



Universidad  
de La Laguna

Escuela Superior de  
Ingeniería y Tecnología  
Sección de Ingeniería Informática

---

## **Portal web para información meteorológica.**

*Meteorological information website.*

Ángela Hernández Delgado.

Dpto. Física Fundamental y Experimental Electrónica y Sistemas.

Escuela Técnica Superior de Ingeniería Informática

Trabajo de Fin de Grado

---

La Laguna, 23 de Julio de 2014.



D. **Juan Carlos Pérez Darías**, con N.I.F. 45.441.625-L profesor Titular de Universidad adscrito al Departamento de Nombre del Departamento de la Universidad de La Laguna

D. **Albano González Fernández**, con N.I.F. 45.448.401X profesor Titular de Universidad adscrito al Departamento de Nombre del Departamento de la Universidad de La Laguna

## **CERTIFICA**

Que la presente memoria titulada:

*“Portal web para información meteorológica”*

ha sido realizada bajo su dirección por D. Ángela Hernández Delgado, con N.I.F. 78.644.165-N.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 23 de Julio de 2014.



## **Agradecimientos**

A D. Juan Carlos Pérez Darías y a D. Albano González Fernández por mostrarme su apoyo, brindarme su tiempo, aportarme sus conocimientos, y sin los cuales este trabajo no hubiera sido posible.

A mi familia por el apoyo continuo y dedicación.

A mis amigos por estar siempre a mi lado apoyándome y dándome ánimos.

A Román Weissbacher Rodríguez, por ser como eres y estar ahí en todo momento.

Y para finalizar he de agradecer a la Universidad de La Laguna por esta oportunidad y a todos los docentes que han hecho posible que llegará hasta aquí.



## Resumen

*En los últimos años, los modelos numéricos para la predicción meteorológica se han convertido en una herramienta indispensable para proporcionar información cada vez más precisa del estado de la atmósfera, no sólo a los especialistas en física de la atmósfera, sino a la sociedad en general.*

*Estos modelos, que hacen uso de las ecuaciones que gobiernan la dinámica de fluidos, se ejecutan en grandes sistemas de cómputo. Sin embargo, la precisión de los mismos viene determinada por su capacidad para describir los procesos complejos que ocurren en el sistema tierra-atmósfera-océanos. La complejidad de estos procesos hace que, a día de hoy, sea imposible simularlos a escala global con la suficiente resolución. Eso hace que, aunque los procesos que ocurren a escala sinóptica quedan bien representados, los procesos que ocurren a escala local debido a la orografía o a las condiciones específicas de la región no sean considerados.*

*Por ello, es necesario utilizar técnicas que permitan, a partir de estas simulaciones globales a baja resolución, obtener unas simulaciones más precisas para una región determinada del planeta. Una de estas técnicas, conocida como “downscaling dinámico”, permite utilizar las salidas de los modelos globales como entradas a otros modelos que son capaces de resolver, para la región de interés, los procesos que ocurren a escala local.*

*Haciendo uso de estas técnicas, el grupo GOTA (Grupo Observatorio de la Tierra y de la Atmósfera) de la Universidad de La Laguna realiza diariamente predicciones meteorológicas a 48 horas para la región de Canarias, usando el modelo regional WRF.*

*En este contexto, el objetivo de este trabajo consiste en elaborar una plataforma web que muestre de forma interactiva los resultados de estas simulaciones, permitiendo al usuario visualizar los mapas de predicción para las múltiples variables meteorológicas. Además, esta aplicación será capaz de mostrar al usuario cómo va a ser la evolución temporal de las principales variables (meteogramas) para una localización concreta dentro de la región de interés. También se pretende que esta plataforma sirva como herramienta de diagnóstico para el equipo de investigación, ya que será capaz de mostrar la comparación entre las predicciones realizadas por el grupo GOTA y datos observacionales adquiridos por estaciones meteorológicas.*

## Palabras claves

WRF, modelos numéricos, parametrización, meteograma, portal meteorológico.







## **Abstract**

*In the last years, the weather forecast numerical models have been becoming an essential tool to provide information about the state of the atmosphere, not only to the atmospheric physics experts but also to the general society.*

*These models, that use dynamic fluids equations, are running in large computational facilities. Nevertheless, their accuracy is determined by the skills to describe the complex processes involved in the earth-atmosphere-ocean system. The complexity of these processes make that, to date, it is not possible to simulate them at a global scale with enough resolution. So, despite they are able to reproduce the processes occurring at a synoptic scale, those developing at a local scale due to the orography or the specific conditions of the region are not well considered.*

*For that reason, is necessary to use technics that allows us, from these low-resolution global models, to obtain more accurate simulations for an specific region of the planet. One of this technics, known as “dynamic downscaling”, allows us to use the results of the global models as input data for other models that have the ability to solve the local scale processes for the selected region.*

*Using these techniques, the group GOTA (Grupo Observatorio de la Tierra y de la Atmósfera) of the University of La Laguna performs every day, 48 hour weather forecasting for Canary region, using the regional model WRF.*

*On this context, the aim of this work is to develop a web site to show, in an interactive way, the results of these weather simulations, allowing to the user to display the forecast maps for the different meteorological variables. In addition, this web application should be able to show the temporal evolution of the main variables (meteogram) for a specific location in the region under study. Also, we want that this platform can be used as a diagnostic tool for the research group, because it would be capable to show the comparative between the forecast made by the GOTA and observational data acquired by meteorological stations.*

## **Keywords**

WRF, numerical models, parameterization, meteogram, meteorological portal.



# Índice general

Agradecimientos .....	5
Resumen .....	7
Abstract .....	10
Índice de figuras y tablas.....	15
CAPÍTULO 1 INTRODUCCIÓN.....	17
CAPÍTULO 2 OBJETIVO .....	20
CAPÍTULO 3 DISEÑO E IMPLEMENTACIÓN .....	23
3.1 Cluster de GOTA.....	24
3.2 Demonio de la aplicación web. ....	25
3.2.1 Generación de simulaciones. ....	26
3.2.2 Obtención de datos de la AEMET.....	29
3.3 Aplicación Web.....	30
3.3.1 Meteograma.....	32
3.3.2 Animación.....	34
3.3.3 Presentación.....	35
3.3.4 Comparativa GOTA vs AEMET.....	37
CAPÍTULO 4 HERRAMIENTAS.....	39
4.1 Herramientas de tratamiento de datos. ....	39
4.1.1 NetCDF.....	39
4.1.2 PyNio.....	39
4.1.3 Numpy.....	40
4.1.4 Matplotlib.....	40
4.1.4.1 Basemap.....	40
4.1.4.2 Generación de gráficas.....	41
4.2 Gestión de la información. ....	42
4.2.1 SQLite.....	42
4.2.2 JSON.....	42
4.3 Aplicación Web.....	42
4.3.1 Django.....	42
4.3.2 HTML.....	42
4.3.3 CSS.....	43
4.3.4 Bootstrap.....	43

4.3.5 Javascript y JQuery.....	43
4.3.6 Geolocalización. ....	44
4.3.7 BeautifulSoup4.....	44
4.3.8 Urllib2.....	45
4.3.9 Python Minimal DOM.....	45
4.3.10 Python JSON. ....	45
4.3.11 Python Datetime. ....	46
4.3.12 Python RE.....	46
4.3.13 Python OS.....	46
4.3.14 Python Glob.....	46
4.3.15 Python Shutil. ....	46
4.3.16 Python Config Parser.....	47
4.3.17 JSON GeoNames.....	47
CAPÍTULO 5 RESULTADOS. ....	48
5.1 Herramientas de representación y difusión. ....	48
5.2 Herramientas de diagnóstico. ....	50
5.3 Meteograma.....	51
5.4 Análisis Jerárquico. ....	52
Conclusiones y trabajos futuros. ....	55
Conclusions.....	57
Bibliografía. ....	59



## Índice de figuras y tablas

Figura 1. Model Grid.....	18
Figura 2. Procesos atmosféricos.....	18
Figura 3. Esquema de funcionamiento de la técnica de Downscaling dinámico. ....	19
Figura 4. Dominios .....	21
Figura 5. Resolución de los dominios. ....	21
Figura 6. Diseño de la aplicación. ....	23
Figura 7. Demonio de la aplicación web. ....	26
Figura 8. MTV Django. ....	30
Figura 9. Demonio cluster. ....	31
Figura 10. Página de inicio.....	32
Figura 11. Meteograma .....	33
Figura 12. Definir mapa. API Google Maps. ....	34
Figura 13. Animación. ....	35
Figura 14. Presentación. ....	36
Figura 15. Comparativa.....	37
Figura 16. Configuración Basemap para la zona de Tenerife. ....	40
Figura 17. API Javascript Google Maps.....	44
Figura 18. Temperatura a 2 metros para la Isla de Tenerife. ....	48
Figura 19. Viento. ....	49
Figura 20. Lluvia. ....	50
Figura 21. Resultado comparativa. ....	51
Figura 22. Resultado meteograma.....	52
Figura 23. Análisis Jerárquico.....	54





## INTRODUCCIÓN

Comenzaremos diferenciando los conceptos de tiempo y clima. El tiempo atmosférico es la percepción obtenida de las variables atmosféricas y su evolución tras el transcurso de las horas, días y semanas, mientras que, el clima es el estudio del progreso del tiempo a mayor escala temporal, es decir, en años, décadas e incluso siglos.

Tradicionalmente, la predicción meteorológica ha sido desarrollada por expertos que, a partir de los datos observacionales a escala global adquiridos desde satélites, estaciones meteorológicas, radiosondeos, instrumentación a bordo de barcos o aviones, etc., generan unos mapas sinópticos que describen el comportamiento general de la atmósfera y su posible evolución [1]. No obstante la atmósfera es un fluido en el que interactúan gran cantidad de fenómenos que si varían mínimamente pueden modificar el sistema en su conjunto. Por ello, con la aparición de los sistemas de supercomputación, los modelos numéricos han ido adquiriendo cada vez más relevancia para complementar las funciones de los expertos, Estos modelos basan su predicción en el conocimiento de las condiciones de la atmósfera en un momento dado y, a partir de ahí será capaz de predecir el estado del fluido en el futuro. [2].

Los modelos numéricos están basados en fundamentos matemáticos que nos permiten predecir a partir de las condiciones iniciales del sistema como variará el mismo. Por ello, es importante recoger la información pertinente para la obtención de unas condiciones iniciales bien definidas. No obstante, en estos modelos existe una cierta incertidumbre que no puede ser predicha. Esta limitación se basa en el efecto mariposa: cualquier variación por pequeña que sea puede provocar grandes reacciones en el sistema. Por lo que existirán fenómenos que no serán vaticinados con el cien por cien de fiabilidad.

Los modelos climáticos y del tiempo se basan en leyes fundamentales de la física como la ley de la dinámica de Sir Isaac Newton, la ley de la termodinámica de Rudolf Clausius y las ecuaciones gobernantes de la transferencia de la radiación de Arthur Schuster.

En el siglo veinte, los científicos comenzaron a aplicar estas leyes fundamentales de la física a la atmosfera. Vilhelm Bjerknes escribió las ecuaciones que se usan para pronosticar el viento en los modelos del tiempo y clima. Lewis Richardson fue el padre del pronóstico del tiempo numérico. John Von Neumann fue el primer científico que controló el funcionamiento de un modelo numérico de predicción en un ordenador.

Los modelos numéricos requieren el uso de grandes cantidades de datos y de la realización de cálculos complejos con una resolución tanto espacial como temporal muy

detallada, para que aminore la incertidumbre de la predicción. Para ello tendríamos que utilizar superordenadores con la suficiente capacidad para realizar dichos cálculos, lo cual es aún imposible.

Hoy en día la resolución de los modelos a escala global no permite simular con detalle todos los procesos que intervienen en este sistema complejo. Por ello, es necesario realizar un amplio conjunto de hipótesis para poder resolverlo. El primer paso consiste en discretizar el problema. Con este fin, se utiliza el modelo Grid en el cual se divide la Tierra en una serie de “boxels” o cajitas en las que se resolverán los distintos procesos físicos que allí ocurren.

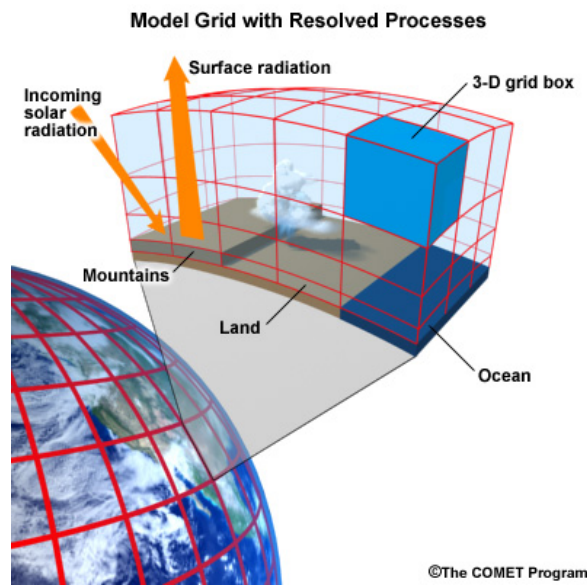


Figura 1. Model Grid

A partir de ahí, para poder simular los procesos que se producen entre los distintos puntos de la malla del sistema se han de usar las ecuaciones de la dinámica de fluidos.

