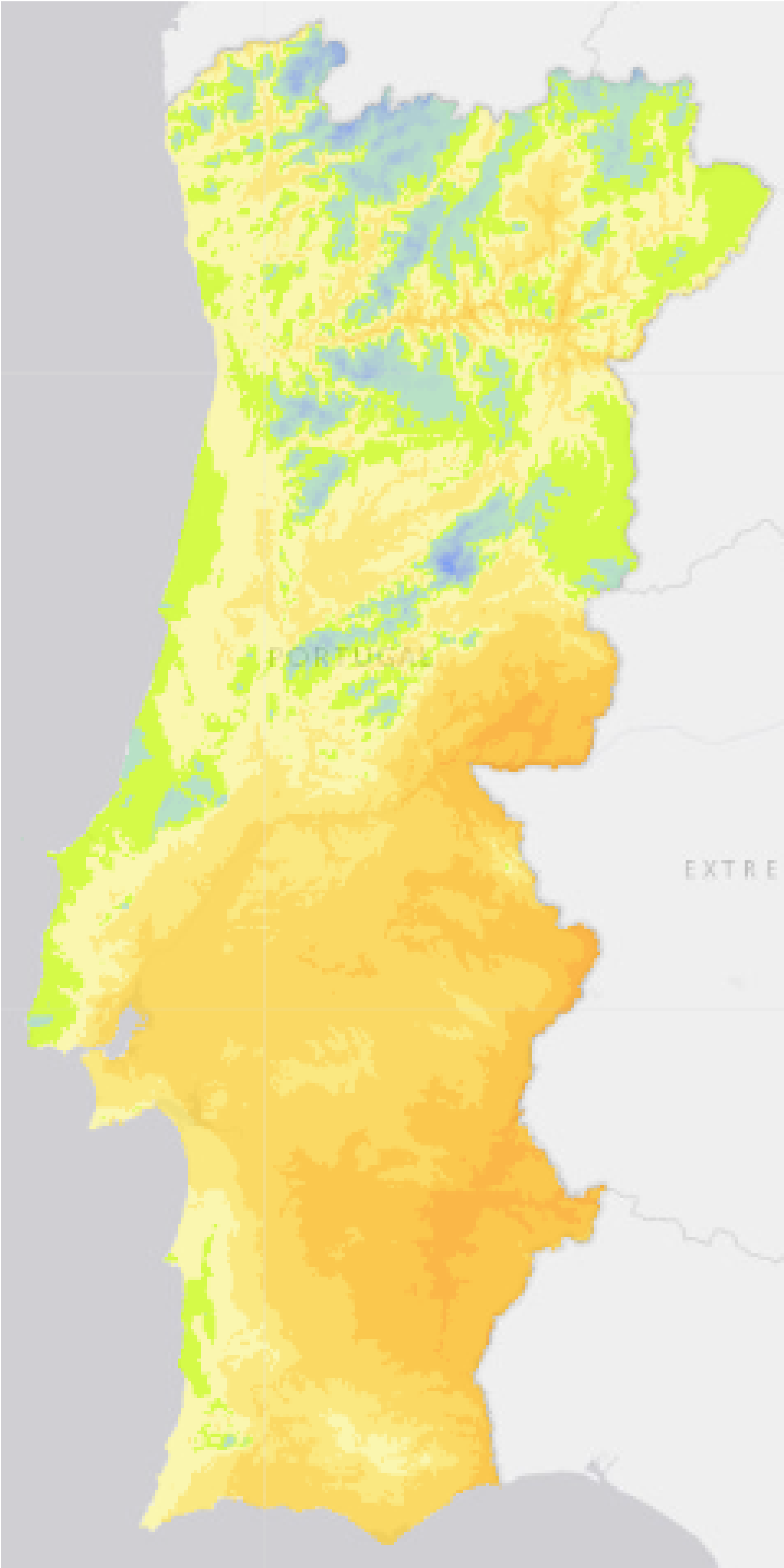


Figure 6.11: IMPA temperature layer [Government, 2021]

