



UNIVERSIDAD DE JAÉN
Centro de Estudios de Postgrado

Trabajo Fin de Máster

**EVALUACIÓN DEL RECURSO
EÓLICO EN SIERRA DEL TRIGO
(JAÉN) A PARTIR DE DATOS DE
VIENTO MEDIDOS Y
PROCEDENTES DEL REANÁLISIS
METEOROLÓGICO ERA-5**

Alumno/a: Castro Yanayaco, Annel Marisol

Tutor/a: Prof. D. Antonio David Pozo Vázquez
Dpto: Departamento de Física

Diciembre, 2021

AGRADECIMIENTOS

Quiero dar las gracias a todas las personas que me han apoyado y motivado a lo largo del máster. El curso 2020-2021 ha sido un curso diferente, marcado por la pandemia, sin embargo, eso no ha sido impedimento para seguir adquiriendo conocimientos y conocer gente nueva.

Gracias a mi familia y amigos, por ser un pilar fundamental en mi vida personal y por ayudarme a motivarme cuando más lo necesitaba.

Gracias a todos los profesores del Máster en Energías Renovables, ya que, sin su dedicación, esfuerzo y paciencia durante a las circunstancias vividas, no hubiera sido posible adquirir los conocimientos impartidos. En especial al profesor David Pozo Vázquez, no solo por su labor como tutor y guía en este trabajo, sino también por ser quien impartió las clases en el grado que me motivaron a elegir este máster.

También quiero dar las gracias a los chicos del despacho, ellos fueron un pilar muy importante en la realización del trabajo. Gracias por su paciencia y por su tiempo al enseñarme a programar, en especial a Miguel y Antonio.

Finalmente, quiero agradecer a los propietarios de los datos de Sierra de Trigo por la cesión de los datos.

RESUMEN

La energía eólica va a experimentar un acusado crecimiento en las próximas décadas. En particular, se espera que juegue un papel central en los sistemas eléctricos de muchos países, en el marco del desarrollo de sistemas energéticos bajos en emisión de carbono. De cara al desarrollo de estos sistemas, es de vital importancia el estudio de la variabilidad temporal y espacial del recurso eólico. Los reanálisis meteorológicos son una herramienta fundamental para llevar a cabo este tipo de análisis. Sin embargo, requieren de una evaluación previa con medidas de tierra, para ver la representatividad de los mismos. El objetivo principal de este trabajo es la evaluación de los datos de viento del reanálisis ERA-5 para su uso en la evaluación del potencial eólico en España, en particular, en zonas montañosas. Para este fin se dispone de datos de viento medidos en una Torre meteorológica situada en Sierra del Trigo (Jaén). Se han llevado a cabo dos análisis. En el primero, se compararon los datos viento de ERA-5 y de la Torre para una altura de 100 metros. En el segundo, se evaluó la diferencia en la generación de energía eólica de varios aerogeneradores representativos a partir de ambas fuentes de datos. Los resultados muestran, en primer lugar, una muy importante diferencia entre las distribuciones de vientos. Así, la mediana de los datos de Torre es de 5,16 m/s mientras que la de ERA-5 es de 2,5 m/s. Las diferencias en los vientos extremos son también muy acusadas. Estas diferencias dan lugar a valores muy diferentes en la generación de energía eólica. Así, los factores de ocupación de los aerogeneradores obtenidos a partir de los datos de Torre llegan a ser 5 veces mayores que los derivados de los datos de ERA-5. A modo de conclusión general, cabe decir que los datos de ERA-5 no pueden ser utilizados de forma directa para el análisis del recurso eólico en emplazamientos como el estudiado, con una alta complejidad topográfica. La razón es que, debido a la baja resolución espacial del modelo (30 km), los efectos que la topografía tiene en la modificación del viento a escala local (efecto valle o efecto colina) no quedan representados en los datos ERA-5. Esto supone un grave problema en regiones como España, donde el recurso eólico está asociado a dichos efectos topográficos.

Abstract

Wind energy is going to experience marked growth in the coming decades. In particular, it is expected to play a central role in low-carbon power systems. With regard to the development of these systems, the study of the temporal and spatial variability of the wind resources is of vital importance. Meteorological reanalyses are a fundamental tool to carry out this type of analysis. However, they require a prior evaluation with ground measurements, to evaluate their representativeness. The main objective of this work is to evaluate the reliability of the ERA-5 reanalysis wind data in order to be used for the analysis of the wind energy resources in Spain, particularly in mountainous areas. For this purpose, wind data measured in a meteorological tower located in Sierra del Trigo (Jaen) are used. Two analyses have been carried out. In the first, the wind data from ERA-5 and from the Tower were compared for a height of 100 meters. In the second, the difference in wind power generation of several representative wind turbines, computed with both data bases, were evaluated. The results show, first of all, a very important difference between the wind distributions. Notably, the median of the tower data is 5.16 m/s while that of ERA-5 is 2.5 m/s. The differences in extreme winds are also very marked. These differences resulted in very different values in wind power generation. Notably, the capacity factors of the wind turbines obtained using the tower data were up to 5 times higher than those derived from the ERA-5 data. As a general conclusion, the ERA-5 data cannot be used directly for the analysis of the wind resource in sites such as the one studied, with a high topographic complexity. The reason is that, due to the low spatial resolution of the reanalysis model (30 km), the effects that the topography has on the modification of the wind at a local scale (valley effect, hill effect) are not represented in the ERA-5 data. This is a serious problem in regions such as Spain, where the wind resource is associated with these topographic effects.

ÍNDICE

1. INTRODUCCIÓN	6
1.1 Variabilidad espacial y temporal del recurso eólico	7
1.2 Utilización de datos de reanálisis para el estudio del recurso eólico	8
1.3. El reanálisis ERA-5	9
1.4. Objetivos y organización del trabajo	9
2. DATOS	10
2.1 Región de estudio	10
2.2 Datos medidos en torre meteorológica	11
2.3. Datos del reanálisis meteorológico ERA-5	12
2.4. Curvas de potencia de aerogeneradores	12
3. METODOLOGÍA	14
3.1 Tratamiento de los datos medidos en torre	14
3.2 Estimación del viento en altura	14
3.3 Estimación de la energía generada	17
3.4 Estadísticos de evaluación	17
4. RESULTADOS	17
4.1 Análisis de los datos medidos en Torre meteorológica	17
4.2 Resultados de la extrapolación en altura de los datos de Torre	20
4.2.1 Estimación del exponente de cizalladura	20
4.2.2 Estimación del viento a 40 y 100 metros	21
4.3 Comparación entre datos de viento medidos en Torre y datos de ERA5	22
4.4 Comparación entre las estimaciones del recurso eólico con datos medidos en Torre y datos de ERA5	24
5. CONCLUSIONES	26
6. REFERENCIAS	27
Anexos	29

1. INTRODUCCIÓN

La generación de energía supone una de las mayores fuentes de emisiones de CO₂. En el caso particular de Europa representa el 75% de las emisiones de gases de efecto invernadero (PNUMA, 2019). En consecuencia, las principales acciones de mitigación del cambio climático y calentamiento global se basan en la sustitución de las fuentes tradicionales de energía por fuentes renovables. La Unión Europea ha establecido una estrategia para la descarbonización de todo el sistema eléctrico para el año 2050. El objetivo a largo plazo es promover sistemas energéticos bajos en emisión de carbono, incluso 100% renovables.

En particular, en España se ha ratificado en este 2021 la primera Ley de Cambio Climático, estableciendo, al menos, la reducción de las emisiones de gases de efecto invernadero en 2030 un 23% con respecto a 1990 (PNIEC, 2020). Una de las principales medidas contempladas por esta ley es la presencia de energías renovables, esperando que el uso final de la energía sea de procedencia renovable en un 42% y que la generación eléctrica cuente con un 74%. Además, se contempla alcanzar la descarbonización del sistema eléctrico en 2050. Para conseguir estos objetivos, en particular el 74% de generación eléctrica de origen renovable, se prevé (Figura 1) un aumento muy importante de la potencia eólica instalada en España para el año 2030, hasta llegar a los 31 GW.

	VALORES DE REFERENCIA (HIDRAULICIDAD MEDIA)	
	DG 2030	ST 2030
Demanda (TWh)	296	285
Capacidad total instalada (MW)	149.439	143.737
Nuclear	7.117	7.117
Carbón	847	4.660
Ciclo combinado	24.560	24.560
Hidráulica (+bombeo)	23.050	23.050
Eólica	31.000	31.000
Fotovoltaica	47.150	40.000
Termosolar	2.300	2.300
Resto renovables	2.550	2.550
Cogeneración y otros	8.500	8.500
Baterías	2.358	0
% RES / Generación	62%	67%
% RES / energía final estimada	29,7%	28,4%
% vertidos energía renovable	2,70%	2,39%
Emisiones kt CO₂	12.593	16.264
Coste variable de generación (€/MWh)	52,0	32,7

Figura 1. Escenarios de desarrollo de la potencia de generación eléctrica a instalar en España en 2030 (Fuente: PNIEC).

A nivel global, se prevé que la energía eólica se convierta, en las próximas décadas, en una de las fuentes principales de energía en casi todos los países del mundo (IRENA, 2021). En particular, se espera la instalación de más de 1200 GW de energía eólica de aquí al año 2030 (Figura 2).

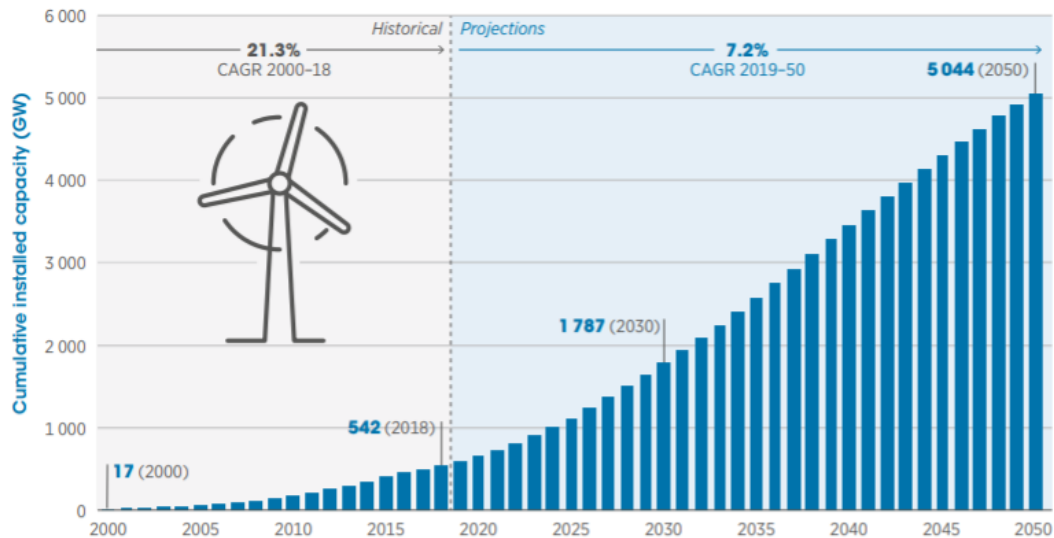


Figura 2. Potencia eólica instalada en el mundo y previsiones de crecimiento, según la organización IRENA (www.irena.org).