Al realizar el estudio de los modelos del tiempo y climáticos de forma global perdemos la percepción de los diversos componentes del modelo Grid. Si observamos un elemento del sistema de la malla a gran escala comprobamos que estamos perdiendo información relevante a la orografía del lugar, cómo se conforma la lluvia, cómo afecta la brisa marina a la costa, cómo interactúa la radiación solar con los distintos elementos, cómo la vegetación afecta en el

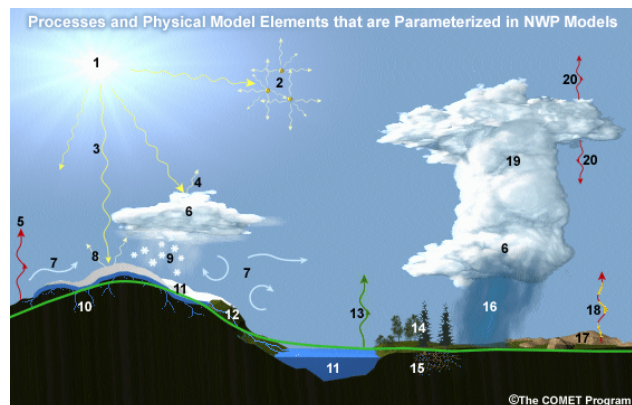


Figura 2. Procesos atmosféricos

viento, la filtración de la lluvia, la convergencia entre el agua dulce de los ríos con el agua marina, o cómo el deshielo de agua marina afecta en la circulación del océano.

Actualmente no existe un sistema de cómputo capaz de abarcar la complejidad del modelo con una adecuada resolución espacial a escala global por lo que hemos de optar por dos opciones: aumentar la complejidad del modelo o realizar un aumento en la resolución de las escalas, es decir, realizar estudios en escalas locales a mayor resolución.

Para estudiar los modelos a escala local hemos de utilizar métodos para realizar una ampliación o “zoom” de los GCMs (Global Climate Models ). Existen dos formas principales para realizar esta resolución de escala. Una es la regionalización dinámica o downscaling, que utiliza la salida de GCM para proporcionar las condiciones iniciales de contorno del modelo regional, el cual está definido con una mayor resolución espacial por lo que es capaz de simular las condiciones locales en mayor detalle. La otra manera es la regionalización estadística que se realiza a partir de la relación estadística de las observaciones entre las variables de gran escala y las variables locales [3]. En el actual proyecto utilizaremos la técnica de downscaling dinámico.

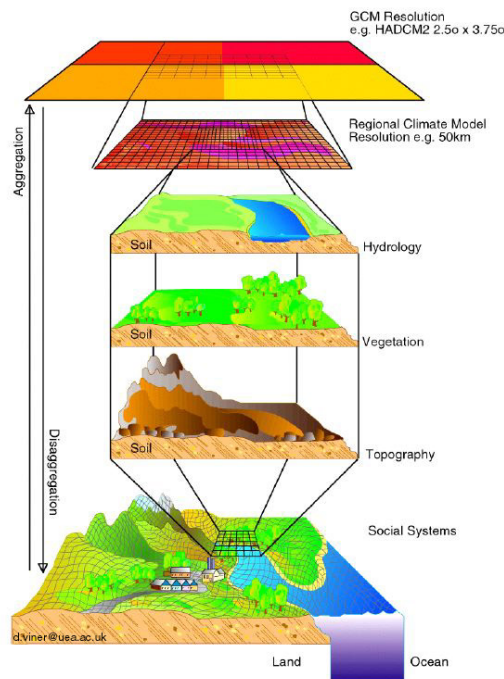


Figura 3. Esquema de funcionamiento de la técnica de Downscaling dinámico. [26]

### OBJETIVO

En este contexto, el objetivo de este proyecto de final de grado consiste en elaborar una plataforma web que permita visualizar de forma interactiva la información meteorológica para las islas Canarias. Esta plataforma mostrará información de predicciones meteorológicas de alta resolución para la región canaria realizadas mediante modelos numéricos por el Grupo de Observación de la Tierra y la Atmósfera (GOTA) de la Universidad de La Laguna (ULL).

GOTA tiene dentro de sus objetivos la investigación de la modelización meteorológica y climática en regiones de orografía compleja como es el caso del archipiélago. En territorios de características complejas como éste se han de estudiar no solo los procesos sinópticos que se encuentran representados en los modelos de circulación general (GCM's de sus siglas en inglés) y que estamos acostumbrados a ver en los medios de comunicación, sino que los procesos mesoescalares también tienen gran relevancia, ya que modifican de forma sustancial estos procesos de escala global. Por tanto, es de gran importancia una correcta modelización del sistema con el fin de representar correctamente estos procesos a escala local.

En este sentido, el grupo de investigación ha adoptado el modelo Weather Research and Forecasting (WRF) [4] para implementar una técnica de downscaling dinámico para la región de Canarias. Así, que se utiliza la información proporcionada por los modelos GCM's como condiciones de inicio y de contorno para alimentar unos dominios de área limitada (en nuestro caso una región centrada en Canarias) en los que la resolución de las ecuaciones que rigen la dinámica atmosférica/oceánica/terrestre tengan en cuenta los distintos procesos que ocurren a escala local (efectos de la orografía, procesos mar/tierra, desarrollo de nubes,...).

Utilizando esta técnica, el Grupo realiza de forma rutinaria simulaciones de predicción meteorológica para las Islas Canarias. Para ello, hace uso de este modelo de área limitada con varios niveles de anidamiento a través de los distintos dominios. Así el primer dominio definido fue diseñado para observar las principales características sinópticas de la región con 27 kms de resolución horizontal. La segunda región está centrada en la zona de interés (el archipiélago canario) a una resolución horizontal de 9 kms. El tercer dominio está centrado en la zona occidental de la misma a una resolución de 3 kms. La cuarta y última zona es la región de Tenerife con una resolución de 1 Km.

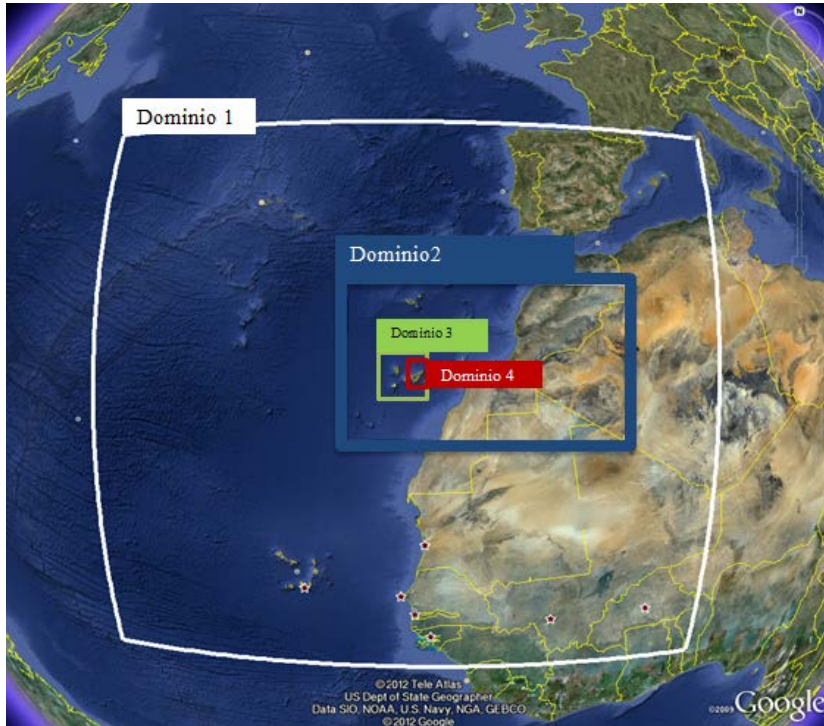


Figura 4. Dominios

Como es lógico, a medida que incrementamos la resolución se mejora la representación de la región de interés. Por ejemplo, en la figura 5 se puede observar cómo se representa la orografía de la Isla de Tenerife cuando se usa la resolución de los distintos dominios. Llama la atención que aún realizando un nivel de anidamiento (dominio 1), la Isla queda representada por un par de píxeles con una altura máxima que no excede los 200 metros. Sin embargo, en el dominio 4, con una resolución de 1 km., ya se puede observar que la orografía ya queda representada con bastante precisión.

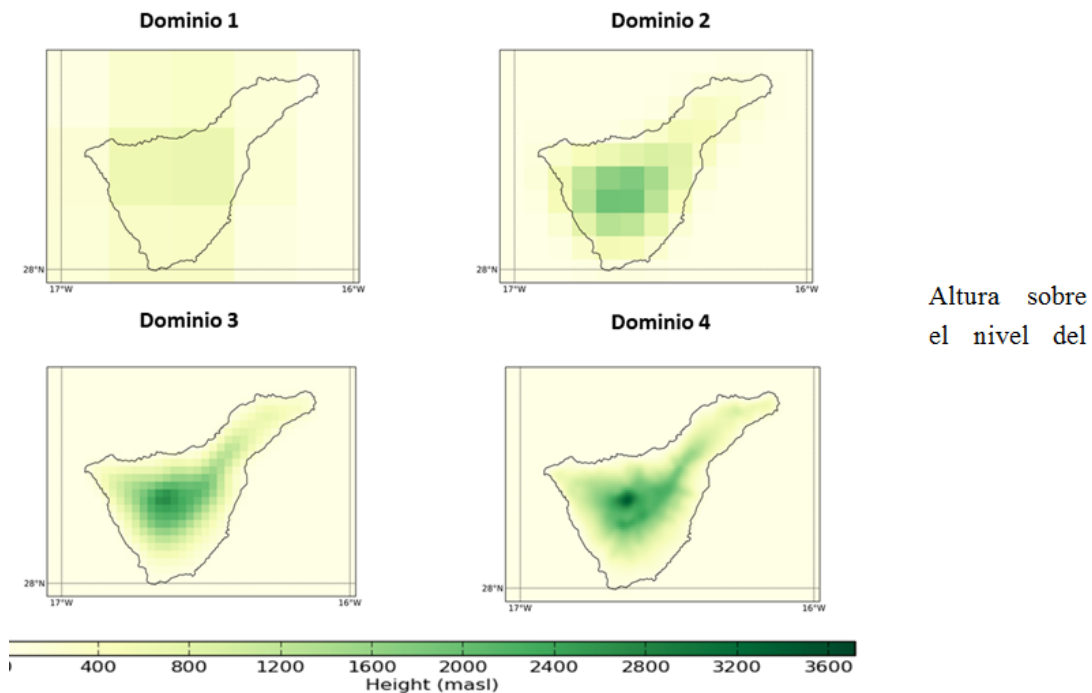


Figura 5. Resolución de los dominios.

Sin embargo, aún incrementando la resolución hasta 1km como en el caso del dominio 4, debido a la complejidad del sistema, muchos de los procesos físicos que ocurren no pueden ser resueltos de forma explícita y es necesario realizar una serie de hipótesis que relacionen los fenómenos que ocurren a una escala muy pequeña con las condiciones que se pueden evaluar directamente a la resolución del modelo. Para resolver estas incógnitas, se hace uso de *parametrizaciones*. Sin embargo, aunque existen un gran número de parametrizaciones desarrolladas (nubes, radiación, procesos en superficie, capa de mezcla,...), no todas son válidas para cualquier región del globo, lo que hace necesario un estudio exhaustivo de las condiciones locales para generar el conjunto de parametrizaciones que mejores resultados proporcionan en la región de interés y en las diferentes condiciones que se desarrollan en este territorio.

El GOTA desde hace varios años viene realizando estudios en esta línea. Por una parte, de forma rutinaria se realizan, diariamente, predicciones meteorológicas a 48 horas. Por otra parte, se realizan estudios climáticos que permitan elaborar proyecciones del efecto del cambio climático sobre la región canaria. El presente proyecto pretende proporcionar una herramienta que permita evaluar la precisión en la determinación de las diferentes variables proporcionadas por el modelo y que permita al usuario (especializado o no) obtener información de las previsiones meteorológicas de alta resolución para Canarias de una forma sencilla.

El desarrollo de la plataforma permitirá disponer al GOTA y a la comunidad en general, de una herramienta que posibilite, además de la visualización gráfica de la información meteorológica, la evaluación de las distintas parametrizaciones utilizadas por el grupo de investigación para el modelado de los distintos procesos que intervienen en la dinámica atmosférica. El portal ha de mostrar de forma interactiva la información meteorológica en Canarias generada a partir de las simulaciones generadas para distintas configuraciones del modelo. A través de este portal, el usuario puede visualizar las salidas para las variables atmosféricas más habituales así como la evolución temporal de estas variables para un punto seleccionado en la región de interés (meteogramas). Además, esta herramienta ha de servir de diagnóstico para el equipo de investigación permitiendo la comparación de las salidas del modelo con datos observacionales medidos en estaciones meteorológicas.

### DISEÑO E IMPLEMENTACIÓN

El diseño e implementación del presente proyecto se encuentra dividido en dos partes: la primera de ellas, que denominaremos “cluster gota”, se ejecuta en el sistema de cómputo donde se realizan las predicciones y se encarga de obtener los datos necesarios para su representación así como del preprocesamiento de los mismos. Por su parte, la aplicación web constará de un demonio que recogerá los datos proporcionados por el cluster y genera las imágenes georreferenciadas para su posterior presentación. Además de una aplicación en Django que se encargará de procesar los datos enviados por el demonio. Estos datos contienen información relevante del cluster y de los archivos generados por el mismo. Django generará la representación de los datos en una página web sencilla y eficaz. En la figura 4 se muestra un esquema de bloques implementados en este trabajo así como su interrelación. En las siguientes secciones se describen los diferentes módulos.