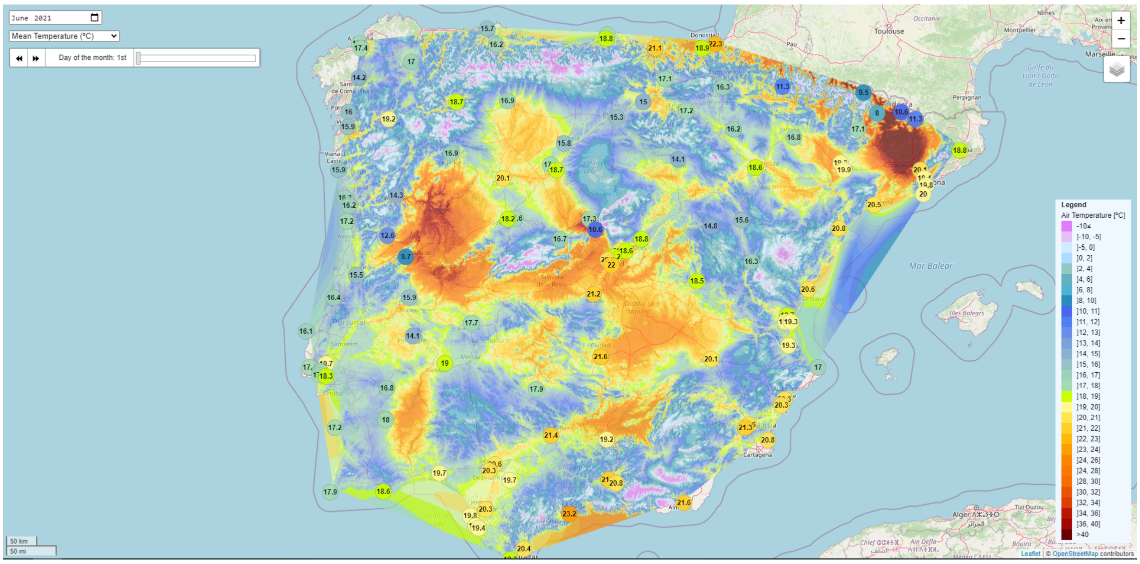


Figure 6.12: Final appearance of the website

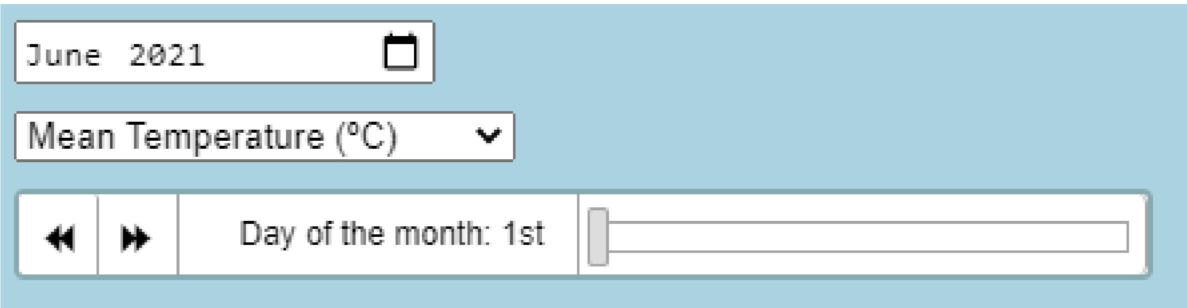


Figure 6.13: Time control and information selector

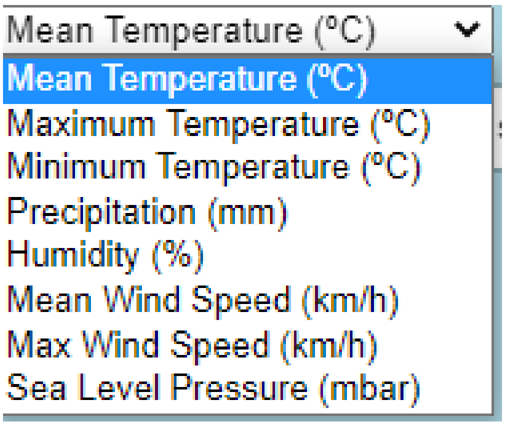


Figure 6.14: Types of information

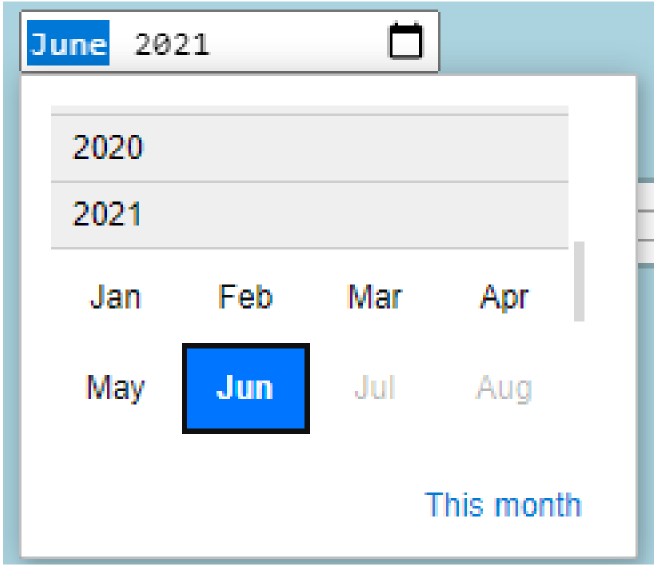


Figure 6.15: Month Selector