1.1 Variabilidad espacial y temporal del recurso eólico

De cara al desarrollo de sistemas eléctricos bajos en carbono, es muy importante estudiar la variabilidad de las fuentes de energía renovable. Dicha variabilidad afecta a los costos (inversiones requeridas, capacidad de transmisión, etc.) así como a la seguridad del sistema eléctrico. Así, el análisis de la variabilidad del recurso solar y eólico es esencial para el modelado y análisis de sistemas energéticos con una proporción significativa de generación a partir de fuentes renovables (Santos-Alamillos et al., 2017). El caso de la energía eólica es especialmente significativo, ya que esta energía muestra gran variabilidad en un amplio rango de escalas temporales que van desde los minutos a las horas, años y décadas (Santos-Alamillos et al., 2017).

Se hace, pues, necesario disponer de bases de datos de estimaciones del recurso eólico, que muestren la variabilidad espacial de dicho recurso y que abarquen el periodo de tiempo suficientemente largo para contemplar la variabilidad interanual del recurso. Se habla así, en este contexto, de mapas eólicos; estos mapas suelen abarcar un periodo de 20 años.

La gran variabilidad espacial del recurso eólico, conjuntamente, con la limitada disponibilidad de medidas de viento en tierra, hacen que el único medio para

obtener estos mapas sea mediante simulaciones con modelos numéricos de predicción meteorológica (NWP por sus siglas en inglés).

1.2 Utilización de datos de reanálisis para el estudio del recurso eólico

Las simulaciones con modelos NWP pueden usarse de una forma muy especial para dar lugar a los llamados reanálisis meteorológicos. Los reanálisis son el resultado de simular durante un periodo de tiempo muy largo y con un único modelo NWP el estado de la atmósfera. Estos reanálisis son las mejores estimaciones del estado de la atmósfera de las que se puede disponer. Los reanálisis tienen la ventaja específica de proporcionar estimaciones coherentes espacial y temporalmente del estado de la atmósfera para las variables atmosféricas, entre ellas el viento (Slivinski et al., 2019). En particular, los reanálisis pueden proporcionar información del viento de gran resolución espacial y temporal, tanto en altura como en superficie, cubriendo amplias regiones (todo el planeta) y a lo largo de periodos de décadas (usualmente abarcan hasta 30 años atrás en el tiempo, 1990-2020)

Debido a estas características, el uso de reanálisis para el análisis del potencial de energías renovables se ha convertido en algo muy común en las últimas décadas. En particular, se ha vuelto una herramienta muy importante en el análisis del potencial eólico (Frank et al., 2020). Existen varios reanálisis globales disponibles, siendo los más importantes el MERRA (MERRA, 2021) y ERA-5 (Hersbach et al., 2018).

A pesar del gran potencial de los reanálisis en el ámbito de las energías renovables, el uso de estas bases de datos no está exento de problemas. En el caso del viento y del recurso eólico, el principal problema es la representatividad de las estimaciones. La resolución espacial de los reanálisis es relativamente baja, del orden de decenas de kilómetros cuadrados. Por ello, en zonas montañosas, que son las de interés en España de cara al desarrollo de parques eólicos, las estimaciones de los reanálisis suelen estar muy sesgadas y ofrecen poca fiabilidad. En general, las estimaciones de la velocidad del viento en estos casos son más pequeñas que la realidad, al no contemplarse la aceleración del viento que se producen zonas montañosas.

Se hace, pues necesario, evaluar y, eventualmente, obtener un modelo de corrección de las estimaciones de los reanálisis para que estos puedan ser usados. El problema radica en que, en el caso de la energía eólica, los datos de interés son las estimaciones a 100 metros de altura. Esta es la altura estándar del buje de los aerogeneradores que se instalan hoy en día. Para poder evaluar y corregir las medidas, deben usarse datos medidos en torres meteorológicas que alcancen esta altura. Sin embargo, existen muy pocas bases de datos de mediciones de torre, de alta calidad, accesibles al público y que tengan registros de largos periodos de tiempo. En Europa existen algunas bases de datos, pero

en España no se dispone de emplazamientos de este tipo con datos públicos. Es por ello que, si bien hay estudios de evaluación de reanálisis en Europa (Sharp et al., 2015), apenas existen estudios de este tipo en España.

1.3. El reanálisis ERA-5

El reanálisis ERA-5 (Hersbach et al., 2018) es el reanálisis más reciente desarrollado por el centro europeo de predicción meteorológica a medio plazo (European Centre for Medium-Range Weather Forecasts, CMWF, siglas en inglés). Este reanálisis proporciona, entre otras variables, estimaciones del viento en superficie (10 metros) y a 100 metros de altura, válidas (potencialmente) para el análisis del recurso eólico. Estas estimaciones están disponibles a escala global, con una resolución espacial de 30 km y temporal de una hora. Actualmente, el reanálisis abarca desde la fecha actual hasta unos 40 años hacia atrás.

1.4. Objetivos y organización del trabajo

El objetivo principal de este trabajo es la evaluación de los datos de viento de ERA-5 para su uso en la evaluación del potencial eólico en España. En particular, se pretende analizar hasta qué punto se podrían usar los datos de ERA-5 para estimar el viento en altura y la energía producida por un aerogenerador en un emplazamiento de montaña en España.

Para este fin se dispone de datos de viento medidos en una torre meteorológica situada en Sierra del Trigo (Jaén), correspondientes al periodo 2019-2020. Utilizando estos datos como “verdad de tierra” se evaluarán, en primer lugar, las estimaciones de la velocidad del viento de ERA-5 a 100 metros de altura sobre el terreno. En segundo lugar, se valorará el error que se cometería en las estimaciones del factor de ocupación y de la energía generada para un conjunto de aerogeneradores representativos.

Los datos a evaluar de ERA-5 son datos de velocidad y dirección del viento a 100 metros de altura sobre el terreno. Sin embargo, los datos de torre disponibles corresponden a medidas del viento a 20 y 40 metros de altura. Por ello, y como paso previo, se llevará a cabo el desarrollo y validación de un modelo de la interpolación del viento en altura que permita estimar la velocidad del viento a 100 metros a partir de las medidas de la torre meteorológica registradas a 20 y 40 metros de altura.

El trabajo se organiza como sigue: en la sección 2 se presenta la región de estudio, así como los datos empleados. En la sección 3 se describe la metodología usada. En la sección 4 se presentan y discuten los resultados y, finalmente, en la sección 5 se presentan las principales conclusiones del trabajo.

2. DATOS

2.1 Región de estudio

La torre meteorológica empleada en este estudio se encuentra en un área de la provincia de Jaén, en concreto en Sierra del Trigo (Figura 3). Sierra del Trigo es un área montañosa que pertenece a las cordilleras subbéticas, cerca de la localidad de Noalejo, en la zona sur de la provincia limitando con la provincia de Granada. Cuenta con una altura máxima de 1649 m.

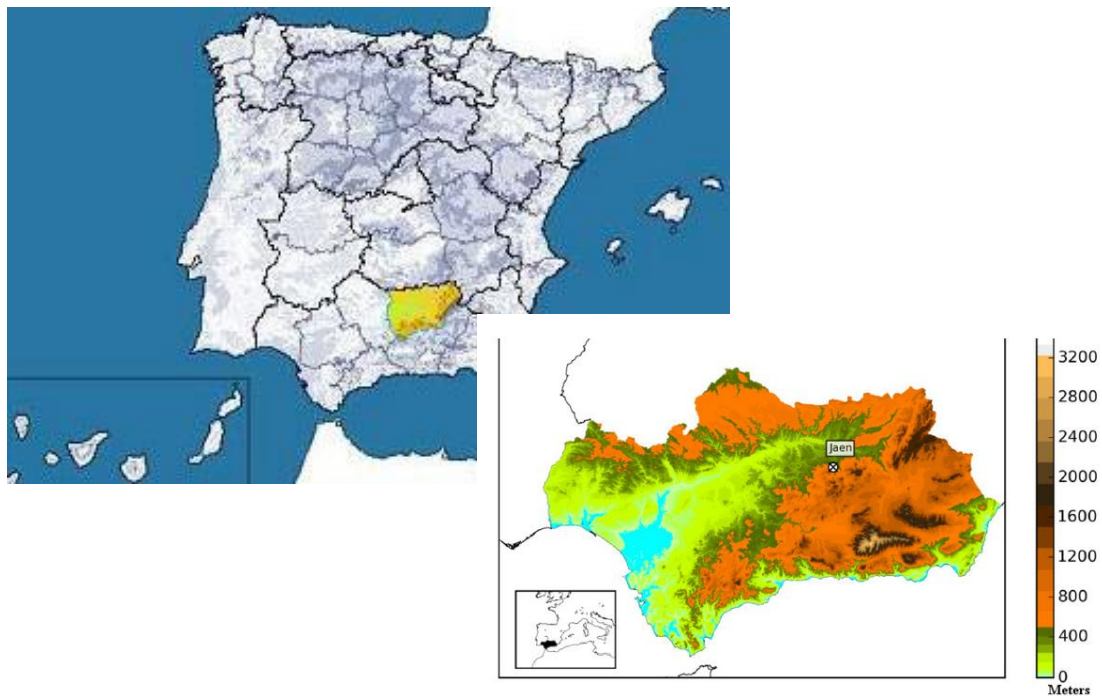


Figura 3. Ubicación del Parque eólico de Sierra del Trigo.

El parque eólico de Sierra del Trigo (Figura 4) está situado en lo alto de la sierra. Consta de una torre meteorológica, la cual se describirá más adelante, y de 23 turbinas eólicas Gamesa G47/660, las cuales tienen una potencia nominal de 660kW. La potencia nominal de parque es de 15 180kW.



Figura 4. Parque eólico de Sierra del Trigo.