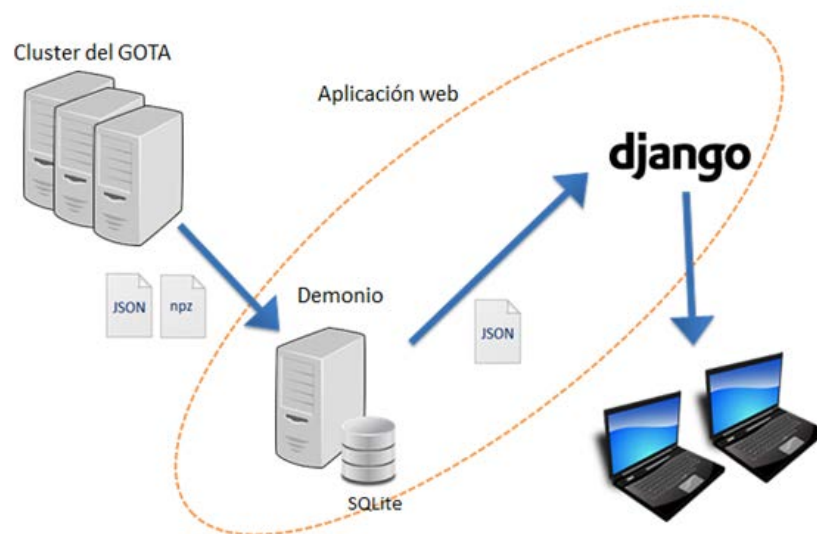


Figura 6. Diseño de la aplicación.

### 3.1 CLUSTER DE GOTA

El Cluster Gota se encarga de generar las simulaciones de predicción meteorológica del grupo GOTA de la Universidad de La Laguna. Estas simulaciones son realizadas de forma rutinaria cada día y en ellas se generan las predicciones para las siguientes 48 horas en la región de las islas Canarias.

El procedimiento habitual es el siguiente: a las 22:00 horas se descargan desde el “National Climatic Data Center” de la Administración Americana NOAA, los datos del modelo global que será utilizado como condiciones iniciales y de frontera de nuestra simulación regional. A continuación se realizan las simulaciones a 48 horas, tomando como punto de partida las 18 horas del día actual. Durante este periodo se almacenan las salidas horarias del modelo para cada uno de los cuatro dominios. Además, habitualmente se realizan distintas simulaciones simultáneas para evaluar el comportamiento de diferentes conjuntos de hipótesis (parametrizaciones) para la región de interés.

El modelo numérico de predicción utilizado (WRF), genera una gran cantidad de información que da cuenta de las distintas variables atmosféricas, del terreno o de los componentes hídricos del modelo que son considerados en el sistema. Sin embargo, para este trabajo se consideran únicamente las variables más utilizadas para describir la meteorología, como son la temperatura, las precipitaciones, los vientos y la cobertura nubosa, que se corresponden a variables T2, U10, V10, RAINC, RAINNC, CLDFRA de WRF, respectivamente.

Los datos de estas variables se almacenan en formato netcdf, pero debido al gran tamaño de estos ficheros, se estimó conveniente extraer únicamente la información de interés a unos ficheros más manejables que se almacenan en formato npz.

La información recopilada nos permite representar la temperatura, lluvia y viento de una región durante las horas de predicción o la visualización de la evolución temporal de la variable en cuestión para un punto determinado de la región.

La temperatura a 2 metros es representada gracias al uso de la variable T2. Con U10 y V10 se recogen los datos de los vectores de dirección del viento a 10 metros de altura, de tal manera que U10 nos proporciona las componente zonal (en la dirección oeste-este) mientras que V10 nos indica la componente meridional (en la dirección norte-sur); con ellos podremos mostrar la dirección y velocidad del viento en cada rejilla del modelo. RAINC y RAINNC son las variables encargadas de la lluvia, RAINC nos proporciona la componente convectiva de la precipitación (aquella que se genera por procesos convectivos), mientras que RAINNC tiene en cuenta la precipitación generada por procesos no convectivos. Al sumar ambas variables podremos obtener el total de



precipitación. Por último, hallaremos la cobertura nubosa de la zona seleccionada, en este caso usaremos CLDFRA, una variable cuatridimensional que proporciona la fracción de píxel (en cada nivel vertical definido en el modelo) que se encuentra cubierta por la nube.

Las variables con las que trabajamos en este proyecto generalmente son tridimensionales (latitud, longitud y tiempo). No obstante, existen variables cuatridimensionales como es el caso de CLDFRA en las que se representa una cuarta variable que indica el nivel vertical de la variable en cuestión.

Se ha de tener en cuenta que estas variables se encuentran disponibles para cada dominio y para cada parametrización. En el presente proyecto se muestran los resultados para predicciones realizadas para diferentes esquemas de microfísica, denominadas WDM6, WSM6 y Thompson.

Una vez se ha recogido la información y tras generar el fichero correspondiente a la predicción, crearemos un fichero ‘prediccion.json’ que nos permita guardar y compartir de una forma sencilla y rápida los datos más relevantes de la predicción realizada.

Este fichero contendrá la siguiente información:

- Fecha de realización de cada predicción.
- La hora de inicio.
- El número de dominios que se han utilizado.
- El número de parametrizaciones que se han usado y sus respectivos identificadores.
- El directorio donde se han guardado los .npz resultantes.
- Duración en horas de la predicción por cada zona.
- Se define una array “output\_time\_increments” que define el intervalo en minutos en los que se almacenan los datos en cada simulación.

Estos datos serán recogidos en el demonio de la aplicación web para la posterior generación de las simulaciones.

### **3.2 DEMONIO DE LA APLICACIÓN WEB.**

El demonio de la aplicación web recogerá los datos preprocesados en el Cluster y generará las imágenes con la representación visual de las diferentes variables para cada simulación y cada uno de la parametrizaciones anteriormente mencionados. También almacenará en la base de datos la información más relevante de las simulaciones, variables, parámetros, dominios y lugar donde guardaremos las imágenes generadas. En él se creará un fichero JSON denominado ‘predicción’ que agrupará los datos

almacenados en la base de datos del demonio con la salvedad de que únicamente contendrá los datos correspondientes a la última semana.

Además, alojará los datos de la AEMET (Agencia Estatal de Meteorología) con el fin de comparar nuestras predicciones. Es decir, al acceder a los datos de la AEMET obtendremos los datos observacionales del día anterior, que cotejaremos con nuestras predicciones para así observar y comprobar la fiabilidad de nuestros modelos.

Estos procedimientos se describirán a continuación.

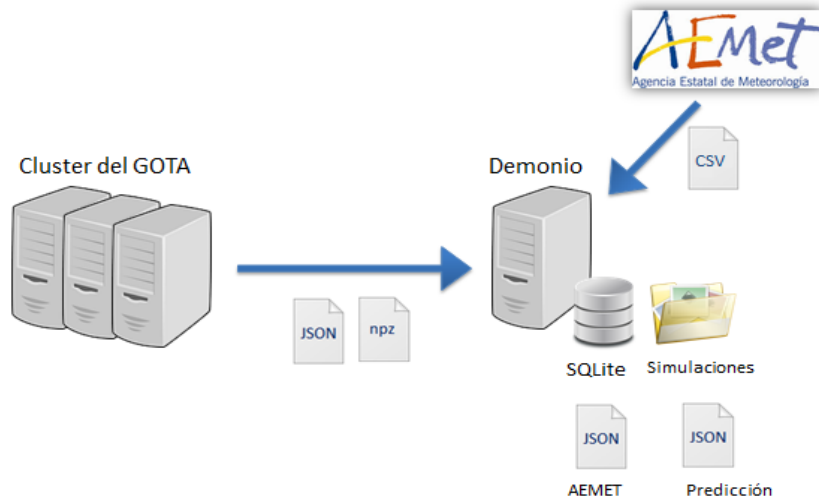


Figura 7. Demonio de la aplicación web.

### 3.2.1 GENERACIÓN DE SIMULACIONES.

La generación de simulaciones consiste en la obtención de los datos facilitados por el cluster, su procesamiento, la generación de las imágenes y la creación de una base de datos donde llevaremos un registro de las simulaciones generadas y los datos que se han utilizados en cada una de ellas.

Primeramente, extraeremos la información del cluster y los datos iniciales del archivo de configuración del demonio, en el cual se definen:

- **Las horas descartadas de las simulaciones.**

Estas horas hacen referencia a las primeras horas de simulación conocidas como spin-up que sirven de inicialización y en las cuales el sistema todavía no ha alcanzado una situación estable.

En el caso de simulaciones climáticas, este periodo transitorio puede ser de años, ya que es necesario estabilizar no solo los procesos atmosféricos sino los procesos de temperatura y humedad que ocurren en el suelo y subsuelo del dominio analizado.

- **Dominios.**

Se definen los dominios o regiones con las que trabajaremos. Como anteriormente ilustramos estas zonas serán tres, puesto que aunque definimos cuatro regiones la primera tiene la finalidad de captar los procesos que ocurren a escala sinóptica.

- **Variables.**

Especificamos las variables meteorológicas que representaremos; en este caso nos centraremos en la lluvia, cobertura nubosa, temperatura y viento según región.

- **Municipios.**

Se especifican los municipios de las islas para facilitar su localización cuando se muestre la información sobre un mapa. Actualmente, solamente hemos determinado a modo de ensayo los municipios de la isla de Tenerife. Para ello hemos de especificar la latitud y longitud de cada uno de ellos para su correcta geolocalización.

- **Datos de basemap.**

Para la representación gráfica de la información utilizamos la librería de Python matplotlib. Gracias a ésta podemos alcanzar nuestro objetivo definiendo los parámetros específicos de cada mapa según la zona a la que pertenezcan.

Basemap dispone de los siguientes parámetros entre otros:

1. **llcrnrlon.** (Longitud en grados de la esquina inferior izquierda del mapa).
2. **llcrnrlat.** (Latitud en grados de la esquina inferior izquierda del mapa).
3. **urcrnrlon.** (Longitud en grados de la esquina superior derecha del mapa).
4. **urcrnrlat.** (Latitud en grados de la esquina superior del mapa).
5. **projection.** (Tipo de proyección para la representación sobre un plano de información geoespacial).
6. **lat\_0.** (Latitud central de referencia).

## 7. lon\_0. (Longitud central de referencia).

Además, trataremos los datos almacenados en los archivos .npz generados en el cluster.

Las operaciones principales que se llevarán a cabo serán las siguientes: generación de los directorios en los que guardaremos las imágenes (se generan las imágenes .png para las distintas variables y para cada una de las horas de predicción. Estas se crean en el demonio para que la aplicación sólo tenga que acceder a ellas para mostrarlas, porque si las generará bajo demanda, tendría un alto costo en tiempo, puesto que la creación del mapa a partir de los datos georreferenciados generados durante la simulación, en matplotlib tarda considerablemente). Además se genera un fichero json con los datos referentes a las imágenes y las simulaciones, y se hace uso de la base de datos para el registro de las simulaciones. Por último, con el fin de optimizar el espacio de almacenamiento, se procede a la eliminación de imágenes y directorios de más de una semana de antigüedad con respecto a la última ejecución.

Las imágenes se crearán para cada uno de los dominios, para cada parametrización considerada y cada variable meteorológica a representar.

Tanto en la base de datos como en el fichero json definiremos los siguientes parámetros: fecha de realización de la simulación, las horas descartadas, la hora de inicio de la simulación, el intervalo entre las salidas del modelo (habitualmente 1 hora), el número de hora totales a representar, el número de dominios, numero de simulaciones, el directorio en donde guardaremos las imágenes, directorio donde se han almacenado previamente los .npz, las parametrizaciones, los dominios y las variables que se representan.

Para optimizar el espacio de almacenamiento el fichero json generado solamente tendrá la información de seis días anteriores al día de realización de las imágenes. Por ello es necesaria la base de datos, porque nos ayudará a tener un registro específico y estable de los datos obtenidos y generados.

Para la eliminación de directorios y ficheros de más de una semana de antigüedad listaremos los directorios actuales que hemos creado, según la fecha de identificación del fichero; a partir de ahí podremos determinar su obsolescencia. Una vez obtengamos la lista de las carpetas a eliminar comenzaremos con el proceso. El borrado de estos datos tienen como finalidad que la aplicación web no contenga y ocupe gran cantidad de espacio en disco. Además, la eliminación de los mismos no produce ningún inconveniente, ya que tenemos un registro de las simulaciones generadas y podríamos volver a recrearlas.

Con el fin de evaluar el comportamiento del modelo de predicción, se propone la comparación de los valores predichos con datos observacionales. Para ello, se ha elegido la red de observaciones más extensa y robusta disponible en Canarias, la de la AEMET (Agencia

Española de METerología). Para ello, se implementa un procedimiento que descarga de la página web la información de las últimas horas de todas las estaciones disponibles.

### **3.2.2 OBTENCIÓN DE DATOS DE LA AEMET.**

Para la obtención de estos datos, accedemos a la página web de la AEMET y recopilaremos los datos observacionales del día anterior. Al obtener estos datos descargamos los ficheros .csv de cada estación meteorológica de Canarias. Por ello creamos un fichero de registro en el cual guardaremos el nombre de la estación, el código de identificación de la misma, la latitud y longitud necesaria para su geolocalización y por último, su altitud.

Para adquirir la información referente a las lluvias, viento y temperatura hemos de acceder a cada uno de los ficheros anteriormente generados. Se ha de tener en cuenta que las simulaciones realizadas por la AEMET tienen una duración de 24h con los valores promediados en cada hora.

Leeremos los .csv y almacenaremos:

- Fecha y hora de cada simulación.
- Temperatura,
- Velocidad del viento.
- Dirección del viento.
- Racha.
- Dirección de la racha.
- Precipitaciones.
- Presión atmosférica.
- Tendencia.
- Humedad.

Con esta información almacenaremos en el fichero JSON “aemet.json” los datos observacionales de las últimas 24 horas y la estación en la cual fueron realizadas. Además, se implementa un mecanismo de borrado de información, de tal manera que ese fichero sólo se conserven los datos de la última semana.