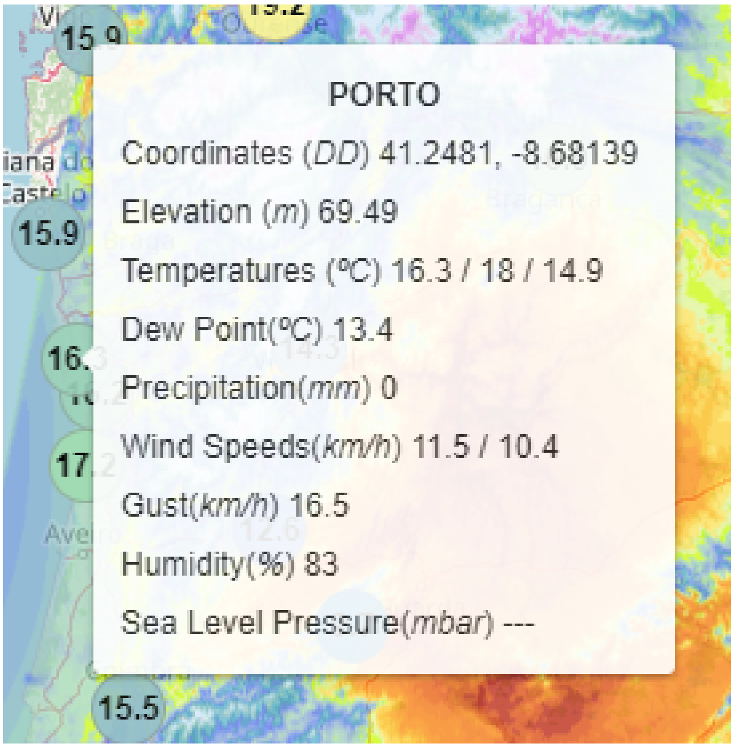


Figure 6.16: Station information menu

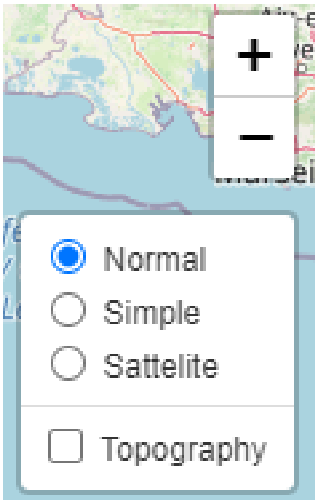


Figure 6.17: Layers Menu

Chapter 7

Conclusions and Future Work

This section will list a summary of the whole creation process, the conclusions that can be taken from the implemented solution, and future work that can improve upon that solution.