2.2 Datos medidos en torre meteorológica

En las instalaciones del parque, se encuentra una torre de medición de velocidad y dirección del viento. Se localiza en las coordenadas: 37.51551721, -3.73882847. La Figura 5 muestra el emplazamiento de la Torre, incluyendo sus características topográficas. Como se observa, la torre está situada en lo alto de una colina.



Figura 5. Localización de la torre de medición. En la figura superior se muestra un mapa topográfico del emplazamiento (Google Earth).

La torre meteorológica (Figura 6) mide la velocidad y dirección del viento en intervalo diez minutas a dos alturas diferentes, 20 y 40 metros. Los datos que se han utilizado en este trabajo corresponden al periodo comprendido entre septiembre de 2019 y junio de 2021.



Figura 6. Imágenes de la torre de medición del viento en el parque de Sierra del Trigo.

2.3. Datos del reanálisis meteorológico ERA-5

ERA-5 es un conjunto de datos de alta precisión de condiciones meteorológicas, generado por el Servicio de Cambio Climático de Copérnico, el cual forma parte del programa de la Unión Europea que realiza el monitoreo meteorológico de la Tierra y que está implementado por ECMWF (Hersbach et al., 2018) (Centro Europeo de Previsiones Meteorológicas a Plazo Medio).

ERA-5 proporciona datos horarios de una gran variedad de parámetros atmosféricos, así como de la superficie terrestre y del estado del mar. Los datos tienen una resolución de 30 km. Del gran listado de parámetros disponibles, se seleccionaron los datos del módulo de la velocidad del viento a 10 y 100 metros de altura correspondientes al periodo entre 2019 y junio de 2021.

2.4. Curvas de potencia de aerogeneradores

El estudio que se llevará a cabo incluye la estimación del factor de ocupación y energía generada por un conjunto de aerogeneradores representativos. La curva de potencia de un aerogenerador representa la potencia eléctrica teóricamente producida por el aerogenerador a distintas velocidades de viento (Sanchez,

2018). En este estudio se van a analizar los siguientes 6 aerogeneradores, cuyas características se han obtenido de la base de datos *The Wind Power* (The Wind Power, 2021) :

- Enercon E82/2300: Potencia nominal 2300kW, funcionamiento entre 2 y 25 m/s, velocidad de potencia nominal de 14m/s.
- Nordex N90/2500: Potencia nominal 2500kW, funcionamiento entre 3,5 y 25 m/s, velocidad de potencia nominal de 14m/s.
- Repower MM82: Potencia nominal 2000kW, funcionamiento entre 3,5 y 22 m/s, velocidad de potencia nominal de 15m/s.
- Vestas V112/3000: Potencia nominal 3000kW, funcionamiento entre 3,5 y 25m/s, velocidad de potencia nominal de 15,5m/s.
- Gamesa G128/4500: Potencia nominal 4500kW, funcionamiento entre 2 y 27m/s, velocidad de potencia nominal de 12,5m/s.
- Siemens Gamesa 2.1-114: Potencia nominal 2100kW, funcionamiento entre 2,5 y 25m/s, velocidad de potencia nominal de 9m/s.

La Figura 7 muestra las curvas de potencia de estos aerogeneradores. Como cabe observar, muestran rangos de funcionamiento y características de potencia muy diferenciadas. Se han elegido para representar los distintos tipos de aerogeneradores que se instalan hoy en día en zonas de montaña.

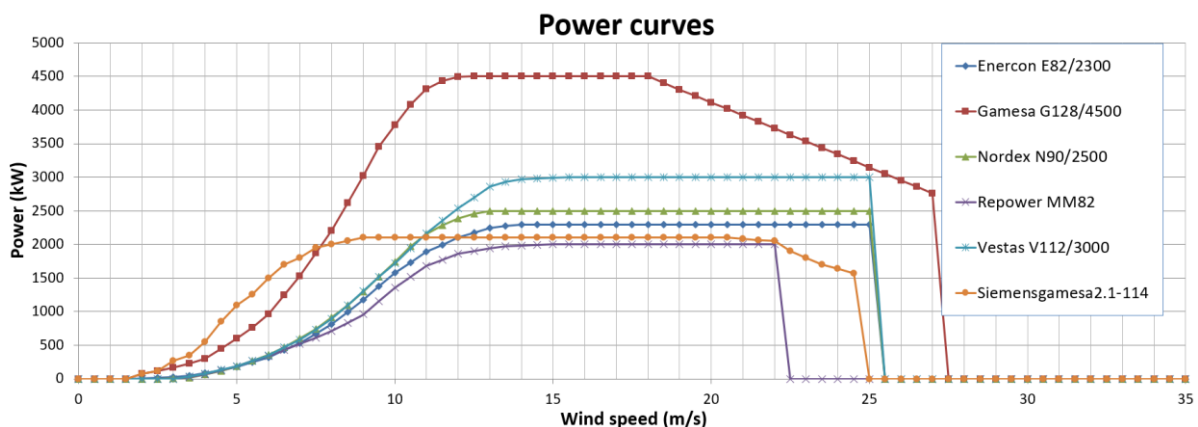


Figura 5. Curvas de potencia de los aerogeneradores (The Wind Power).

3. METODOLOGÍA

3.1 Tratamiento de los datos medidos en torre

Una vez recopilados los datos, el primer paso es realizar una comprobación visual y aplicar un filtro para eliminar datos erróneos, así como realizar el tratamiento de los datos de registros faltantes (*missing data*). El resultado, tras el filtro, es que el 96,3% de los registros eran válidos y estaban disponibles.

La velocidad de viento viene expresada en los datos de Sierra del Trigo con el nombre WS (*Wind Speed*) y viene registrada a dos alturas diferentes (WS1, a 20 metros y WS2, a 40 metros de ahora en adelante). Se llevó a cabo un estudio estadístico de los datos a ambas alturas, en el que se obtuvieron valores medios. También se representó la dispersión de los datos utilizando un gráfico de cajas (*box-plot*). Finalmente, se construyó la rosa de los vientos para ambas alturas.

3.2 Estimación del viento en altura

Como se ha comentado anteriormente, el interés de este estudio radica en evaluar los datos de ERA-5 a 100 metros de altura. Sin embargo, las alturas disponibles de los datos de la torre son 20 y 40 metros. Estas suelen ser alturas comunes en torres de medición de viento. Por motivos económicos y prácticos, las mediciones de viento en torres meteorológicas se realizan a un conjunto de alturas dadas (20, 40, 60 metros). A partir de estas medidas es posible estimar el viento, con una alta fiabilidad, a cualquier otra altura de interés mediante un modelo de extrapolación. La extrapolación vertical consiste en obtener la velocidad del viento a una altura mayor que las alturas proporcionadas, las cuales sirven como alturas de referencia.

Existen diferentes modelos de extrapolación de distinta complejidad. Los más sencillos tienen en cuenta sólo las condiciones de la superficie donde se encuentra la torre; los más complejos también contemplan las condiciones de estabilidad de la atmósfera (Gualtieri and Secci, 2011). Entre los modelos existentes se encuentra la ley potencial, la ley logarítmica (Frank et al., 2020) y la ley log-lineal (Monin-Obukhov) (Irwin, 1979).

En este estudio, dados los datos registrado en la torre, se va a emplear la llamada ley potencial, que viene expresada como:

$$v_2 = v_1 \left(\frac{z_2}{z_1} \right)^\alpha, \quad (1)$$

siendo v_1 y v_2 la velocidad del viento en la medición de la altura z_1 y z_2 , respectivamente. El exponente, α , es el exponente de cizalladura del viento, el cual se determina en función de las características de la superficie del terreno.

El valor de α se encuentra teóricamente entre 0 y 0,4, siendo 0 una superficie en suelo liso o agua y 0,4 en zonas urbanas con edificios altos. Además, α varía en función de las estaciones del año y del día y la noche (Spera and Ph. D., 2009). Si se dispone de medidas de viento a dos alturas es posible determinar el valor de α a partir de la ley potencial:

$$\alpha = \frac{\ln v_2 - \ln v_1}{\ln z_2 - \ln z_1} \quad (2)$$

En este trabajo se van a utilizar las medidas de viento a 20 y 40 metros para estimar el exponente α . Posteriormente, usando los datos registrados a 40 metros y el valor de α , se extrapolará el viento a 100 metros de altura. Esta extrapolación es la que se usará para validar los datos de ERA-5. A continuación se detalla todo el procedimiento.

Para el cálculo de α , se utilizó la siguiente ecuación:

$$\alpha = \frac{\ln(WS2_{modelo}) - \ln(WS1_{modelo})}{\ln(40) - \ln(20)}, \quad (3)$$

siendo WS2 los datos de viento medidos a 40 metros y WS1 los datos medidos a 20 metros. Dada la variabilidad que presenta el exponente de cizalladura, se llevó a cabo el cálculo de α utilizando diferentes subconjuntos de datos: α anual, α día/noche y α según las estaciones del año, diferenciando día y noche. En concreto se analiza el exponente para los casos:

- **α anual:** Para el cálculo de α anual, se obtuvo la media de los valores disponibles a partir de la ecuación anterior.
- **α día/noche:** Para el cálculo de α día - noche, los valores α se separaron en función de las horas del día. Considerando “día” a las horas comprendidas entre 06h - 20h y noche a las comprendidas entre las 21h - 05h. Por lo que se obtuvieron dos valores para α : α -día y α -noche.
- **α estacional día/noche:** ($\alpha_{est-día/noche}$). Para el cálculo de α en función de las estaciones y de la diferencia entre día y noche, previamente se separaron los valores de α en función de las estaciones del año. Se consideró: diciembre, enero y febrero como invierno; marzo, abril y mayo como primavera; junio, julio y agosto como verano; y septiembre, octubre y noviembre como otoño. Después de esto, se realizó la separación entre día y noche. Se deben obtener ocho resultados: α -invierno-día, α -

invierno-noche, α -primavera-día, α -primavera-noche, α -verano-día, α -verano-noche, α -otoño-día, α -otoño-noche.

Como se ha comentado anteriormente, el modelo de extrapolación de la ley potencial se pretende usar para estimar los datos medidos en torre a 100 metros. Estas extrapolaciones, dada la ausencia de datos medidos a 100 metros, son las que se utilizan en el presente trabajo para evaluar los datos de ERA-5. Esta forma de proceder para evaluar los datos de ERA-5 se fundamenta en el hecho de que, en zonas de montaña, especialmente en colinas, donde se localiza la torre de medición, la variación del viento con la altura es muy limitada. En realidad, el efecto colina hace que el viento prácticamente no varíe con la altura (Xu et al., 2018).