Por último, se eliminarán todos los archivos .csv que hemos descargado, debido a que ya no serán de utilidad.

### 3.3 APLICACIÓN WEB.

La aplicación web será la encargada de mostrar de forma interactiva la información meteorológica en Canarias. Esto se llevará a cabo mediante la representación, para las distintas simulaciones realizadas, de las variables meteorológicas más habituales así como de los meteogramas. Además, deberá permitir la comparación de los resultados obtenidos en las predicciones de los días previos con los datos de observaciones registrados para el día anterior.

Para la generación de la página web hemos utilizado el framework Django. Esta herramienta se basa en el MVT (Modelo Vista Template). En nuestro caso el modelo será más simple, porque la base de datos propia de Django no juega un papel importante; en ella se almacena sólo información relevante a la aplicación, es decir, almacena datos referidos a accesos, sesiones, etc.

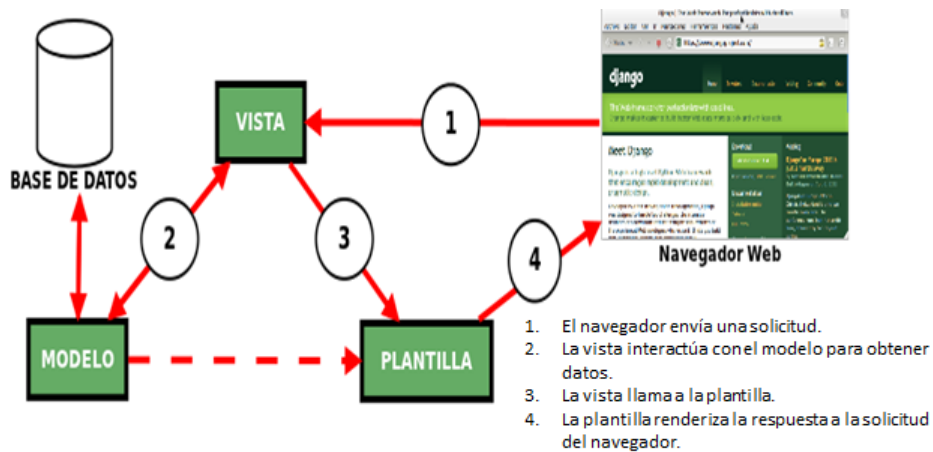


Figura 8. MTV Django.

Tal y como ilustramos en la “Figura 8. MTV Django”, el MTV se inicia tras el acceso de un usuario a la página web quien realizará una petición, ésta será controlada por la vista que consultará al modelo para obtener los datos necesarios para resolverla. Cuando éste los obtiene devuelve el resultado a la plantilla que les da el formato deseado y por último se le muestra la plantilla renderizada al usuario con la apariencia adecuada.

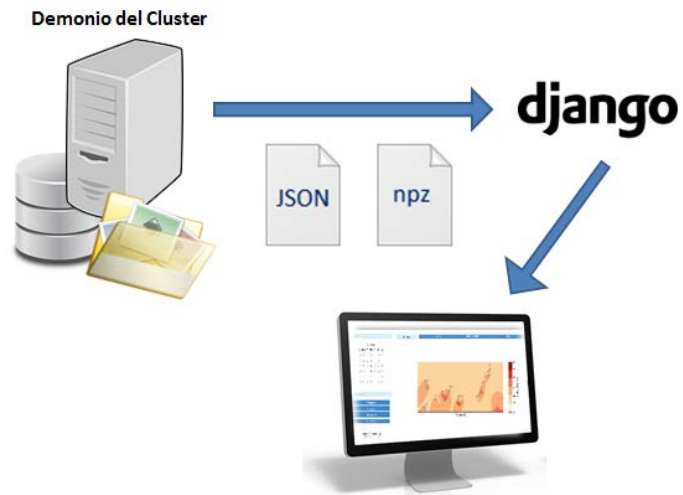


Figura 9. Demonio cluster.

Para describir la implementación de las diferentes funcionalidades de la aplicación, seguiremos un caso de uso típico de la misma.

La página de inicio de la aplicación mostrará un calendario con los días a los que les podemos acceder. Además, disponemos de un menú para elegir el tipo de representación que deseamos y el dominio de interés.

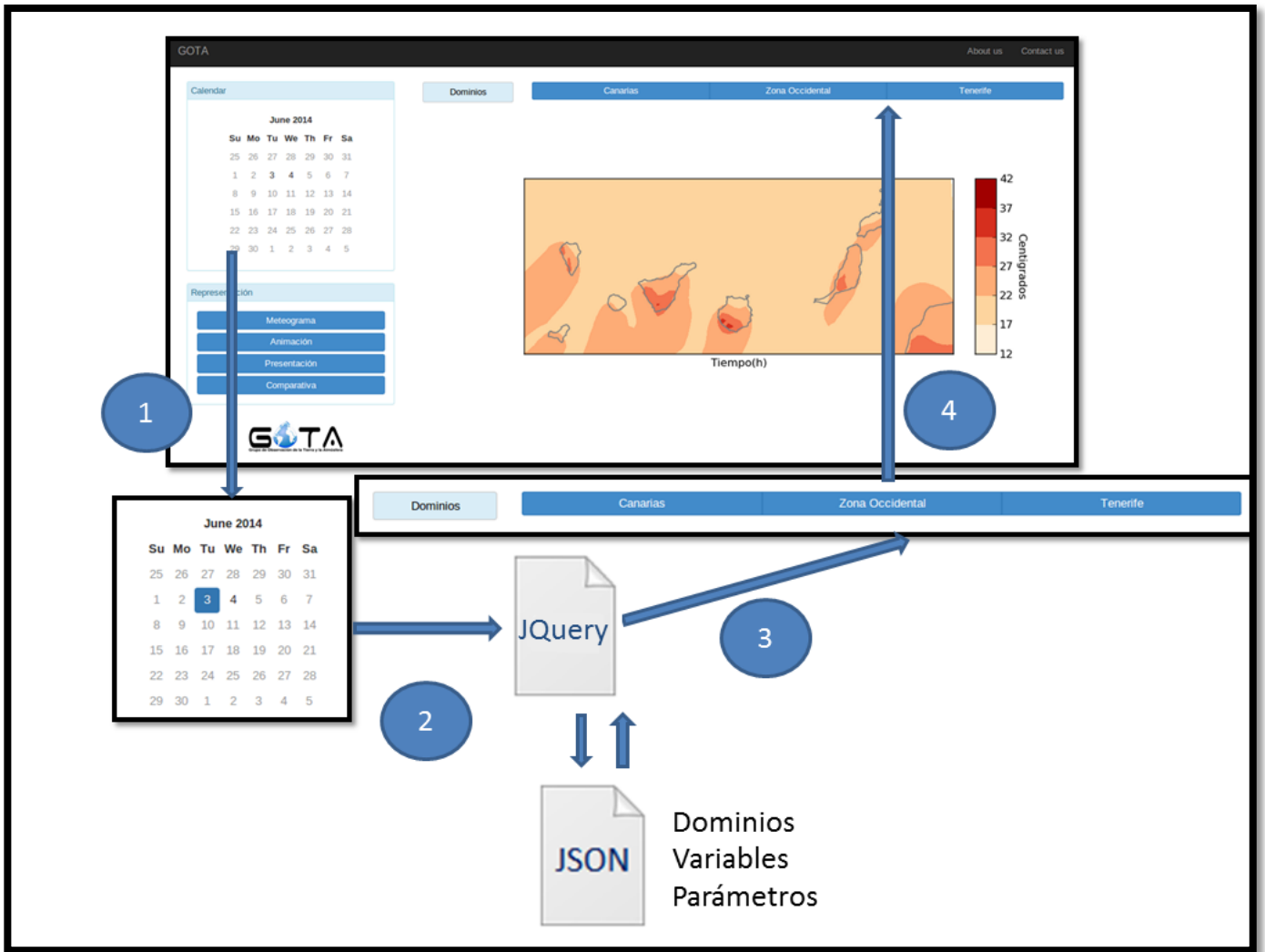


Figura 10. Página de inicio.

La barra de dominio cambiará según el día de predicción que seleccionemos. Para ello el módulo JQuery obtiene el valor del día seleccionado y le hace una petición a 'prediccion.json' que devolverá el dominio, variables y parametrizaciones disponibles para ese día. Seguidamente, JQuery modificará dichos valores en la página principal, se resetearán las variables y dominios seleccionadas.

### 3.3.1 METEOGRAMA.

El meteograma permite observar, sobre un mismo gráfico, la evolución temporal de las variables meteorológicas más habituales: temperatura, precipitación y vientos para un punto determinado.



Además, en el caso de que se hayan realizado múltiples simulaciones (parametrizaciones) permite evaluar el comportamiento de la media (ensemble) así como la dispersión de las distintas simulaciones.

En cambio el viento es representado el valor promediado cada tres horas, haciendo uso de diagrama de barbas, en las que se observan de forma clara su dirección y velocidad.

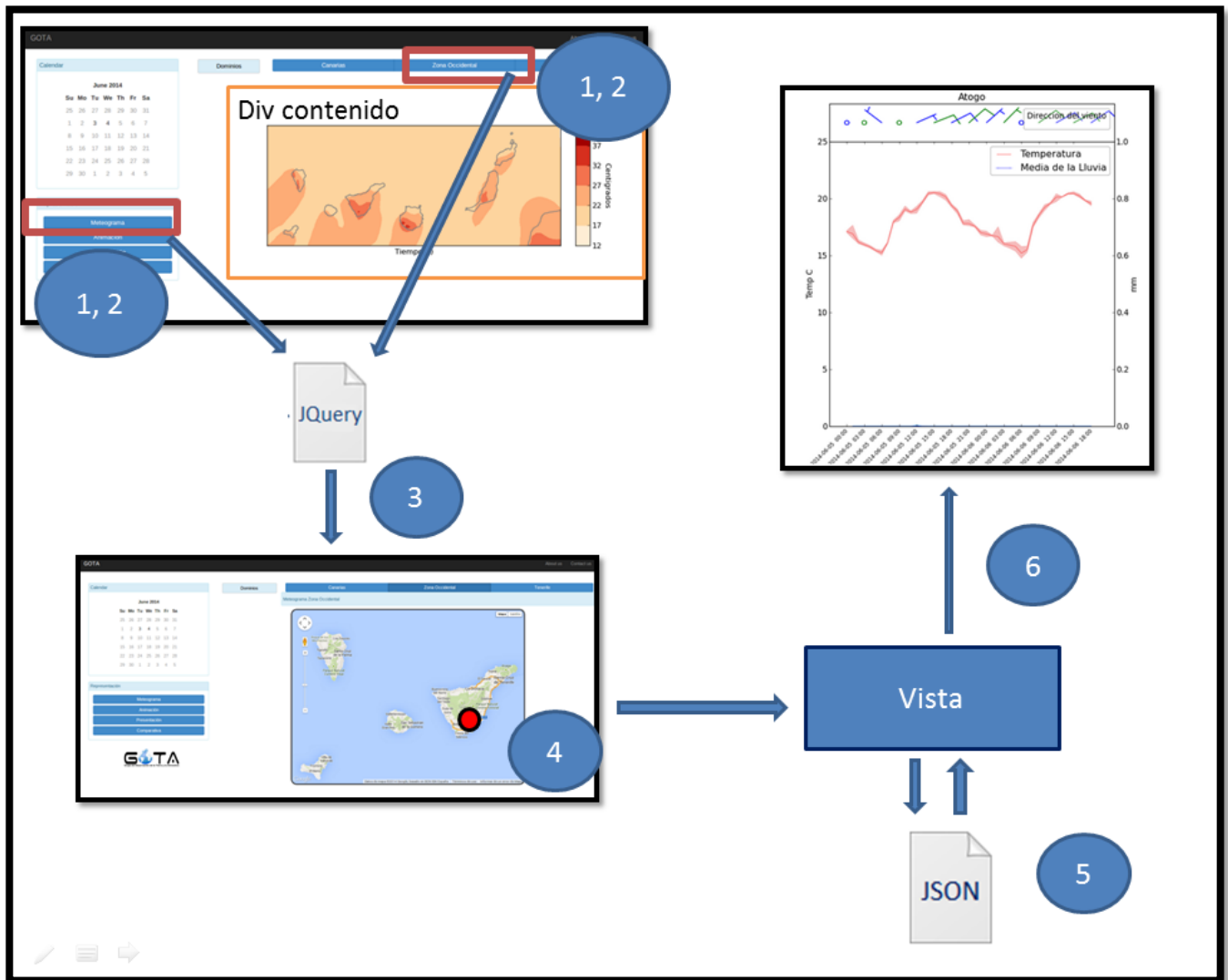


Figura 11. Meteograma

En primer lugar, seleccionaremos el tipo de representación y dominio. Posteriormente, se enviará a JQuery el valor de cada variable, y se comprobará que éstas no sean nulas. Seguidamente, JQuery comprueba que el flujo de operación es el adecuado en función del tipo de representación seleccionada. En este caso el valor es “meteograma” por lo que oculta todos los elementos de la etiqueta “div contenido” que define esa sección del HTML y se muestra el mapa de la zona. Este mapa será generado con la API de Google Maps y JavaScript. Para ello se ha definido las variables que determinan las condiciones iniciales del mapa para cada zona.

```
var myLatLng = new google.maps.LatLng(28.492833,-15.771732);
var mapOptions = {
  zoom: 8,
  center: myLatLng,
  mapTypeId: google.maps.MapTypeId.ROADMAP
};
map = new google.maps.Map(document.getElementById('map_canvas_2'), mapOptions);
```

Figura 12. Definir mapa. API Google Maps.