7.1 Conclusions

The numerous amount of existing weather stations don't offer a common API to consult their data. In order to agglomerate this data and present it in a user-friendly way, a website was proposed as a solution to this problem. This website would use records from weather stations and display them using a GIS tool to correctly map the information.

The literature review found that multiple weather applications already existed, which motivated this project to differentiate itself from the rest, by having available historical data and interpolating temperature values with the terrain's characteristics. It was also registered that multiple data sources were available and some GIS frameworks where free and simple to use.

The scope of the project was better defined to fit the hardware and software constraints and its architecture was laid out. The website would be composed of a backend were all the information was stored and a frontend represented by the website itself.

The implemented solution was tested and found to work. Several aspects of the performance of this solution were analyzed, and the optimal method for the solution was found. The solution is displayed as a web application that incorporates the user requirements gathered in the begging.

The solution presented an innovate approach to interpolating weather data and displayed it to a user. From the results, it can be concluded that generating grid weather data using the terrain elevation adds precision to the values. The solution used this project, however, has some impressions possible due to poor decisions on the use of the topographic data and the functions used to calculate the influence of the terrain over the temperature.

7.2 Future Work

After analyzing the project as a whole and reviewing the creation process, multiple changes can be performed to add more value to the proposed solution.

To further improve upon the developed work, the main aspect to take into consideration is the use of better formulas to calculate how the temperature varies in relation to the elevation. The use of different interpolation methods should also be considered and experimented on, in order to find a more suitable solution for this project. The temperature interpolation process can be extrapolated for other weather phenomena. An example being rainfall, which needs additional data, such as the slope of the terrain, to accurately interpolate and taking into account the height of the area.

Fulfilling the final user requirement of displaying the monthly data in a graph with its respecting section on the website, would add value to the solution as the user could observe the evolution of the weather, during that time period, in an informative way. It can also allow for better data readings and explanation of some values that differ from the expected norm.

The scale of the project can be enlarged to incorporate global data, this would require scaling the hardware available to increase computational power or changing the implemented solution to a more efficient one.

The use of a paid weather database or combining multiple sources of reliable data, can help to solve the lapse rate issue. Acquiring specific vector information such as: wind and precipitation directions can then be transformed into animated particles and displayed on the map.

Hosting the website, would present multiple opportunities, such as providing the user with location-based information, allowing the use of internet cookies to personalize the information displayed and possibly using the project has paid product. If putting the website online is desired, the UI/UX should be improved by using an icon library, adding more relevant weather information and/or enhancing the CSS to make the website more aesthetic and seamless to use.

Data mining and machine learning algorithms are also very powerful tools to use when calculating missing values, this applies to the calculation of the temperature grid and the compensation of missing data due to station failure.

Finally, weather forecast is a very desirable feature in weather services, that this project can accommodate, by already providing historic information, forecasting could easily be displayed when the respective methods are implemented.

References

[Air Force Combat Climatology Center, U.S. Air Force, U.S. Department of Defense et al., 2003] Air Force Combat Climatology Center, U.S. Air Force, U.S. Department of Defense, 557th Weather Wing, U.S. Air Force, U. S. Department of Defense, and National Climatic Data Center, NESDIS, NOAA, U.S. Department of Commerce (2003). U.s.a.f. datsav3 surface observations, 1901-continuing.

[Alduchov and Eskridge, 1996] Alduchov, O. A. and Eskridge, R. E. (1996). Improved magnus form approximation of saturation vapor pressure. *Journal of Applied Meteorology and Climatology*, 35(4):601–609.

[Alfeld, 1984] Alfeld, P. (1984). A trivariate clough—tocher scheme for tetrahedral data. *Computer Aided Geometric Design*, 1(2):169–181.

[Apple, 2021] Apple (2021). Dark sky. Apple. Available online at https://darksky.net/. Last seen in 2021-02-25.

[Baldridge et al., 2009] Baldridge, A. M., Hook, S., Grove, C., and Rivera, G. (2009). The aster spectral library version 2.0. *Remote Sensing of Environment*, 113(4):711–715.

[Bartholomew, 2012] Bartholomew, D. (2012). Mariadb vs. mysql. *Dostopano*, 7(10):2014.