No obstante, es posible obtener una estimación del error que se comete al extrapolar el viento a 100 metros usando la ley potencial utilizando el error que se comente al extrapolar de 20 a 40 metros. Así se ha utilizado la ley potencia para estimar el viento a 40 metros:

$$v_{40m} = (WS1_{verificación}) \cdot \left(\frac{40}{20}\right)^{\alpha_{est-dia/noche}} \quad (4)$$

Para evaluar el error de estimación se utilizan los primeros 20 días de cada mes para estimar los coeficientes α , y los días restantes de cada mes para evaluar las estimaciones del viento a 40 metros de forma independiente. Dada la poca variación del viento con la altura, cabe considerar que el error de estimación del viento a 40 metros de altura obtenido con este procedimiento de la ley potencial es representativo del error que se comete al extrapolar el viento de 40 a 100 metros utilizando la misma ley potencial y los mismos coeficientes de cizalla α .

De esta forma, finalmente, los coeficientes α obtenidos y los datos de viento a 40 metros se utilizarán para la estimación del viento a 100 metros siguiendo la ecuación 5:

$$v_{100m} = (WS2) \cdot \left(\frac{100}{40}\right)^{\alpha_{est-dia/noche}} \quad (5)$$

Estos datos extrapolados serán los que se utilizan para validar los datos de ERA-5, que corresponden a una altura de 100 metros.

3.3 Estimación de la energía generada

El *Capacity Factor* (CF), o factor de ocupación, viene definido como el cociente entre la potencia producida a partir de una velocidad dada del viento (P) y la potencia nominal de un aerogenerador (Pn).

$$CF = \frac{P}{P_n} \quad (6)$$

Para cada uno de los aerogeneradores presentados en la sección 2.4 se calculó el CF horario utilizando los datos viento a 100 metros de altura extrapolados de la torre y los datos de ERA-5. Posteriormente, también se obtuvo la energía producida por cada aerogenerador para el año 2020, agregando la energía producida en cada hora.

3.4 Estadísticos de evaluación

En los distintos análisis del presente estudio se han utilizado los estadísticos RMSE (error cuadrático medio) y el BIAS, definidos en la forma:

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=i}^n (y_i - \hat{y}_i)^2} \quad (7)$$

$$BIAS = \sum_{j=i}^n \frac{(y_i - \hat{y}_i)}{n} \quad (8)$$

Donde n es el número de datos, $\hat{y}_i$ es el valor predicho e y_i es el valor observado.

En la evaluación de los resultados también se han empleado herramientas gráficas. En particular, para el módulo del viento se han utilizado gráficos de caja (o *box-plot*) y gráficos de distribución de datos. Por otro lado, para la evaluación de las direcciones del viento se han utilizado gráficos de rosa de los vientos.

4. RESULTADOS

4.1 Análisis de los datos medidos en Torre meteorológica

En la sección se presentan los resultados del análisis estadístico de los datos de Torre. La Tabla 1 muestra los valores medios de la velocidad para las dos alturas

(WS1 y WS2). Como se observa, el valor WS2 es muy parecido a WS1, indicando muy poca variación del viento medio con la altura.

Valor medio de la velocidad del viento (m/s)	
WS1 (20 m)	6,010
WS2 (40 m)	6,016

Tabla 1. Valor medio de WS1 y WS2.

Otra forma de analizar los datos es mediante la realización de un gráfico de caja, que sirve para ilustrar la dispersión de los valores de viento a las distintas alturas. La Figura 8 muestra dicho gráfico para los datos registrados a las dos alturas de Torre. Como se observa en dicha figura, de nuevo los datos a las dos alturas son muy semejantes. Se observa también que los vientos máximos medidos rondan los 30 m/s, si bien WS1 tiene mayor rango de variabilidad del viento, alcanzando rachas de hasta 31 m/s. Los percentiles 25% a 50% están en el rango 2.5 a 7.5 m/s, aproximadamente.

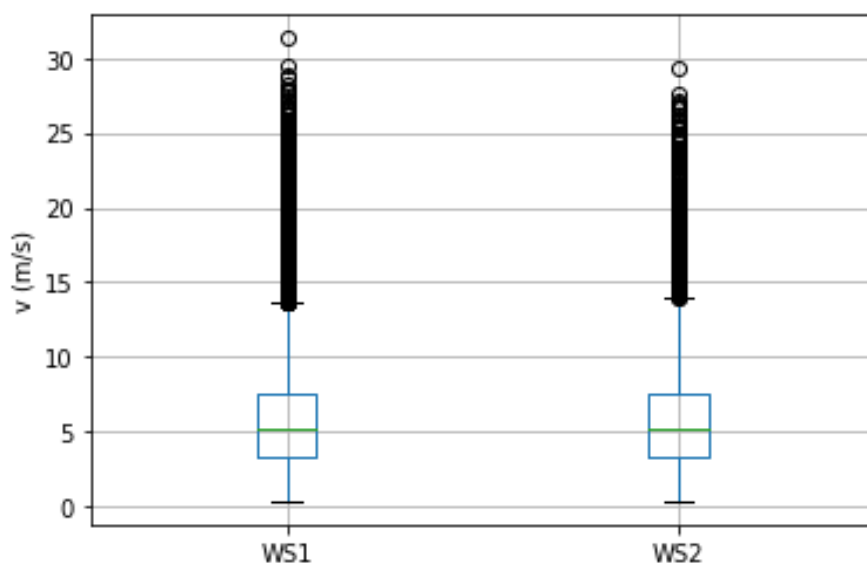


Figura 6. Gráfico de caja de WS1 y WS2.

La rosa de los vientos es otra herramienta útil para analizar las direcciones de los datos de viento. Es el gráfico que se utiliza para representar la velocidad media asociada al viento, sus direcciones predominantes y frecuencia de aparición.

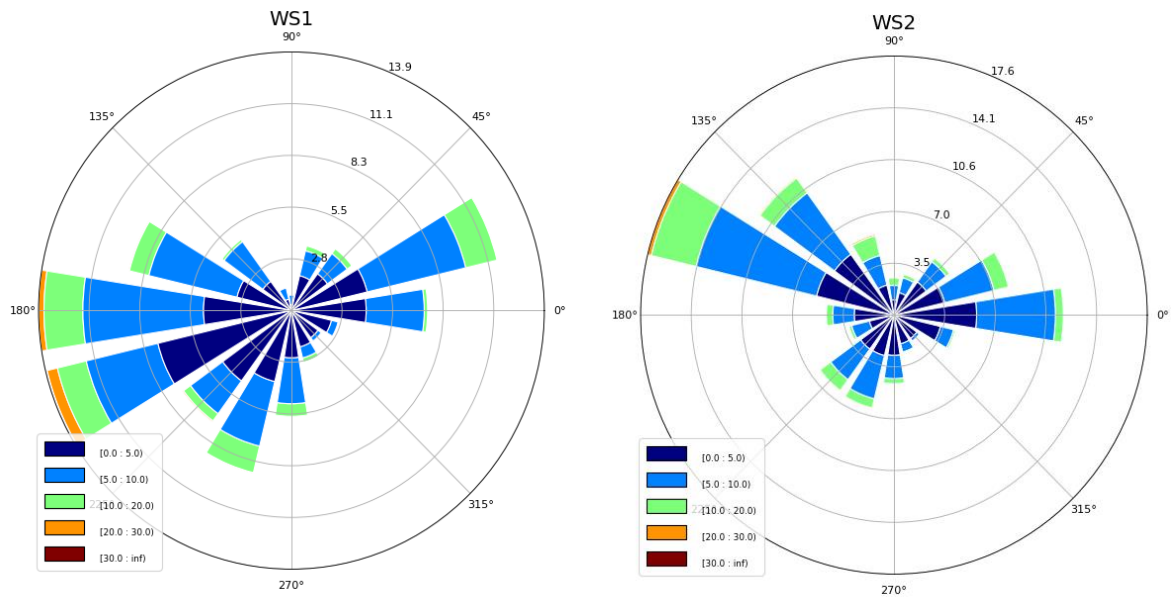


Figura 7. Rosa de los vientos para las dos alturas de Torre.

En la Figura 9, que representa la rosa de los vientos de WS1 y WS2, se puede observar una clara “canalización” de los datos a ambas alturas, como cabía esperar por la localización en una colina del emplazamiento. También se observa una rotación de dicha canalización con la altura.

Respecto a la canalización, en la primera rosa de los vientos (WS1) que corresponde a los datos de menor altura (20 metros), la dirección del viento es más variable, posiblemente por la mayor presencia de turbulencia. La turbulencia se produce por el rozamiento del viento con la orografía, esto ocurre a escasos metros del suelo. Por otro lado, la segunda rosa de los vientos (WS2) presenta una mejor canalización, lo cual es indicativo de menor presencia de turbulencia.

Como se comentó en la sección 2, los datos originales medidos en Torre están disponibles con una resolución temporal de 10 minutos. Sin embargo, los datos de ERA-5 tiene resolución horaria. De cara a la evaluación, se llevó a cabo un promediado horario de los datos medidos en la torre meteorológica, con el fin de homogeneizar los datos antes de compararlos. La Figura 10 muestra un ejemplo de los datos originales y promediados.

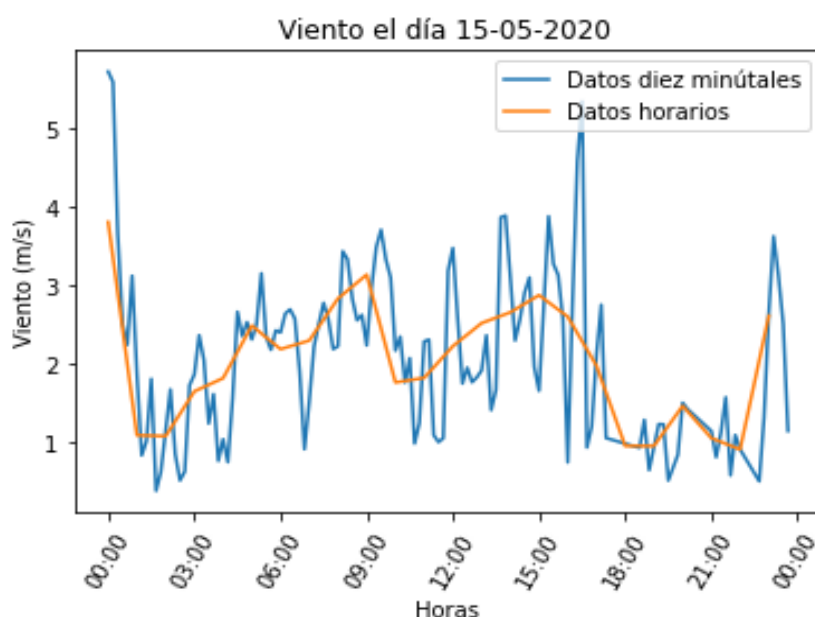


Figura 8. Viento WS2 en Jaén el día 15-05-2020. Valores originales diez minútales y promedio horario.