En la figura 12 podemos observar cómo se define un mapa en la API. Primeramente, la variable “myLatLng” define la latitud y longitud central del mapa, en mapOptions se especifica el zoom y el tipo de mapa que vamos a representar. Por último, la variable map genera en la zona “map\_canvas\_2” del HTML el mapa final.

Una vez se seleccione una localización dentro del mapa accederemos a la vista “coordenada” encargada de la generación del meteograma. La vista se encargará de conectarse con un JSON GeoNames para obtener según la longitud y la latitud del punto seleccionado el nombre del lugar. En la generación hallamos los días y horas de predicción, el máximo, mínimo y media de la temperatura y lluvia de las diferentes parametrizaciones (wdm6, wsm6 y Thompson en el ejemplo mostrado). Además, representaremos mediante barbas la dirección y velocidad del viento.

### 3.3.2 ANIMACIÓN.

La animación se encargará de representar dinámicamente la simulación que deseemos. Para ello previamente hemos de elegir el dominio, la parametrización y la variable atmosférica a observar.

Con esta opción se mostrarán las imágenes generadas por el demonio de la aplicación web. Serán visualizadas en forma de carousel con movimiento automático.

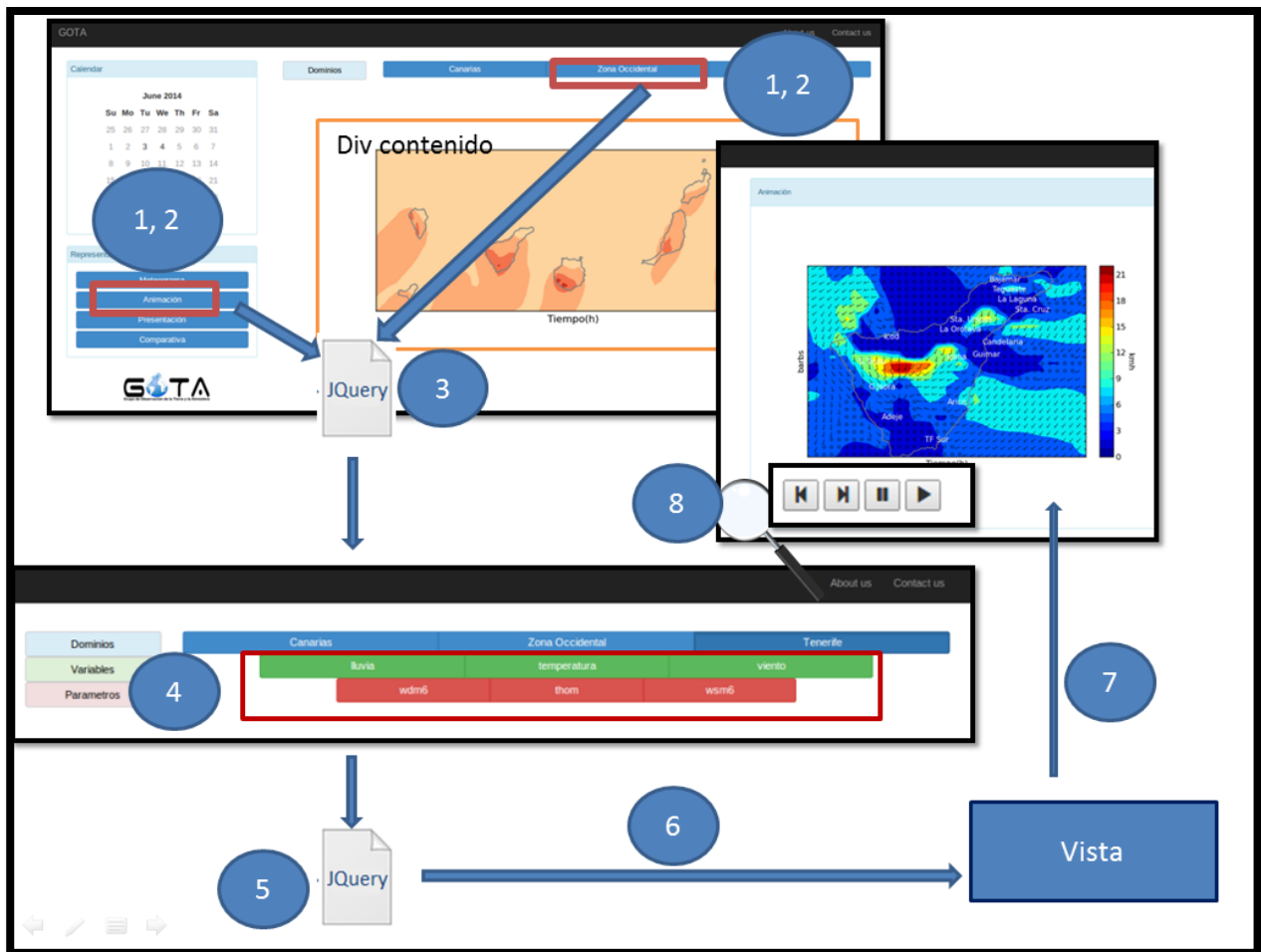


Figura 13. Animación.

Elegiremos el dominio y representación en el orden que se desee. JQuery comprobará que ninguno sea nulo, y si la representación es “animación” se desplegarán dos menús uno para la elección del parámetro y otra para escoger la variable. JQuery comprobará que estas variables sean correctas y se las pasará a la vista.

La vista se encargará de recoger los días y horas para los que se ha realizado la predicción, devuelve el link donde se encuentran las imágenes y una variable *animación=True* que será interpretada por el *template* para que el carrusel funcione de manera automática y genere controles integrados que nos permitan seleccionar, pausa o acceder a la imagen anterior o posterior, así como reanudar la animación. Esto permite una visualización continua y dinámica de los resultados.

### 3.3.3 PRESENTACIÓN.

La opción *presentación* se encarga de mostrar las simulaciones según el dominio, parametrización y variables atmosféricas seleccionadas. Su diferencia con el apartado

Animación es la visualización. Se presentan de forma estática las imágenes generadas y se puede elegir la hora de la simulación y acceder directamente.

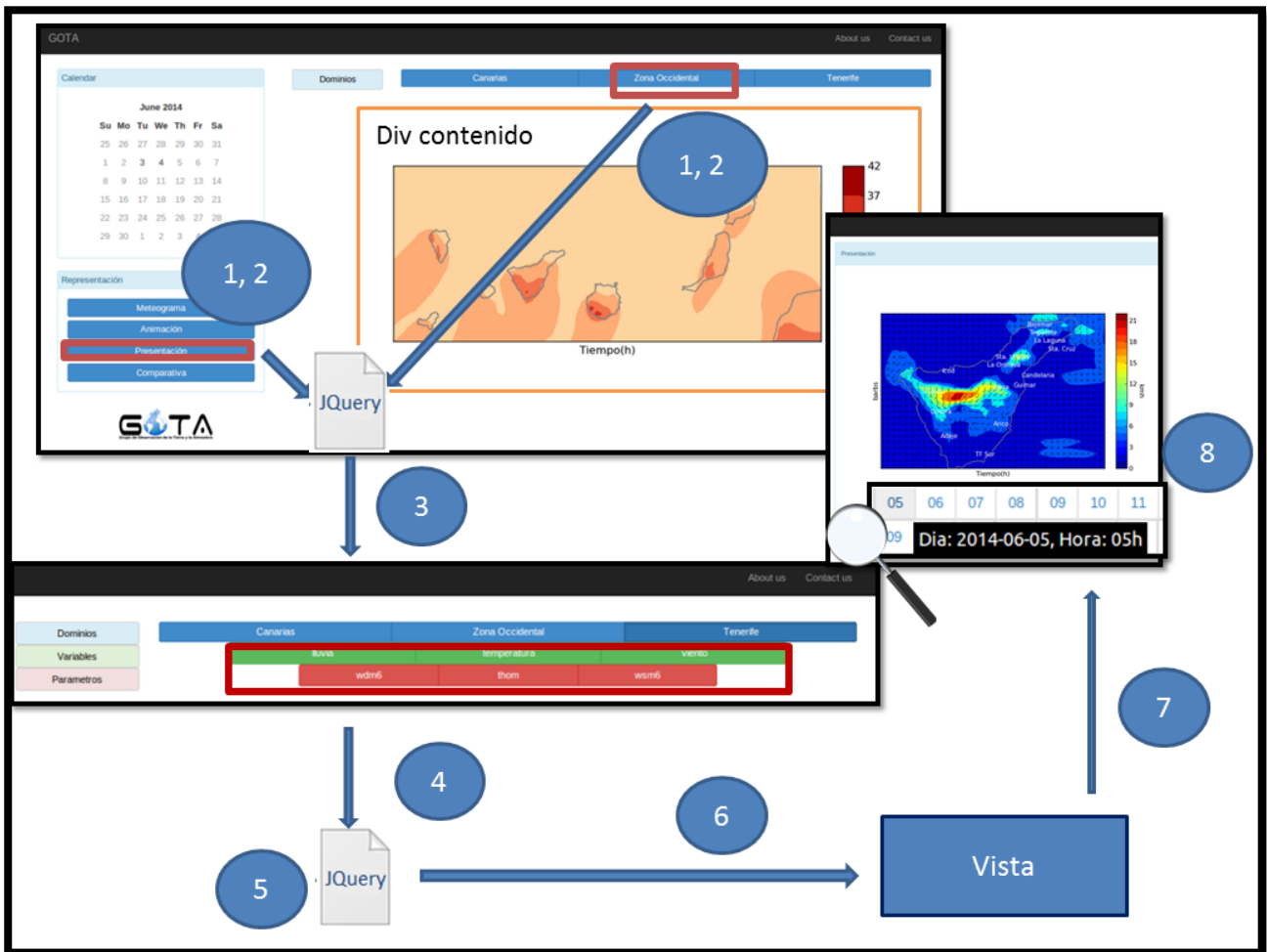


Figura 14. Presentación.

Comenzaremos seleccionando *presentación* y *dominio*, no importa el orden en el que lo hagamos. JQuery evaluará su valor, si es null mostrará un mensaje de error, si no, se desplegarán dos menús uno para la elección del parámetro y otro para la elección de la variable. JQuery comprobará que estas variables no sean null ni sean indefinidas. A partir de los valores obtenidos en cada una de ellas JQuery enviará la información obtenida a la vista, en ella se hallarán los días y horas de la representación y se mostrarán de forma estática para el mejor estudio de las simulaciones. Podemos elegir la hora y el día de la simulación, al posicionarnos encima de una de las horas del menú.

### 3.3.4 COMPARATIVA GOTA VS AEMET.

Con esta opción se elegirá la variable, el dominio y la estación meteorológica con la que queremos comparar los resultados de las simulaciones.

La comparación se hará entre la predicción hecha por GOTA el día anterior y con la recopilación de los datos observacionales de AEMET de ese mismo día (últimas 24 horas). Se elegirán automáticamente las horas en común de ambos y se representarán los resultados.

El objetivo de esta comparación es observar si las predicciones realizadas por el grupo GOTA se adecuan a los datos observacionales, es decir a las variables atmosféricas que ya han sucedido y así mostrar y comprobar su precisión en las predicciones de las mismas.

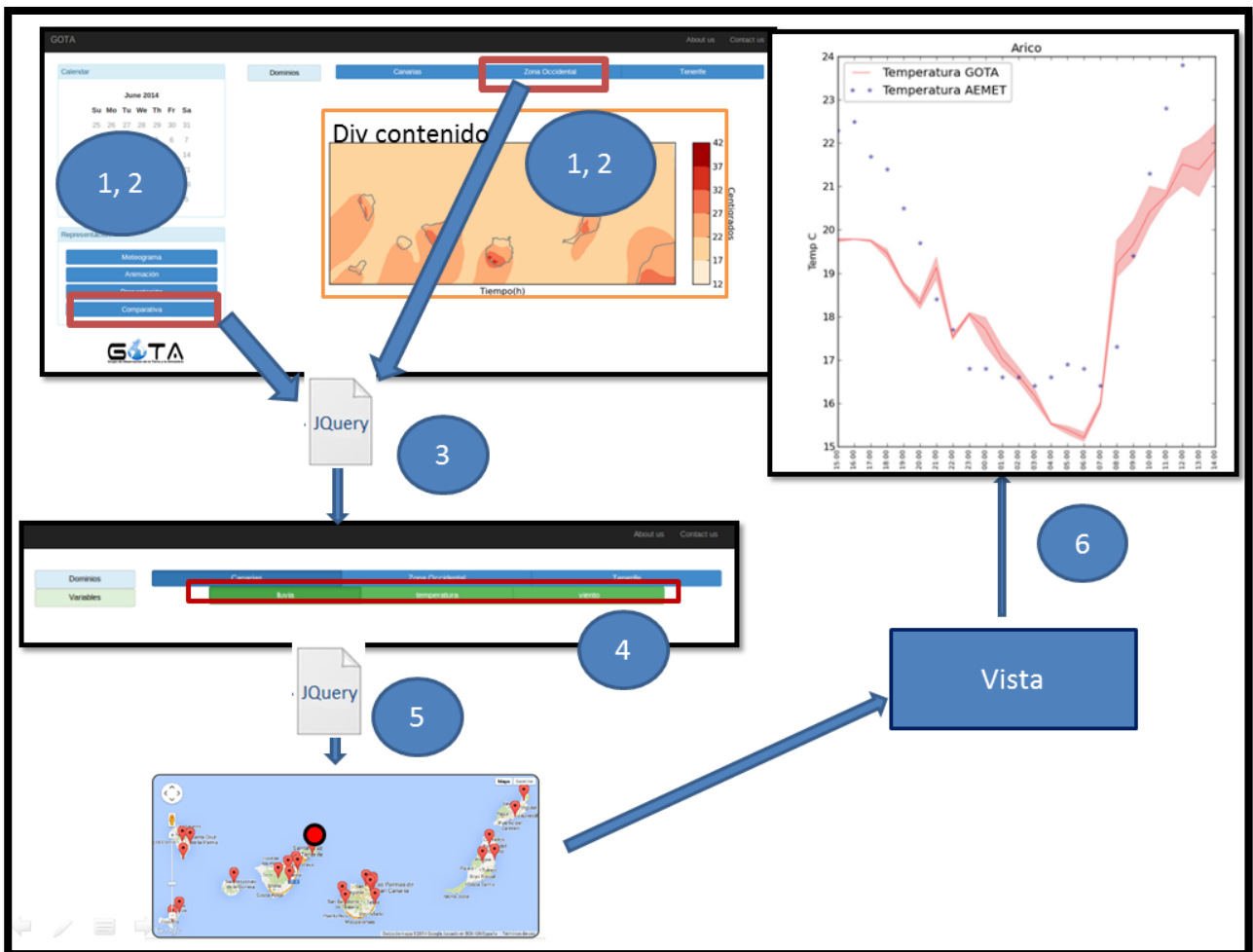


Figura 15. Comparativa.