[Beccario, 2016] Beccario, C. (2016). earth. https://github.com/cambecc/earth.

[Business, 2020] Business, Brian Fung, C. (2020). Apple acquires popular weather app dark sky.

[Caron, 2011] Caron, J. (2011). On the suitability of bufr and grib for archiving data. *AGU Fall Meeting Abstracts*, pages 1619–.

[Clarke, 1986] Clarke, K. C. (1986). Advances in geographic information systems. *Computers, Environment and Urban Systems*, 10(3-4):175–184.

[Committee et al., 1998] Committee, F. G. D. et al. (1998). Geospatial positioning accuracy standards, part 3: National standard for spatial data accuracy. *Subcommittee for Base Cartographic Data; Federal Geographic Data Committee: Reston, VA, USA*, page 25.

[Couclelis, 2003] Couclelis, H. (2003). The certainty of uncertainty: Gis and the limits of geographic knowledge. *Transactions in GIS*, 7(2):165–175.

[Danielson and Gesch, 2011] Danielson, J. J. and Gesch, D. B. (2011). *Global multi-resolution terrain elevation data 2010 (GMTED2010)*. Number 2011–1073 in Open-File Report. U.S. Department of the Interior.

[dwilhelm89, 2017] dwilhelm89 (2017). Leafletslider. https://github.com/dwilhelm89/LeafletSlider.

[Fee, 2008] Fee, J. (2008). Do you still use arcview 3. x? *Spatially Adjusted. Archived from the original on*, 23.

[French, 2016] French, J. (2016). The time traveller's capm. *Investment Analysts Journal*, 46:1–16.

[Glickman, 2000] Glickman, T. (2000). *Glossary of meteorology*. American Meteorological Society.

[Government, 2021] Government, P. (2021). Instituto português do mar e da atmosfera. Available online at https://www.ipma.pt/pt/index.html. Last seen in 2021-02-25.

[Hand, 2016] Hand, E. (2016). Obsolescence looms for balloon data. *Science*, 352(6283):281–282.

[Hemond and Fechner, 2015] Hemond, H. F. and Fechner, E. J. (2015). Chapter 4 - the atmosphere. In Hemond, H. F. and Fechner, E. J., editors, *Chemical Fate and Transport in the Environment (Third Edition)*, pages 311–454. Academic Press, Boston, third edition edition.

[Hirt and Rexer, 2015] Hirt, C. and Rexer, M. (2015). Earth2014: 1 arc-min shape, topography, bedrock and ice-sheet models–available as gridded data and degree-10,800 spherical harmonics. *International Journal of Applied Earth Observation and Geoinformation*, 39:103–112.

[Holloway, 2016] Holloway, D. (2016). Ibm finalizes deal for weather channel product and tech business.

[Hormann, 2014] Hormann, K. (2014). Barycentric interpolation. In Fasshauer, G. E. and Schumaker, L. L., editors, *Approximation Theory XIV: San Antonio 2013*, pages 197–218, Cham. Springer International Publishing.

[Kelso, 2018] Kelso, V. N. (2018). natural-earth-vector. https://github.com/nvkelso/natural-earth-vector.

[Le Treut et al., 2007] Le Treut, H., Somerville, R., Cubasch, U., Ding, Y., Mauritzen, C., Mokssit, A., Peterson, T., and Prather, M. (2007). Historical overview of climate change science. chapter 1.

[Lu and Wong, 2008] Lu, G. Y. and Wong, D. W. (2008). An adaptive inverse-distance weighting spatial interpolation technique. *Computers & Geosciences*, 34(9):1044–1055.

[Lukačovič, 2014] Lukačovič, I. (2014). Home @ windy community. Available online at https://community.windy.com//. Last seen in 2021-02-25.

[Maliene et al., 2011] Maliene, V., Grigonis, V., Palevičius, V., and Griffiths, S. (2011). Geographic information system: Old principles with new capabilities. *URBAN DESIGN International*, 16(1):1–6.

[Maneewongvatana and Mount, 2002] Maneewongvatana, S. and Mount, D. M. (2002). Analysis of approximate nearest neighbor searching with clustered point sets. *Data Structures, Near Neighbor Searches, and Methodology*, 59:105–123.