4.2 Resultados de la extrapolación en altura de los datos de Torre

4.2.1 Estimación del exponente de cizalladura

En esta sección se presentan los resultados relativos a la estimación del coeficiente α de la ley potencial a partir de los valores medidos a 20 y 40 metros. Todos los valores se calcularon con los datos de viento del día 1 al día 20 de cada mes. Se llevaron a cabo varios estudios; en primer lugar, un estudio anual, utilizando todos los valores. En segundo lugar, un estudio en el que se distinguió entre valores diurnos y nocturnos. En tercer lugar, se separaron los valores diurnos y nocturnos también en función de la estación del año.

La tabla 2 muestra los datos para el estudio anual. El valor de α anual presenta un valor cercano a cero, indicando que apenas hay variación del viento con la altura.

α anual	-0,008
----------------	--------

Tabla 2. α anual.

La Tabla 3 muestra los resultados para el estudio día/noche. Igual que en el caso anterior, los valores son muy cercanos a cero, si bien son ligeramente mayores durante la noche, como cabría esperar al ser menor el efecto de la turbulencia.

α día	-0,021
α noche	0,015

Tabla 3. Valor de α del estadio día – noche.

Finalmente, la Tabla 4 muestra los resultados para el estudio estacional-día/noche. De nuevo los valores son todos muy cercano a cero, si bien los valores del exponente de cizalladura son mayores durante la noche. Como se ha comentado, este era esperable, ya que durante el día existe mayor turbulencia producida por la convección, que hace que el viento apenas varíe con la altura. Durante la noche hay mayor estabilidad, aumentando así el valor del exponente de cizalladura ligeramente.

α invierno-día	0,0001
α invierno-noche	0,0466
α primavera-día	-0,0511
α primavera-noche	-0,0288
α verano-día	-0,0136
α verano-noche	0,0264
α otoño-día	-0,0147
α otoño-noche	0,0234

Tabla 4. Valor de α del estadio estacional, día – noche.

Como conclusión de este estudio cabe decir que el emplazamiento muestra muy poca variación del viento con la altura, resultando en valores de α prácticamente nulos. Estos resultados concuerdan con los de estudios similares en zonas de montaña (Xu et al., 2018), y se deben al impacto de la topografía en el viento. El efecto colina hace que el flujo se acelere, dando lugar a una muy escasa variación del viento con la altura.

4.2.2 Estimación del viento a 40 y 100 metros

Para llevar a cabo la extrapolación del viento medido en la torre de 40 a 100 metros se utilizaron los valores del exponente de α calculados en el estudio estacional día-noche. Esta selección se basó en que de esta forma se contempla

en la estimación las variaciones estacionales y noche/día que se producen en las condiciones de la superficie del terreno donde se localiza la torre de medición.

Como se comentó en la sección 3 de metodología, la estimación del viento a 40 metros a partir de los datos de 20 metros permite cuantificar el error que se comete al extrapolar los datos de 40 metros a 100 metros de altura. La Tabla 5 muestra el valor de RMSE para estas estimaciones, utilizando los propios datos medidos a 40 metros para validarlas. Cabe recordar que se utilizaron valores independientes para la estimación del exponente α y para el cálculo del valor del RMSE. El valor encontrado para el RMSE de las estimaciones del viento a 40 metros es de 1,105 m/s. Se trata de un error relativamente bajo, en línea con el encontrado en estudios similares (Murcia et al., 2022). Este error es el que asumimos que cometemos en la extrapolación del viento de 40 a 100 metros de altura, y que usaremos para validar los datos de ERA-5.

RMSE (viento a 40m)	1,105 m/s
---------------------	-----------

Tabla 5. RMSE resultante de la comparación del viento a 40 metros.

4.3 Comparación entre datos de viento medidos en Torre y datos de ERA-5

A partir de los valores obtenidos del exponente α en el estudio estacional día/noche, y usando los datos registrados por la torre a 40 metros, se extrapolaron los datos de torre a 100 metros. Estos datos se compararon con los datos de ERA-5, que corresponden a dicha altura.

La Figura 11 muestra las distribuciones de viento de la torre y de ERA-5, y la Figura 12 muestra un gráfico de cajas para ambos conjuntos de datos.

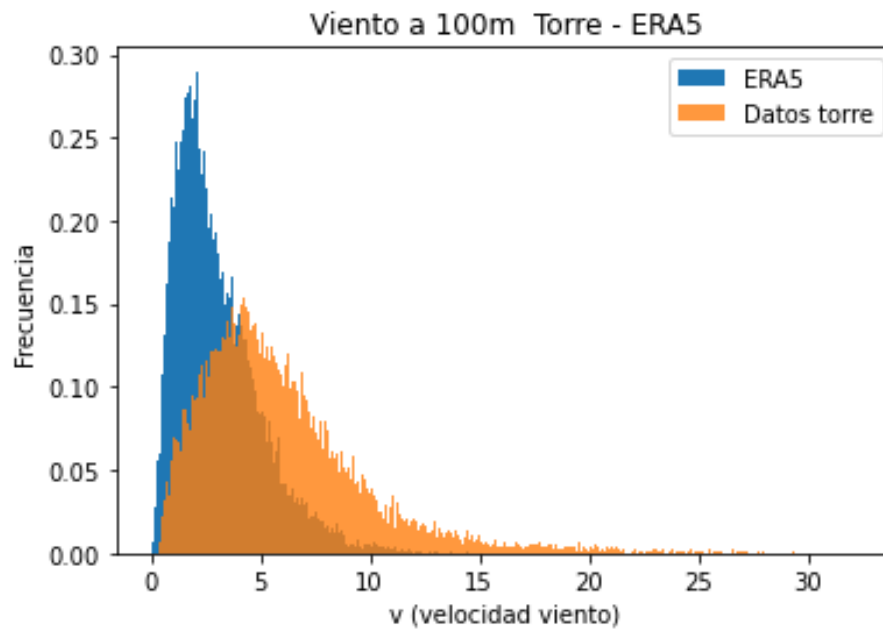


Figura 9. Distribución de *Weibull* a 100m de altura de ERA-5 y torre.

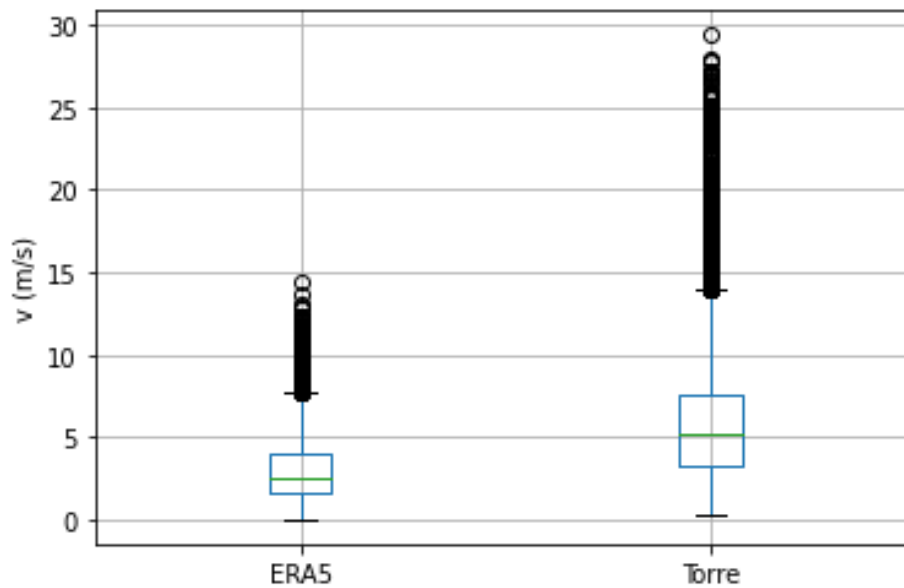


Figura 10. Diagrama de caja y bigotes de ERA-5 y Torre a 100m de altura.

Como se observa en la Figura 11, los datos de torre y de ERA-5 muestran una distribución muy distinta. La distribución de datos de viento de ERA-5 está desplazada hacia la izquierda, indicando valores mucho menores que los de Torre. La consecuencia es que los datos ERA-5 apenas contemplan velocidades del viento por encima de 10 m/s. Las mismas conclusiones cabe extraer de la

Figura 12. La velocidad más frecuente en ERA-5 se encuentra a 2,55 m/s, mientras que la de Torre alcanza los 5,16 m/s, distando más de 2,5 m/s, indicando así una diferencia significativa entre ellos. Notemos que la diferencia en las medias de ambas bases de datos es mucho mayor que el error extrapolación de los datos de la Torre (1,105 m/s) (Tabla5). Esto indica que, aun asumiendo este error, las diferencias entre los datos de Torre y los datos ERA-5 son significativas.

El cuartil superior de ERA-5 es inferior a la mediana de Torre, lo cual indica que los datos de ERA-5 son mucho más pequeños, además, los datos atípicos presentan valores superiores en Torre, alcanzando rachas de viento superiores a 25 m/s en algunas ocasiones (Fig.12).

En conclusión, la diferencia entre los datos de velocidad de viento a 100 metros para ERA-5 y Torre es muy importante, y puede tener notables consecuencias en la estimación de la energía y por tanto en el estudio de la viabilidad económica del parque. Este impacto se aborda en el siguiente apartado.

4.4 Comparación entre las estimaciones del recurso eólico con datos medidos en Torre y datos de ERA-5.

En esta sección se comparan las estimaciones de CF y energía de los seis aerogeneradores presentados en la sección 4 de datos, obtenidos a partir de los datos de Torre y ERA-5. Solo se muestran para 5 de los aerogeneradores; los valores del Nordex N90/2500 no se presenta al ser muy similares a los del Enercon E82/2300 y Repower MM82. La Tabla 6 muestra los valores de CF medio y la desviación estándar para los restantes 5.