Seleccionaremos presentación y dominio, no importa el orden en el que lo hagamos. JQuery evaluará su valor, si es null o indefinido mostrará un mensaje de error, sino se desplegarán un menú para la elección de la variable. JQuery comprobará que ésta no sea null ni sea indefinida. Se elegirá una estación meteorológica de AEMET previamente representada en un mapa geolocalizado. A partir del valor obtenido JQuery enviará la información a la vista, en ella se generará una gráfica comparativa entre la predicción de GOTA y los datos observacionales de AEMET. En la vista se seleccionan los días en común entre los datos observacionales y la predicción para generar la comparativa. Se obtendrá el nombre de la estación AEMET y finalmente se mostrará visualmente la gráfica resultante.

### HERRAMIENTAS.

A continuación, se detallarán las herramientas utilizadas durante el desarrollo del trabajo de final de grado.

Listaremos las herramientas utilizadas y su finalidad. Las dividiremos en tres grandes grupos de herramientas: herramientas de tratamiento de datos, herramientas para la gestión de la información y herramientas de aplicación web.

#### 4.1 HERRAMIENTAS DE TRATAMIENTO DE DATOS.

##### 4.1.1 NETCDF.

NetCDF es el conjunto de bibliotecas de software y de formatos de datos independientes auto-descritos que apoyan la creación, acceso y el intercambio de datos científicos [5].

Utilizamos estas librerías para tener acceso a los archivos generados por WRF y para poder tratarlos. Como anteriormente comentamos los archivos WRF son muy completos y complejos que ocupan demasiado espacio, para solucionar este problema hemos seleccionado las variables con las que principalmente trabajaremos y las hemos exportado a un tipo de fichero más manejable, en este caso ha sido .npz, un formato utilizado por el módulo *numpy* de Python.

Para esta exportación necesitamos la librería PyNio que se encargará de la lectura y transformación de los datos. Esta librería permite la manipulación de ficheros NetCDF

##### 4.1.2 PYNIO.

PyNio es un paquete que nos da acceso a la escritura y lectura de ficheros NetCDF y Grib. Éste está compuesto por diferentes librerías (netCDF3-4, GRIB1-2, HDF4, HDF-EOS2, HDF-EOS5, CCM, OGR) que le proporcionan mayor campo de trabajo al poder escribir y leer en diferentes formatos científicos para el mejor acceso a la información obtenida [6].

En este trabajo se hace uso de estas librerías para la manipulación de los archivos netcdf.

### 4.1.3 NUMPY.

Numpy es un módulo open-source añadido a Python que proporciona modelos matemáticos y numéricos pre-compilados, y funciones rápidas. Numpy (Numeric Python) nos proporciona rutinas básicas para manipular grandes arrays y matrices de números. [7]

### 4.1.4 MATPLOTLIB.

Matplotlib es una librería de Python dedicada a la representación gráfica. Está basada en la herramienta matemática MATLAB.

Matplotlib es una herramienta cómoda y sencilla de tratamiento de datos para su posterior representación, con ella podemos generar gráficos, histogramas, espectros de potencia, graficas de barras, etc. [8]

#### 4.1.4.1 BASEMAP.

Como bien comentamos anteriormente Matplotlib nos permite realizar la representación gráfica de nuestros datos ya sea en un histograma, gráfica o incluso en un mapa.

Principalmente, generamos imágenes georreferenciadas con los datos de viento, lluvia y temperatura para cada región. Para ello definimos un mapa “basemap” para cada una de ellas. El mapa basemap nos permitirá geolocalizar los datos recogidos con anterioridad. Un ejemplo de configuración de las variables que definen el mapa basemap es el siguiente:

#### Zona de Tenerife.

```
m = bm.Basemap(llcrnrlon = -17,  
               llcrnrlat = 28,  
               urcrnrlon = -16,  
               urcrnrlat = 28.6,  
               projection = lcc,  
               lat_0= 28,  
               lon_0= -17,  
               resolution='h')
```

Figura 16. Configuración Basemap para la zona de Tenerife.



A continuación, describiremos algunos aspectos que hay que tener en cuenta a la hora de representar las diferentes variables meteorológicas.

- **Temperatura.**

Se generan imágenes teniendo en cuenta la hora de inicio de las simulaciones, la duración total y las horas descartadas.

Se pintará el relleno del mapa con la variable “temp” a la que se le restará 273.15 para la conversión de Kelvin a Celsius. Se definirán la máxima y mínima para su representación. Además, a la derecha de la imagen incorporaremos una leyenda con el código de colores utilizado para su creación. Por último se superponen las costas de la zona a través de una línea de color gris para facilitar su visualización.

- **Viento.**

En la imagen que generaremos del viento representaremos los diagramas de barbas para representar el módulo y la dirección del viento.

Además de las barbas, se representa en colores el módulo del viento previa conversión a km/h.

Con el objetivo de mostrar de forma clara las barbas las representamos de tres en tres píxeles para que así no se solapen unas con las otras.

- **Lluvia y Nubes.**

Hemos de sumar las variables RAINC y RAINNC generadas por WRF, que en nuestro código denominamos lluvia y lluvia2. Además debemos tener en cuenta que el modelo proporciona esta variable de forma acumulada desde el inicio de la simulación.

Por otra parte, se superpone la cobertura nubosa en escala de grises en función de la proporción de pixel cubierto. Esta cobertura nubosa se genera a partir de las fracciones nubosas que el modelo genera para cada capa vertical de la atmósfera utilizando la técnica “maximum random overlap”.

#### **4.1.4.2 GENERACIÓN DE GRÁFICAS.**

Como comentamos con anterioridad hemos de representar meteogramas y gráficas comparativas entre AEMET y GOTA. Para ello hemos de crear gráficas capaces de mostrar los datos directamente en la aplicación Django. Para ello hacemos uso de FigureCanvas.

Este método nos permite crear un canvas en el cual se mostrará nuestra imagen sin la necesidad de estar guardándola cada vez que accedamos a ella.

## **4.2 GESTIÓN DE LA INFORMACIÓN.**

### **4.2.1 SQLITE.**

Utilizaremos la base de datos SQLite para guardar los datos relevantes a las simulaciones realizadas, siendo la base de datos actualizada por el demonio. La base de datos ha de ser creada previamente antes de su utilización.

### **4.2.2 JSON.**

JSON (JavaScript Object Notation) es un formato de intercambio de datos ligero, de fácil lectura para el usuario y sencillo de parsear y generar por un ordenador [9]. Por tal motivo hemos utilizado esta tecnología para el intercambio de información entre las distintas partes que conforman la aplicación.

Los archivos JSON empleados a lo largo del proyecto han sido parseados y generados desde archivos Python.

## **4.3 APLICACIÓN WEB.**

### **4.3.1 DJANGO.**

Django framework desarrollado en Python basado en el MVT (Modelo Vista Template).

En él desarrollamos las vistas necesarias para la generación de las gráficas comparativas con los datos observacionales y los meteogramas. Además, una vista que se encargará de la recolección de las imágenes para su posterior visualización en la web.

Una vez las vistas están creadas se desarrollan las páginas HTML donde se mostrarán sus resultados finales.

### **4.3.2 HTML.**

Hemos utilizado HTML5 como lenguaje marcado.

### **4.3.3 CSS.**

En lo referente a las hojas de estilo hemos utilizado CSS, para generar la apariencia que mejor se adecúe a nuestra aplicación.

CSS (Cascading Style Sheets), es una herramienta que permite definir cómo se mostrarán los elementos en la pantalla. Se utilizan para dar estilos a páginas HTML y XML. Cualquier cambio realizado en un fichero CSS afectará a todas aquellas páginas que lo utilicen [10].

### **4.3.4 BOOTSTRAP.**

Bootstrap framework de HTML, CSS y JS para desarrollo adaptable para cualquier dispositivo. Utiliza los preprocesadores de CSS, Sass y Less [11].

Bootstrap tiene un diseño originalmente inspirado en Twitter. Lo hemos utilizado por su facilidad y claridad; además, se basa en un sencillo sistema grid que nos permite definir de forma asequible las partes de nuestra página.

### **4.3.5 JAVASCRIPT Y JQUERY.**

JavaScript es un lenguaje interpretado, orientado a objetos; éste es un dialecto del estándar ECMAScript. Permite la construcción de objetos en tiempo de ejecución, la creación de scripts dinámicos mediante el uso de eval, la introspección de objetos mediante el uso de for ... in, ... [12]

jQuery es una biblioteca de JavaScript rápida, pequeña y rica en funciones. Se ejecuta de forma continua tal y como lo hacen los ficheros de HTML. Esta se encarga de que la manipulación, manejo de eventos, animación y Ajax sean más simple [13].

Por tales motivos se han implementado en la plataforma web para conseguir efectos de muestra, ocultación de zonas de código, redirección a urls, creación y concatenación de código HTML al HTML inicial, la creación de funciones según el comportamiento de un elemento, ...



### 4.3.8 URLLIB2.

Este módulo de Python permite acceder a url externas [16]. En nuestro caso utilizamos este modelo para acceder a los JSON creados en el cluster y el demonio. A través, de este comando somos capaces de acceder a la url que nos brinda de JSON.

### 4.3.9 PYTHON MINIMAL DOM.

Implementa un modelo DOM más sencillo, pero igual de completo. Contiene una función parse que toma el nombre del archivo u objeto abierto. Esta función modifica el alimentador de documentos del analizador y activa el soporte de espacio de nombres [17].

En nuestro caso usamos miniDOM y la función parse para procesar el archivo JSON que son abiertos con anterioridad por urllib2 y que se encargan de conseguir el nombre del lugar más cercano al punto que hemos elegido en un mapa de google maps según la latitud y longitud que obtenga.

Una vez se ha parseado el archivo se obtendrán todos los elementos que se correspondan con el tag 'name' y se guardarán.

DOM (Document Object Model) es una API que proporciona un conjunto estándar de objetos para representar documentos HTML y XML. A través de ella se puede acceder y modificar el contenido, estructura y estilo de dichos documentos [25].

### 4.3.10 PYTHON JSON.

Permite manipular ficheros JSON tanto crearlos como parsearlos.

Ejemplo de lectura de JSON.

1. `datos = urllib2.urlopen('url.json').read()` | Abrir fichero.
2. `obj = json.loads(datos)` | Convertirlo al formato JSON.
3. `fecha = obj['wrfdata']['fecha']` | Leer datos de JSON.

Ejemplo de creación de un fichero JSON:

1. Abrir un fichero en modo de escritura.

2. `json.dumps( datos_a_parsear )`
3. Cerrar fichero.

#### **4.3.11 PYTHON DATETIME.**

Este módulo suministra las clases necesarias para manipular fechas y horas. Se permite introducir y pasar a *datetime* las fechas y horas de forma aritmética y viceversa [18]. Con esto podremos manipular las fechas y obtener las deseadas.

Uno de los casos en los que utilizamos este módulo es para comprobar que las fechas de los archivos de imágenes o predicciones en el *json* no sean de más de una semana a partir de la última simulación. Convertimos la fecha de formato aritmético a *datetime* y le restaremos “`timedelta(days=X)`”, con ello obtenemos la fecha de hace una semana. Gracias a la fecha obtenida compararemos los archivos o predicciones realizadas anteriormente a esa determinada fecha y serán eliminados.

#### **4.3.12 PYTHON RE.**

Este módulo de Python nos provee de un comparador de expresiones regulares [19].

#### **4.3.13 PYTHON OS.**

El módulo OS es la manera portable de realizar operaciones del sistema operativo desde Python [20].

#### **4.3.14 PYTHON GLOB.**

El módulo glob permite encontrar las direcciones que concuerden con los términos especificados y permite crear una lista de estos [21].

#### **4.3.15 PYTHON SHUTIL.**

Ofrece un gran número de operaciones a alto nivel en ficheros o en directorios. En particular se trata de funciones que se encargan de proporcionar soporte de copia y borrado de directorios y árboles de directorios. En cambio cuando tratamos de borrar un archivo hemos de utilizar el modulo “`os.remove()`” [22].

Los métodos `shutil`, `os.remove()` y `glob` los hemos utilizado para listar los ficheros que no deseamos y borrarlos. Si son directorios será a través de `shutil` tal y como hemos comentado o si son ficheros a través de `os.remove()`.

#### **4.3.16 PYTHON CONFIG PARSER.**

Este módulo define la clase `ConfigParse` que implementa un lenguaje que analiza los archivos de configuración básica como los archivos de Microsoft `.INI` [23].