[Merkel, 2014] Merkel, D. (2014). Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2.

[Merriam-Webster, 2003] Merriam-Webster (2003). *Merriam-Webster's Collegiate Dictionary*. Merriam-Webster, 11th edition.

[Miliaresis and Argialas, 1999] Miliaresis, G. C. and Argialas, D. (1999). Segmentation of physiographic features from the global digital elevation model/gtopo30. *Computers & Geosciences*, 25(7):715–728.

[Muralikrishna and Manickam, 2017] Muralikrishna, I. V. and Manickam, V. (2017). Chapter fourteen - air pollution control technologies. In Muralikrishna, I. V. and Manickam, V., editors, *Environmental Management*, pages 337–397. Butterworth-Heinemann.

[Nasr, 2007] Nasr, M. R. (2007). Open source software: The use of open source gis software and its impact on organizations. *PDF). Middlesex University/MIT*.

[Nielson, 1983] Nielson, G. (1983). A method for interpolating scattered data based upon a minimum norm network. *Mathematics of Computation - Math. Comput.*, 40:253–253.

[Nieva, 2019] Nieva, R. (2019). Google maps has now photographed 10 million miles in street view.

[Ninyerola et al., 2007] Ninyerola, M., Pons, X., and Roure, J. M. (2007). Objective air temperature mapping for the iberian peninsula using spatial interpolation and gis. *International Journal of Climatology: A Journal of the Royal Meteorological Society*, 27(9):1231–1242.

[NOAA, 2021] NOAA (2021). Climat messages.

[of Commerce, 2021] of Commerce, U. D. (2021). National oceanic and atmospheric administration. Available online at `http://www.noaa.gov/`. Last seen in 2021-02-25.

[OpenWeather, 2021] OpenWeather (2021). Openweathermap. Available online at `https://openweathermap.org/`. Last seen in 2021-02-25.

[Organization, 2003] Organization, W. M. (2003). *Aircraft meteorological data relay (AMDAR) reference manual*. Secretariat of the World Meteorological Organization.

[Organization, 2009] Organization, W. M. (2009). Practical help for compiling climat reports.

[Organization, 2017] Organization, W. M. (2017). *Manual on Codes International Codes Volume I.1 Annex II to the WMO Technical Regulations Part A – Alphanumeric Codes. 2011 edition, Updated in 2017,*. World Meteorological Organization.

[Parson, 2003] Parson, E. A. (2003). *Protecting the Ozone Layer*. Oxford University Press.

[Pina, 2021] Pina, J. (2021). Fogos.pt. Available online at `https://fogos.pt/`. Last seen in 2021-02-25.

[qgis, 2021] qgis (2021). Qgis. `https://github.com/qgis/QGIS`.

[RENKA et al., 1984] RENKA, R. J., Renka, R. L., and CLINE, A. K. (1984). A triangle-based c^1 interpolation method. *The Rocky Mountain Journal of Mathematics*, 14(1):223–237.

[Scher and Messori, 2019] Scher, S. and Messori, G. (2019). How Global Warming Changes the Difficulty of Synoptic Weather Forecasting. *Geophysics Research Letters*, 46(5):2931–2939.

[Software, 2014] Software, B. D. (2014). Compare projects - open hub 2014.

[Stonebraker, 2010] Stonebraker, M. (2010). Sql databases v. nosql databases. *Communications of the ACM*, 53(4):10–11.

[Taneja and Gupta, 2014] Taneja, S. and Gupta, P. R. (2014). Python as a tool for web server application development. *Int. J. Information, Commun. Comput. Technol*, 2(1):2347–7202.

[Van Zyl, 2001] Van Zyl, J. J. (2001). The shuttle radar topography mission (srtm): a breakthrough in remote sensing of topography. *Acta Astronautica*, 48(5-12):559–565.

[WMO, 2018] WMO (2018).

[World Meteorological Organization, 2003] World Meteorological Organization (2003). FM 92 GRIB. PDF. Available online at https://www.wmo.int/pages/prog/www/DPS/FM92-GRIB2-11-2003.pdf.