	ENERCON		REPOWER		VESTAS		GAMESA		SIEMENS GAMESA	
	$\overline{CF}$	σ_{CF}	$\overline{CF}$	σ_{CF}	$\overline{CF}$	σ_{CF}	$\overline{CF}$	σ_{CF}	$\overline{CF}$	σ_{CF}
ERA-5	0,039	0,098	0,041	0,101	0,031	0,084	0,066	0,127	0,178	0,267
TORRE	0,191	0,263	0,195	0,260	0,166	0,245	0,252	0,290	0,499	0,372

Tabla 6. CF medio y desviación estándar de ERA-5 y datos de Torre.

Los resultados obtenidos muestran que el CF medio de los datos de Torre son mucho mayores que los obtenidos usando datos de ERA-5, llegando incluso a casi quintuplicar su valor para algunos aerogeneradores. Esto se debe al limitado rango de velocidades del viento que muestran los datos ERA-5, que repercute en las horas de funcionamiento de los aerogeneradores. Respecto a la

desviación estándar, los datos de viento de la torre muestran una mayor variabilidad que incide en un mayor valor de la desviación estándar, tal y como se observa en la Tabla 6.

En cuanto a los aerogeneradores, se obtiene un CF mayor para el Gamesa y el Siemens Gamesa. Esto se debe a la morfología de sus curvas de potencia (Fig. 7), la cual, tras alcanzar la potencia nominal a una velocidad máxima, sigue generando potencia hasta alcanzar la velocidad de corte. El Siemens Gamesa presenta un CF mayor que el Gamesa debido a que, pese a que su potencia nominal es prácticamente la mitad, funciona mejor a velocidades pequeñas, mientras que el Gamesa hace lo contrario. Respecto a los otros aerogeneradores, cabe resaltar que, a bajas velocidades, entre 3 y 7,5 m/s, la curva de potencia presenta una pendiente muy acusada en el Siemens Gamesa, mientras que para otros aerogeneradores como el Vestas, el Repower o el Enercon ocurre todo lo contrario.

Finalmente, para ver el impacto que tiene la diferencia entre los datos de viento de ERA-5 y de Torre en la viabilidad de un proyecto de parque eólico, se ha calculado la energía anual que producirían los distintos aerogeneradores a partir de ambos tipos de datos. Los resultados se muestran en la Tabla 7.

	ENERGÍA (MWh)				
	ENERCON	REPOWER	VESTAS	GAMESA	SIEMENS GAMESA
ERA-5	225	209	213	806	1899
TORRE	2079	1805	2240	5681	6961
Producción ERA-5 respecto a Torre (%)	10,4	11,6	9,53	14,20	27,29

Tabla 7. Estimación de la energía anual producida con diferentes aerogeneradores.

La energía producida con los datos de viento de Torre es mucho mayor a la producida con ERA-5, como cabía esperar. El efecto acumulativo anual es muy acusado para todos los aerogeneradores, siendo la producción estimada con ERA-5 un 27% de la estimada con Torre en el mejor de los casos (Siemens-Gamesa). Para el aerogenerador Vestas este porcentaje es menor y se encuentra en torno al 10%.

En cualquier caso, el aerogenerador de mayor interés para implantar en el emplazamiento sería el Siemens Gamesa 2.1-114, ya que, dado el recurso eólico del parque, presenta la mayor generación entre los analizados.

5. CONCLUSIONES

En este trabajo, se han llevado a cabo varios análisis tendentes a evaluar la fiabilidad que tienen los datos del reanálisis ERA-5 de cara a ser empleados para la evaluación del recurso eólico en una zona montañosa. En el estudio se han empleado datos medidos a 20 y 40 metros de altura en una torre meteorológica localizada en un emplazamiento de montaña localizado en Sierra del Trigo (Jaén).

En un primer análisis se determinó el exponente de cizalladura del viento, necesario para extrapolar datos de viento con la altura, a partir de los datos medidos a 20 y 40 metros. Los resultados mostraron que, en el lugar de estudio, el viento apenas varía con la altura, obteniéndose un exponente de cizalladura cercano a cero.

En un segundo estudio se extrapolaron los datos de 20 a 40 metros usando la ley potencial y el exponente de cizalladura calculado. Los datos extrapolados se evaluaron usando un conjunto independiente de datos medidos también a 40 metros. En la evaluación se obtuvo un error cuadrático de 1,105 m/s. Este error de estimación se considera extrapolable a 100 metros debido al ínfimo valor del exponente de cizalladura.

En el tercer estudio se compararon los datos de viento de ERA-5 con los datos de Torre extrapolados a 100 metros. Los resultados mostraron una muy importante diferencia en las distribuciones de ambas bases de datos. En particular, se encontró una mediana de los datos de torre de 5,16 m/s mientras que para los datos de ERA-5 fue de tan solo de 2,5 m/s. La diferencia es, pues mayor que el error de extrapolación de los datos de la torre y, por lo tanto, significativa. La diferencia en los valores extremos de ambas distribuciones también se encontró ser muy significativa.

Finalmente, se compararon los factores de ocupación y la energía generada por diferentes aerogeneradores de referencia usando las dos fuentes de datos. Los resultados mostraron que los factores de ocupación obtenidos a partir de los datos de Torre son hasta 5 veces mayores que los derivados de los datos de ERA-5. Diferencias similares fueron encontradas para la energía generada en un año por los distintos aerogeneradores usando ambas bases de datos.

A modo de conclusión general, cabe decir que los datos de ERA-5 no pueden ser utilizados de forma directa para la estimación del recurso eólico en emplazamientos como el estudiado, con una alta complejidad topográfica. La razón es que, debido a la baja resolución espacial del modelo (30 km), los efectos que la topografía tiene en la modificación del viento a escala local (efecto valle o efecto colina) no quedan representados en los datos ERA-5. Esto supone un

grave problema en regiones como España, donde el recurso eólico está asociado a dichos efectos topográficos.

Para tratar de solventar este problema, cabe llevar a cabo un intento de corrección de los datos de ERA-5, para ver si es posible incluir los efectos topográficos y mejorar las estimaciones de viento. Para ello se puede derivar un modelo de corrección de los datos ERA-5 mediante la comparación con los datos medidos en la torre meteorológica. Se pretende llevar a cabo este estudio en un próximo futuro.

6. REFERENCIAS

- IRENA, (2019). "FUTURE OF WIND Deployment, Investment, Technology, Grid Integration and Socio-Economic Aspects A Global Energy Transformation Paper Citation About IRENA."
- MERRA. (2021). "MERRA-2." Retrieved December 2, 2021 (<https://gmao.gsfc.nasa.gov/reanalysis/MERRA-2/>).
- The Wind Power. (2021). "The Wind Power." Retrieved December 2, 2021 (https://www.thewindpower.net/store_manufacturer_turbine_es.php?id_type=7).
- Frank, Christopher W., Bernhard Pospichal, Sabrina Wahl, Jan D. Keller, Andreas Hense, and Susanne Crewell. 2020. "The Added Value of High Resolution Regional Reanalyses for Wind Power Applications." *Renewable Energy* 148:1094–1109. doi: 10.1016/J.RENENE.2019.09.138.
- Gualtieri, Giovanni, and Sauro Secci. 2011. "Comparing Methods to Calculate Atmospheric Stability-Dependent Wind Speed Profiles: A Case Study on Coastal Location." *Renewable Energy* 36(8):2189–2204. doi: 10.1016/J.RENENE.2011.01.023.
- Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., Schepers, D., Simmons, A., Soci, C., Dee, D., Thépaut, J. N. 2018. "ERA5 Hourly Data on Pressure Levels from 1979 to Present." *Copernicus Climate Change Service (C3S); Climate Data Store (CDS)*, 10.24381/cds.bd0915c6.
- Irwin, John S. 1979. "A Theoretical Variation of the Wind Profile Power-Law Exponent as a Function of Surface Roughness and Stability." *Atmospheric Environment* (1967) 13(1):191–94. doi: 10.1016/0004-6981(79)90260-9.
- Murcia, Juan Pablo, Matti Juhani Koivisto, Graziela Luzia, Bjarke T. Olsen, Andrea N. Hahmann, Poul Ejnar Sørensen, and Magnus Als. 2022. "Validation of European-Scale Simulated Wind Speed and Wind Generation Time Series." *Applied Energy* 305:117794. doi: 10.1016/J.APENERGY.2021.117794.
- PNIEC. (2020). "Plan Nacional Integrado de Energía y Clima (PNIEC) 2021-

2030.” Retrieved December 2, 2021
(<https://www.miteco.gob.es/es/prensa/pniec.aspx>).

- PNUMA. (2019). *Informe Sobre La Disparidad En Las Emisiones de 2019*. Programa de las Naciones Unidas para el Medio Ambiente, Nairobi.
- Sanchez, E. R. 2018. “OBTENCIÓN DE LA CURVA DE POTENCIA DE UN AEROGENERADOR DE 2MW, MEDIANTE UN MODELO CFD (OBTAINMENT OF THE POWER CURVE OF A WIND TURBINE 2MW, THROUGH A CFD MODEL).” *Pistas Educativa* 40(130).
- Santos-Alamillos, Francisco J., David J. Brayshaw, John Methven, Nikolaos S. Thomaidis, José A. Ruiz-Arias, and David Pozo-Vázquez. 2017. “Exploring the Meteorological Potential for Planning a High Performance European Electricity Super-Grid: Optimal Power Capacity Distribution among Countries.” *Environmental Research Letters* 12(11). doi: 10.1088/1748-9326/AA8F18.
- Sharp, Ed, Paul Dodds, Mark Barrett, and Catalina Spataru. 2015. “Evaluating the Accuracy of CFSR Reanalysis Hourly Wind Speed Forecasts for the UK, Using in Situ Measurements and Geographical Information.” *Renewable Energy* 77:527–38. doi: 10.1016/J.RENENE.2014.12.025.
- Slivinski, Laura C., Gilbert P. Compo, Jeffrey S. Whitaker, Prashant D. Sardeshmukh, Benjamin S. Giese, Chesley McColl, Rob Allan, Xungang Yin, Russell Vose, Holly Titchner, John Kennedy, Lawrence J. Spencer, Linden Ashcroft, Stefan Brönnimann, Manola Brunet, Dario Camuffo, Richard Cornes, Thomas A. Cram, Richard Crouthamel, Fernando Domínguez-Castro, J. Eric Freeman, Joëlle Gergis, Ed Hawkins, Philip D. Jones, Sylvie Jourdain, Alexey Kaplan, Hisayuki Kubota, Frank Le Blancq, Tsz Cheung Lee, Andrew Lorrey, Jürg Luterbacher, Maurizio Maugeri, Cary J. Mock, G. W. Ken. Moore, Rajmund Przybylak, Christa Pudmenzky, Chris Reason, Victoria C. Slonosky, Catherine A. Smith, Birger Tinz, Blair Trewin, Maria Antónia Valente, Xiaolan L. Wang, Clive Wilkinson, Kevin Wood, and Przemysław Wyszzyński. 2019. “Towards a More Reliable Historical Reanalysis: Improvements for Version 3 of the Twentieth Century Reanalysis System.” *Quarterly Journal of the Royal Meteorological Society* 145(724):2876–2908. doi: 10.1002/QJ.3598.
- Spera, D. A., and Ph. D. 2009. *Wind Turbine Technology: Fundamental Concepts in Wind Turbine Engineering*. Second Edition. ASME Press.
- Xu, Chang, Chenyan Hao, Linmin Li, Xingxing Han, Feifei Xue, Mingwei Sun, and Wenzhong Shen. 2018. “Evaluation of the Power-Lawwind-Speed Extrapolation Method with Atmospheric Stability Classification Methods for Flows over Different Terrain Types.” *Applied Sciences (Switzerland)* 8(9). doi: 10.3390/APP8091429.