Utilizamos este método para leer el fichero de configuración del demonio.

#### **4.3.17 JSON GEONAMES.**

GeoNames es una base de datos geográfica que contiene más de 10 millones de nombres geográficos, está disponible para su descarga gratuita bajo una licencia Creative Commons. [24].

### RESULTADOS.

A continuación, se mostrarán los resultados obtenidos durante el desarrollo del trabajo de final de grado.

Se mostrarán algunos ejemplos de las funcionalidades implementadas en la plataforma, como son la presentación de mapas, meteorogramas o las herramientas de diagnóstico. Además, mostraremos el análisis jerárquico de la aplicación, en el que se indica el objetivo de cada apartado y la representación gráfica de los datos obtenidos.

#### 5.1 HERRAMIENTAS DE REPRESENTACIÓN Y DIFUSIÓN.

Las herramientas de representación y difusión nos permiten mostrar gráficamente los datos recogidos para cada una de las variables atmosféricas estudiadas a lo largo de este proyecto (temperatura, viento, lluvia y nubes), para cada parametrización (wdm6, thom, wsm6) y por cada dominio (Canarias (región 2), Zona Occidental (región 3) y Tenerife (región 4)).

A continuación se mostrarán algunos ejemplos de resultados obtenidos:

1. Si seleccionamos la variable temperatura con la parametrización Thom en la zona de Tenerife obtenemos la siguiente figura.

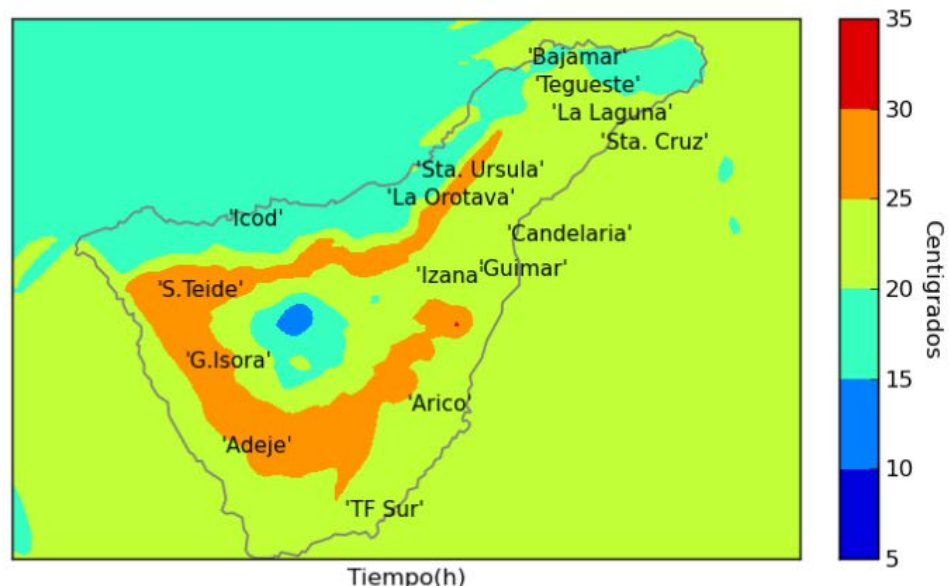


Figura 18. Temperatura a 2 metros para la Isla de Tenerife.

En ella se puede observar, para una hora determinada, la variación de la temperatura, en



grados centígrados, en las diferentes zonas de Tenerife. Además, se muestra la paleta de colores que permite asociar el color con el valor numérico de temperatura.

2. En cambio, si para la misma fecha seleccionamos la variable viento para la parametrización Wdm6 en la región de Canarias, se obtiene lo que se muestra en la figura 19.

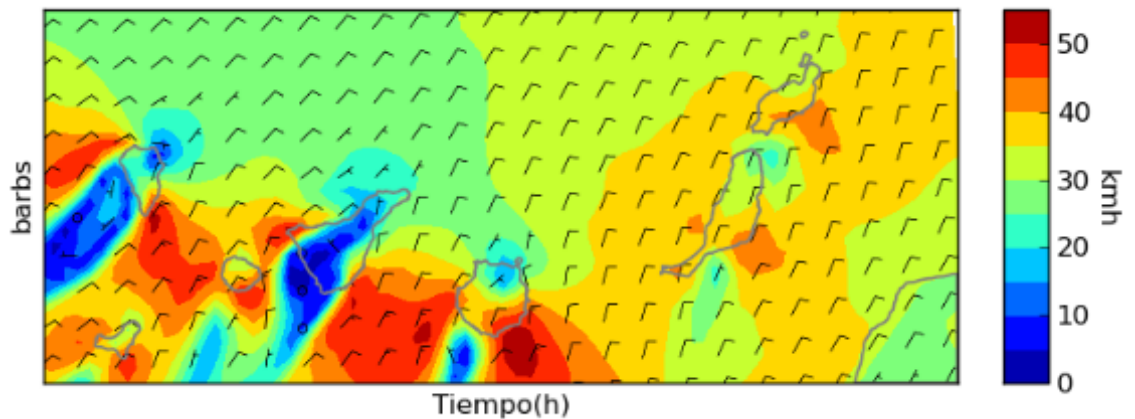


Figura 19. Viento.

Podemos observar la intensidad del viento en km/h atendiendo a la escala de color de la leyenda y al número y tamaño de las barbas (notación estándar). También, podremos conocer la dirección (orientación de las barbas).

3. A continuación se muestra en la figura 1.9 el producto de la selección de las variables lluvia y nubes con una parametrización wsm6 en el dominio Zona Occidental.

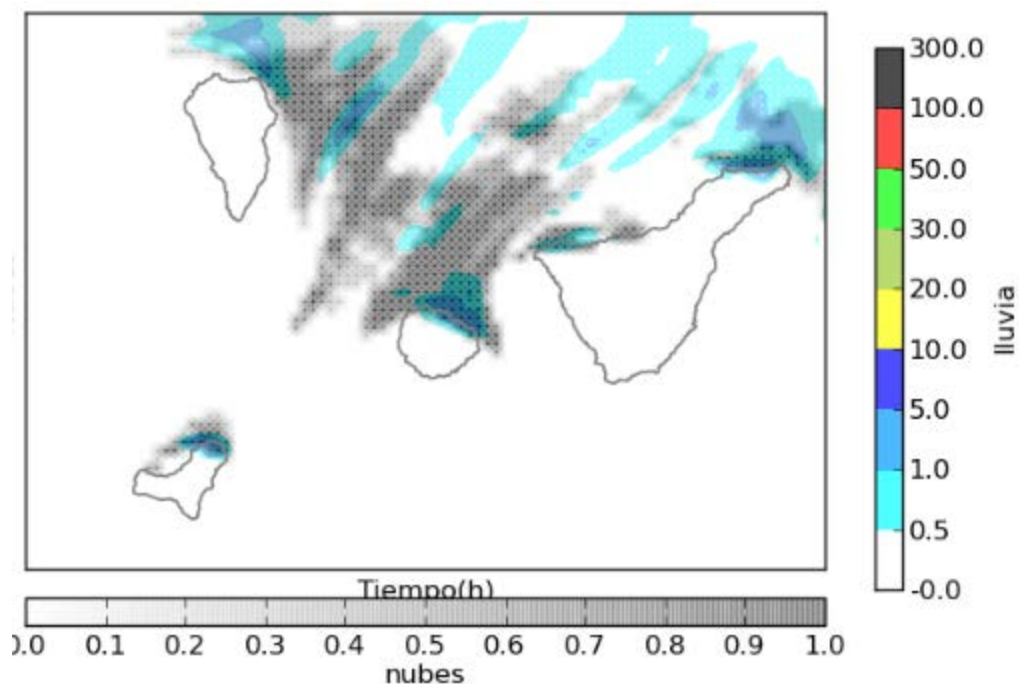


Figura 20. Lluvia.

En ella podemos contemplar las nubes (en escala de grises) y las lluvias para una hora determinada. Para su representación, se calcula para cada pixel de la imagen la cobertura efectiva de nube a partir de los datos de fracción nubosa de los distintos niveles atmosféricos en los que WRF genera esta variable. Además, representamos la lluvia usando una escala de colores que coincide con la paleta que utiliza la AEMET para la representación de esta variable en su portal web. Para que no se superpongan los datos de las nubes y la lluvia hemos aplicado unos criterios de transparencia (lluvias 0.7 y nubes de 0.3) por lo que se visualizarán ambas sin ninguna dificultad.

## 5.2 HERRAMIENTAS DE DIAGNÓSTICO.

La herramienta de diagnóstico es la comparativa de las predicciones de GOTA y los datos observacionales obtenidos por la AEMET. Con esto podremos comprobar la precisión de las distintas simulaciones.

En las Islas Canarias disponemos de 62 estaciones de la AEMET que, con diferente frecuencia, registran las principales variables meteorológicas. La información obtenida en cada una de ellas puede ser consultada on-line en la página web de la AEMET ([www.aemet.es](http://www.aemet.es)).

Para realizar la comparativa, hemos de calcular los días y horas en común entre las predicciones generadas por GOTA y los datos observacionales de la AEMET. Posteriormente extraemos los datos para esos instantes de tiempo y generamos la correspondiente comparativa.

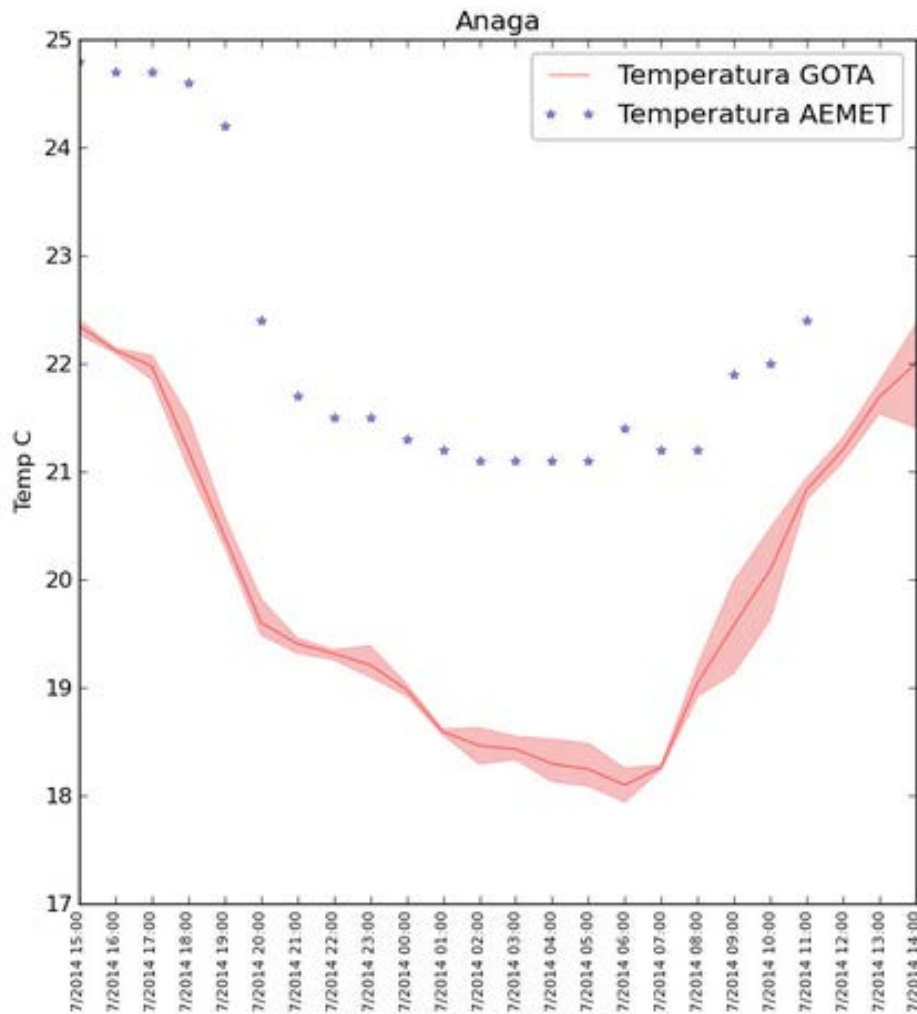


Figura 21. Resultado comparativa.

Los datos representados por la AEMET y por GOTA pueden no coincidir. En el caso de temperatura, esto se debe principalmente a la diferencia de altura entre el punto del modelo y la estación meteorológica debido a la alta complejidad orográfica de las Islas. Por ejemplo, en el caso de la figura 21, correspondiente a la estación de Anaga, ésta se encuentra a 19 metros sobre el nivel del mar, pero el pixel de modelo para esa ubicación, se encuentra a 160 metros, lo que justifica que la temperatura proporcionada por el modelo es inferior a la medida por la estación.

### 5.3 METEOGRAMA.

En el meteograma se podrá observar la evolución temporal de cada una de las variables, para todas las horas de predicción a partir de periodo spin-up. Se representan los máximos, mínimos y media de la lluvia y temperatura para las distintas simulaciones realizadas

(parametrizaciones). Además, se muestra en la parte superior la dirección y velocidad del viento mediante diagramas de barbas de viento.