Anexos

A continuación, se adjunta el código empleado en la elaboración de este trabajo (lenguaje de programación Python).

```
1.  # -*- coding: utf-8 -*-
2.  """
3.  Created on Mon Oct 11 11:45:24 2021
4.
5.  @author: anel
6.  """
7.  #Script para la realización del TFM
8.
9.  from sklearn.metrics import mean_squared_error
10. import matplotlib.dates as mdates
11. import math
12. import pandas as pd
13. import numpy as np
14. from datetime import datetime, timedelta
15. from matplotlib import pyplot
16. import matplotlib.pyplot as plt
17. from scipy import interpolate
18. from windrose import WindroseAxes
19. from sklearn.metrics import mean_absolute_error
20.
21.
22.
23.
24. #####
25. #
26. #                                     Funciones
27. #
28. #####
29.
30. #=====
31. # Curva de potencia Gamesa 660
32. #=====
33. def GamesaGeneracion(v_modulo, v_min = 4.5, v_max=25., tipoInterp = 'cubic', v_viento=None,
34.                       potencia=None):
35.     """
36.     Función que calcula la curva de potencia para los aerogeneradores Gamesa 660
37.     inputs:
38.         v_modulo    float    velocidad del viento m/s
39.         v_min       float    veloc. arranque generador m/s vxd=4.5 (valor por defecto)
40.         v_max       float    veloc. pare generador m/s vdx=25
41.     outputs:
42.         generacion    float    potencia generada en kW
43.     """
44.     if v_viento==None:
45.         v_viento = [4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5, 11, 11.5, 12, 12.5,
46.                     13, 13.5, 14, 14.5, 15, 15.5, 16, 16.5, 17, 17.5, 18, 18.5, 19, 19.5, 20, 20.5, 21, 21.5, 22,
47.                     22.5, 23, 23.5, 24, 24.5, 25]
48.         if potencia==None:
49.             potencia = [26,55,78, 105, 133, 165, 209, 254, 301, 347, 405, 462, 509, 556, 593, 629,
50.                         649, 667, 679, 690, 693, 696, 698, 700, 700, 700, 700, 700, 700, 700, 700, 700, 700, 700, 700,
51.                         700, 700, 700, 700, 700, 700, 700]
52.         #la potencia está en kW
53.
54.         if v_modulo < v_min:
55.             generacion = 0.
56.         elif v_modulo > v_max:
57.             generacion = 0.
58.         else:
59.             if v_modulo <= 16.:
60.                 funcionInterp = interpolate.interp1d(x=v_viento, y=potencia, kind=tipoInterp)
61.                 generacion = funcionInterp(v_modulo)
62.             else:
63.                 generacion =700
64.         return generacion
65.
```

```

66. #=====
67. # Asigna formato a fecha
68. #=====
69. def function_date(x): return (datetime.strptime( x[: -6], '%Y-%m-%dT%H:%M:%S') -
    timedelta(hours=int(x[-5:-3])))
70.
71.
72. #=====
73. # Estadísticos
74. #=====
75.
76.
77. #*****
78. # Error MSE
79. #*****
80. def RMSE(true, pred):
81.     """
82.     Función que devuelve el error RMSE
83.     inputs:
84.         true    float    Valor observado
85.         pred    float    valor predicho
86.     outputs:
87.         rmse    float    valor del error MSE
88.     """
89.
90.     rmse = math.sqrt(mean_squared_error(true, pred))
91.     return rmse
92.
93. #*****
94. # Error MAE
95. #*****
96.
97. def MAE(true, pred):
98.     """
99.     Función que devuelve el error MAE
100.    inputs:
101.        true    float    valor observado
102.        pred    float    valor predicho
103.    outputs:
104.        mae     float    valor del error MAE
105.    """
106.    mae = mean_absolute_error(true, pred)
107.    return mae
108.
109. #*****
110. # Error BIAS
111. #*****
112.
113. def BIAS(true, pred):
114.     """
115.     Función que devuelve el error BIAS
116.     inputs:
117.         true    float    valor observado
118.         pred    float    valor predicho
119.     outputs:
120.         bias    float    valor del error BIAS
121.     """
122.     bias = np.mean(true-pred)
123.     return bias
124.
125.
126.
127. #=====
128. # Datos de modelo y corrección
129. #=====
130.
131. def modelo_validacion (x,tiempo):
132.     """
133.     Función que separa los datos en datos para el modelo y datos para la corrección
134.     inputs:
135.         x            float        DataFrame a tratar
136.         tiempo       string       nombre de columna de tiempo del DataFrame a tratar
137.     outputs:
138.         modelo        DataFrame   Datos del 1-20 de cada mes de todo el DataFrame x
139.         correccion    DataFrame   Datos del 21 a fin de mes de todo el DataFrame x.
140.     """
141.     x = x.reset_index()
142.     dias = np.zeros(len(x))
143.     for i in range(len(x)):

```

```

144.         f = x[tiempo][i]
145.         dias[i] = f.day
146.         modelo = x[(dias<=20)]
147.         validacion = x[(dias>20)]
148.         modelo = modelo.set_index(tiempo)
149.         validacion = validacion.set_index(tiempo)
150.         return modelo, validacion
151.
152.
153. #=====
154. # Separación día - noche para  $\alpha$ 
155. #=====
156.
157. def  $\alpha_{\text{dia\_noche}}$ (x):
158.     """
159.     Función que separa los  $\alpha$  en función del día y la noche
160.     inputs:
161.         x            float   DataFrame a tratar
162.     outputs:
163.          $\alpha_{\text{dia}}$     float   media de  $\alpha$  para las horas correspondientes al día (6am - 8pm)
164.          $\alpha_{\text{noche}}$  float   media de  $\alpha$  para las hora correspondientes a la noche (9pm - 5am)
165.     """
166.      $\alpha_{\text{dia}}$  = x.between_time('06:00:00' , '20:00:00')
167.      $\alpha_{\text{noche}}$  = x.between_time('21:00:00' , '05:00:00')
168.      $\alpha_{\text{dia}}$  =  $\alpha_{\text{dia}}$ .mean()
169.      $\alpha_{\text{noche}}$  =  $\alpha_{\text{noche}}$ .mean()
170.     return  $\alpha_{\text{dia}}$ ,  $\alpha_{\text{noche}}$ 
171.
172.
173. #=====
174. # Separación de los meses según las estaciones del año
175. #=====
176.
177. def estaciones(x, date):
178.     """
179.     Función que separa los  $\alpha$  en función de las estaciones del año.
180.     x            float   DataFrame a tratar
181.     date         string  nombre de columna de tiempo del DataFrame a tratar
182.     outputs:
183.         invierno  float    $\alpha$  para los meses de invierno (diciembre, enero, febrero)
184.         primavera float    $\alpha$  para los meses de primavera(marzo, abril, mayo)
185.         verano    float    $\alpha$  para los meses de verano (junio, julio, agosto)
186.         otoño     float    $\alpha$  para los meses de otoño (septiembre, octubre, noviembre)
187.     """
188.
189.     x = x.reset_index()
190.     meses = np.zeros(len(x))
191.     for i in range(len(x)):
192.         f = x[date][i]
193.         meses[i] = f.month
194.         invierno = x[(meses<3) | (meses>11)]
195.         primavera = x[(meses>=3) & (meses<=5)]
196.         verano = x[(meses>=6) & (meses<=8)]
197.         otoño = x[(meses>=9) & (meses<=11)]
198.         invierno = invierno.set_index(date)
199.         primavera= primavera.set_index(date)
200.         verano = verano.set_index(date)
201.         otoño = otoño.set_index(date)
202.
203.     return invierno, primavera, verano, otoño
204.
205.
206.
207. #=====
208. #  $\alpha$  según las estaciones y día-noche
209. #=====
210.
211. def  $\alpha_{\text{estacional}}$  (invierno, primavera, verano, otoño):
212.     """
213.     Función que separa los  $\alpha$  en función de las estaciones del año, teniendo en cuenta el día y la noche. En total devuelve 8 valores.
214.     inputs:
215.         invierno  float    $\alpha$  para los meses de invierno (diciembre, enero, febrero)
216.         primavera float    $\alpha$  para los meses de primavera(marzo, abril, mayo)
217.         verano    float    $\alpha$  para los meses de verano (junio, julio, agosto)
218.         otoño     float    $\alpha$  para los meses de otoño (septiembre, octubre, noviembre)
219.     outputs:
220.         invierno_dia    float    $\alpha$  medio para los meses de invierno durante el día
221.         invierno_noche  float    $\alpha$  medio para los meses de invierno durante la noche