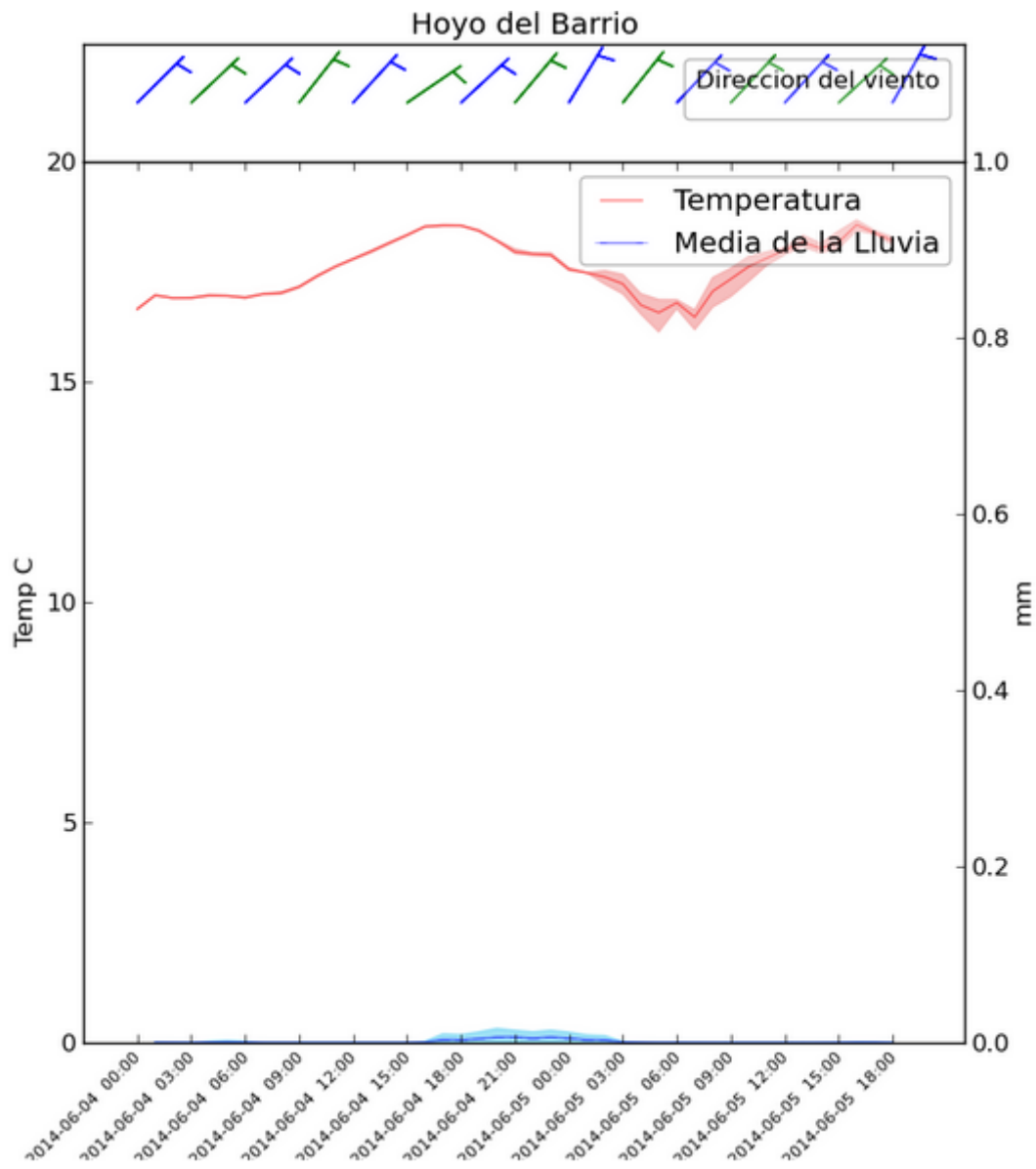


Figura 22. Resultado meteograma.

## 5.4 ANÁLISIS JERÁRQUICO.

En el análisis jerárquico definimos la representación gráfica de la descomposición de la estructura de la aplicación web. A través de ella se muestra el uso lógico de la aplicación, su

funcionalidad y flexibilidad para cada componente, puesto que en un futuro podríamos ampliar las variables, las parametrizaciones y los dominios.

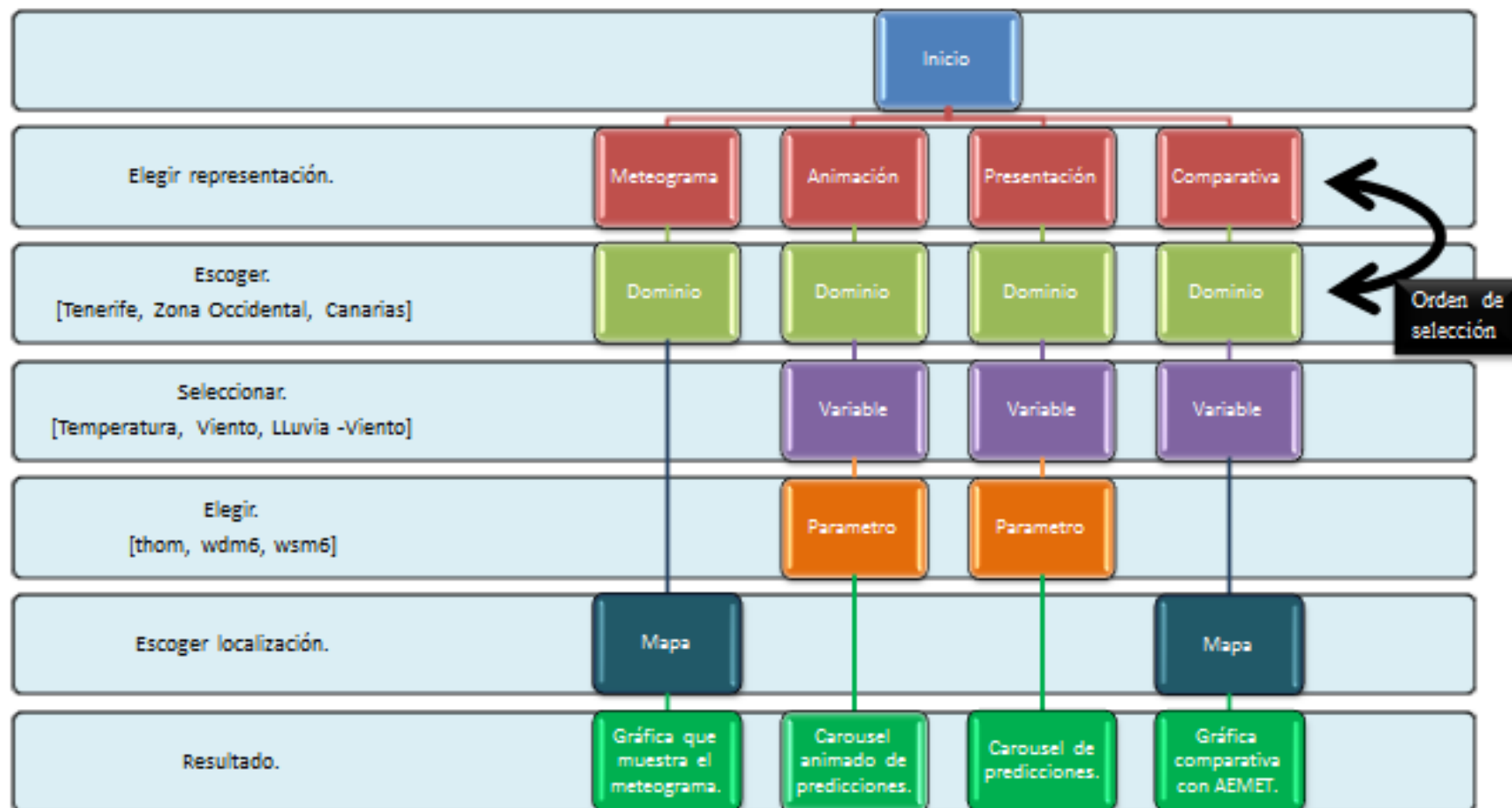


Figura 23. Análisis Jerárquico.

## Conclusiones y trabajos futuros.

Este proyecto de final de grado ha tenido como resultado la creación de un portal web amigable y sencillo que permite obtener información de las predicciones meteorológicas realizadas por el grupo GOTA y realizar una comparativa de las predicciones realizadas con los datos observacionales de la AEMET.

Para poder llevar a cabo este proyecto comenzamos estudiando los modelos numéricos de predicción meteorológica y el porqué era necesario realizarlos a escala local en una región como la canaria. Seguidamente, comprendimos el funcionamiento de las herramientas científicas como WRF y cómo tratar sus datos multidimensionales y transformarlos a un fichero más manejable como NPZ.

Tras comprender el contenido de las variables que seleccionamos, pasamos a mostrar sus datos mediante el uso de librerías de representación como matplotlib.

Creamos demonios tanto en el cluster como en la propia aplicación web para la recogida y el tratamiento de datos. La información que nos prestan ambos demonios es utilizada en la aplicación web para mostrar las imágenes de las predicciones realizadas, creación dinámica de meteogramas, y realización de comparativas.

Todo este proceso nos ha permitido ampliar nuestros conocimientos en meteorología, a relacionarnos con nuevas herramientas tales como matplotlib, JQuery, Python, BeautifulSoup4, entre otras, conocer mejor aquellas herramientas que ya habíamos utilizado en el pasado tales como Bootstrap, Django, API Google Maps, HTML5, ...

Durante este proceso se nos han presentado diversas dificultades derivadas al desconocimiento de todo el potencial de las herramientas utilizadas. Un caso destacable son las herramientas JavaScript y JQuery herramientas que no conocíamos en su totalidad.

El presente Trabajo Final de Grado ha sido un proyecto enriquecedor, ha proporcionado y reforzado nuevos conocimientos.

Finalmente, las líneas de trabajo futuras son: la realización de test para poner a prueba del funcionamiento de la aplicación web, generar comparativas con otras estaciones meteorológicas, mostrar las predicciones de otras variables atmosféricas, generación de perfiles verticales para las variables 4D, integrar otras redes de datos observacionales o información de satélites.





## Conclusions.

In this work we have developed a friendly and simple web application that allows us to display the forecast information that the GOTA performed in an operational way. Also, this tool is able to make comparative graphs to evaluate the forecast accuracy comparing model outputs with observational data from the AEMET weather stations.

To make this project possible we started studying the weather numerical models and their necessity to perform accurate atmospheric forecast at a local scale in an orographically complex region as Canary Islands. After that, we understood how the scientific tools were working and how to treat with their data outputs and to transform them into smaller files, which are easier to manage, like the NPZ format.

After understanding the contents of the variables that we have selected, we started to show the results using visual tools like the representation library matplotlib.

We created two daemons, one in the cluster and other in the web application itself, with the aim of gathering and processing that information. With the results provided by these daemons we show the images of the forecasting, and we provide the tool to dynamically create a meteogram and to generate comparatives in the application web.

All this process allows us to increase our meteorological knowledge, to be involved with new tools like matplotlib, JQuery, Python, BeautifulSoup4, among others, and to improve our skills with those tools which have been used previously such as Bootstrap, Django, Google Maps API, HTML5, ...

During this process we have to treat with different problems, consequence of our lack of knowledge about the potential of the tools that we used. A remarkable case are the JavaScript and JQuery tools, that we had used only in basic developments.

This final degree project have been an enriching work, that has provided us new knowledge and reinforced some concepts and skills.

Finally, the future work lines are the followings: to make a test bed to check if the web application runs properly under diverse situations, to produce comparatives with other meteorological stations, to create forecast displays with other atmospheric variables, to generate vertical profiles to the 4D variables, and to integrate other observational databases or satellite information.



## Bibliografía.

"introduction to climate models" de [www.meted.ucar.edu](http://www.meted.ucar.edu).

Imágenes <http://pixabay.com/>

[1]

[http://es.wikipedia.org/wiki/Meteorolog%C3%ADa#Historia\\_de\\_la\\_meteorolog.C3.ADa](http://es.wikipedia.org/wiki/Meteorolog%C3%ADa#Historia_de_la_meteorolog.C3.ADa)

[2]

[http://es.wikipedia.org/wiki/Modelo\\_num%C3%A9rico\\_de\\_predicci%C3%B3n\\_meteorol%C3%B3gica](http://es.wikipedia.org/wiki/Modelo_num%C3%A9rico_de_predicci%C3%B3n_meteorol%C3%B3gica)

[3] <http://en.wikipedia.org/wiki/Downscaling>

[4] <http://www.wrf-model.org/index.php>

[5] [http://www.unidata.ucar.edu/software/netcdf/docs/index.html#mainpage\\_netcdf](http://www.unidata.ucar.edu/software/netcdf/docs/index.html#mainpage_netcdf)

[6] <https://www.pyngl.ucar.edu/Nio.shtml>

[7] <http://www.engr.ucsb.edu/~shell/che210d/numpy.pdf>

[8] <http://matplotlib.org/>

[9] <http://json.org/>

[10] <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>

[11] <http://getbootstrap.com/>

[12] <https://developer.mozilla.org/es/docs/JavaScript>

[13] <http://jquery.com/>

[14] <https://developers.google.com/maps/documentation/javascript/?hl=es>

[15] <http://www.pythonforbeginners.com/python-on-the-web/beautifulsoup-4-python/>

[16] <https://docs.python.org/2/library/urllib2.html>

[17] <https://docs.python.org/2/library/xml.dom.minidom.html>

[18] <https://docs.python.org/2/library/datetime.html>

[19] <https://docs.python.org/2/library/re.html>

[20] <https://docs.python.org/2/library/os.html>

[21] <https://docs.python.org/2/library/glob.html>

[22] <https://docs.python.org/2/library/shutil.html#module-shutil>

[23] <https://docs.python.org/2/library/configparser.html>

[24] <http://www.geonames.org/>

[25] [http://es.wikipedia.org/wiki/Document\\_Object\\_Model](http://es.wikipedia.org/wiki/Document_Object_Model)

[26] <http://consulclima.co.uk/climate-modelling/downscaling-gcm-outputs/#downscaling>