```

```

222.     primavera_dia      float     $\alpha$  medio para los meses de primavera durante el día
223.     primavera_noche    float     $\alpha$  medio para los meses de primavera durante la noche
224.     verano_dia         float     $\alpha$  medio para los meses de verano durante el día
225.     verano_noche       float     $\alpha$  medio para los meses de verano durante la noche
226.     otoño_dia          float     $\alpha$  medio para los meses de otoño durante el día
227.     otoño_noche        float     $\alpha$  medio para los meses de otoño durante la noche
228.     """
229.
230.
231.     invierno_dia, invierno_noche =  $\alpha$ _dia_noche(invierno)
232.     primavera_dia, primavera_noche =  $\alpha$ _dia_noche(primavera)
233.     verano_dia, verano_noche =  $\alpha$ _dia_noche(verano)
234.     otoño_dia, otoño_noche =  $\alpha$ _dia_noche(otoño)
235.      $\alpha$ _estaciones = pd.DataFrame([[invierno_dia[0]], [invierno_noche[0]], [primavera_dia[0]],
[primavera_noche[0]], [verano_dia[0]], [verano_noche[0]], [otoño_dia[0]], [otoño_noche[0]],
index = ['invierno_dia', 'invierno_noche', 'primavera_dia', 'primavera_noche', 'verano_dia',
'verano_noche', 'otoño_dia', 'otoño_noche']])
236.
237.     return  $\alpha$ _estaciones
238.
239.
240.
241. #=====
242. # Separación de días en día y noche
243. #=====
244.
245. def dia_noche (x):
246.     """
247.     Función que separa los días en día y noche
248.     inputs:
249.         x            float    DataFrame a tratar
250.     outputs:
251.         dia          float    valores de velocidad para las hora correspondientes al día (6am - 8pm)
252.         noche        float    valores de velocidad para las hora correspondientes a la noche (9pm -
5am)
253.     """
254.     dia = x.between_time('06:00:00', '20:00:00')
255.     noche = x.between_time('21:00:00', '05:00:00')
256.     return dia, noche
257.
258. #=====
259. # Interpolaciones
260. #=====
261.
262.
263. #*****
264. # Ley potencial
265. #*****
266.
267. def ley_potencial( v1, date, a, z1, z2):
268.     """
269.     Función que devuelve la velocidad a la altura requerida
270.     inputs:
271.         v1          float    valor de referencia de la velocidad del viento
272.         z1          float    altura a la que se mide la velocidad inicial
273.         z2          float    altura a la que se mide la velocidad resultante
274.          $\alpha$         float    exponente de cizalladura
275.     outputs:
276.         v2          float    valor de la velocidad resultante a la altura deseada
277.     """
278.
279.     invierno, primavera, verano, otoño = estaciones(v1, date)
280.
281.     invierno_dia, invierno_noche = dia_noche(invierno)
282.     primavera_dia, primavera_noche = dia_noche(primavera)
283.     verano_dia, verano_noche = dia_noche(verano)
284.     otoño_dia, otoño_noche = dia_noche(otoño)
285.
286.     v2_invierno_dia = invierno_dia*pow((z2/z1), a[0][0])
287.     v2_invierno_noche = invierno_noche*pow((z2/z1), a[0][1])
288.     v2_primavera_dia = primavera_dia*pow((z2/z1), a[0][2])
289.     v2_primavera_noche = primavera_noche*pow((z2/z1), a[0][3])
290.     v2_verano_dia = verano_dia*pow((z2/z1), a[0][4])
291.     v2_verano_noche = verano_noche*pow((z2/z1), a[0][5])
292.     v2_otoño_dia = otoño_dia*pow((z2/z1), a[0][6])
293.     v2_otoño_noche = otoño_noche*pow((z2/z1), a[0][7])
294.
295.     v2 = pd.concat([v2_invierno_dia, v2_invierno_noche, v2_primavera_dia, v2_primavera_noche,
v2_verano_dia, v2_verano_noche, v2_otoño_dia, v2_otoño_noche])

```



```

296.     v2 = v2.sort_index()
297.     return v2
298.
299.
300. #####
301. # Curvas de potencia (Wind Power)
302. #####
303.
304. def Generacion(v_modulo, v_min , v_max, v_corte, v_viento, potencia, tipoInterp = 'cubic'):
305.     """
306.     Función que calcula la curva de potencia para los aerogeneradores obtenidos de Wind Power
307.     inputs:
308.         v_modulo    float    datagrama de velocidad del viento m/s
309.         v_min       float    veloc. arranque generador m/s
310.         v_max       float    veloc. pare generador m/s
311.         v_viento    float    rango de viento que viene en curva de potencia
312.         v_corte     float    velocidad a la que se deja de tener potencia nominal, pero se
313.             sigue generando potencia
314.         potencia    int      potencia del aerogenerador que viene en curva de potencia
315.     outputs:
316.         datos_altura float    potencia generada en kW
317.         """
318.         datos_altura = pd.DataFrame(v_modulo.values, index = v_modulo.index, columns =
319.             ['velocidad'])
320.         potencia_datos_altura = np.zeros(len(datos_altura))
321.         for i in range(len(datos_altura)):
322.             e = datos_altura['velocidad'][i]
323.
324.             if e < v_min:
325.                 generacion1 = 0.
326.             elif e > v_max:
327.                 generacion1 = 0.
328.             elif e <= v_max and e > v_corte:
329.                 funcionInterp_lineal =
330.                 interpolate.interp1d(x=v_viento[np.where(v_viento==v_corte)[0][0]-
331.                 1:np.where(v_viento==v_max)[0][0]+1], y=potencia[np.where(v_viento==v_corte)[0][0]-
332.                 1:np.where(v_viento==v_max)[0][0]+1], kind='linear')
333.                 generacion1 = funcionInterp_lineal(e)
334.             else:
335.                 idMax = potencia.argmax()
336.                 if e <= v_viento[idMax]:
337.                     funcionInterp_cubica = interpolate.interp1d(x=v_viento[0:(idMax + 1)],
338.                     y=potencia[0:(idMax + 1)], kind=tipoInterp)
339.                     generacion1 = funcionInterp_cubica(e)
340.                 else:
341.                     generacion1 = potencia.max()
342.             potencia_datos_altura[i] = generacion1
343.         datos_altura['potencia_datos']=potencia_datos_altura.copy()
344.         return datos_altura
345.
346.
347.
348. #####
349. #
350. # Main
351. #
352. #####
353.
354.
355. #.....
356. #Aerogenerador del parque eólico: Gamesa 660
357. #.....
358. #La variable "v" contiene los valores de velocidad del viento de 0.5 en 0.5(velocidades en las
359. #que funciona el aerogenerador)
360. v=[x*0.5 for x in range(9,51)]
361. print(v)
362.
363. #Datos aerogenerador Gamesa 660kW
364. v_viento = [4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5, 11, 11.5, 12, 12.5, 13,
365.             13.5, 14, 14.5, 15, 15.5, 16, 16.5, 17, 17.5, 18, 18.5, 19, 19.5, 20, 20.5, 21, 21.5, 22,
366.             22.5, 23, 23.5, 24, 24.5, 25]

```

```

366. potencia = [26,55,78, 105, 133, 165, 209, 254, 301, 347, 405, 462, 509, 556, 593, 629, 649,
667, 679, 690, 693, 696, 698, 700, 700, 700, 700, 700, 700, 700, 700, 700, 700, 700, 700, 700, 700, 700, 700]
367.
368. v_viento= np.array(v_viento)
369. potencia = np.array(potencia)
370.
371. if v_viento.shape[0] != potencia.shape[0]:
372.     print('Error!!! The inputs arrays have different shapes.')
373.
374.
375. f_funcion= interpolate.interp1d(x=v_viento, y=potencia, kind='linear')
376.
377. f_funcion_cubic=interpolate.interp1d(x=v_viento, y=potencia, kind='cubic')
378.
379.
380.
381. # Gráfico de curva de potencia Gamesa:
382. x_new = np.arange(4.5, 25, 0.1)
383. y_linear=f_funcion(x_new)
384. y_cubic=f_funcion_cubic(x_new)
385.
386. plt.plot(v_viento, potencia, 'bo', label='Gamesa')
387. plt.plot(x_new, y_linear, 'r-', label='Linear_interp')
388. plt.grid('on', ls='--')
389. plt.legend()
390. plt.xlabel('v [m/s]')
391. plt.ylabel('P [kW]')
392. plt.title('Curva potencia Gamesa')
393. plt.show()
394.
395.
396. #.....
397. # Fichero Sierra del Trigo:
398. #.....
399.
400.
401. df_raw = pd.read_table('Todos-sdt1509-22_06_21_20210622_092000.WND', delimiter='\s+', skiprows
= 3 , na_values='-', parse_dates=[[ 'Date', 'Time' ]], index_col=[ 'Date_Time' ], decimal=',')
402.
403.
404. #.....
405. # Aplicación de máscara
406. #.....
407.
408. df_mascara=df_raw
409.
410. df2=df_mascara[(df_mascara['WS1-STDev']!=0.0) & (df_mascara['WS2-STDev']!=0.0)]
411. #df2=df_mascara[(df_mascara['WS1-STDev']==0.0) | (df_mascara['WS2-STDev']==0.0)]
412.
413. #Para conocer con qué porcentaje de datos del dataFrame iniciales trabajamos:
414. print((len(df2)/len(df_mascara))*100)
415. # Nos quedamos con el 96.27% de los datos iniciales
416.
417. df=df2
418.
419.
420. #.....
421. # Análisis de datos de torre:
422. #.....
423.
424. #Media de velocidades del viento:
425.
426. WS1_media = df['WS1'].mean()
427. WS2_media = df['WS2'].mean()
428.
429.
430. #df = df.dropna(1)
431.
432.
433. media_horaria = df.resample('1H').mean()
434.
435.
436. #Histograma de velocidad del viento de Sierra del Trigo a 20m de altura:
437.
438. media_horaria.WS1.plot.hist(bins=np.arange(0.35, 32, 0.1), density=True)
439.
440. plt.title('Histograma velocidad del viento')
441. plt.xlabel('v (velocidad viento)')

```

```

442. plt.ylabel('Frecuencia')
443. x_new = np.arange(0, 33, 2)
444. #intervalos=( 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
23, 24, 25, 26, 27, 28, 29, 30, 31, 32)
445. #plt.xticks(x_new)
446. plt.show()
447.
448.
449. #Gráfica de caja de bigotes de WS1 y WS2:
450.
451.
452. boxplot = media_horaria.boxplot(column=['WS1','WS2'], return_type= 'axes')
453.
454. boxplot.set_ylabel('v (m/s)')
455.
456.
457. #Cajas y bigotes por la noche:
458. media_horaria_noche = media_horaria.between_time('00:00:00', '06:00:00')
459. boxplot = media_horaria_noche.boxplot(column=['WS1','WS2'])
460.
461.
462.
463. #Histograma de velocidad del viento de Sierra del Trigo a 40m de altura:
464.
465. media_horaria.WS2.plot.hist(bins=np.arange(0.35, 32, 0.1), density=True)
466.
467. plt.title('Histograma velocidad del viento(WS2)')
468. plt.xlabel('v (velocidad viento)')
469. plt.ylabel('Frecuencia')
470. x_new = np.arange(0, 33, 2)
471. #intervalos=( 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
23, 24, 25, 26, 27, 28, 29, 30, 31, 32)
472. #plt.xticks(x_new)
473. plt.show()
474.
475.
476. #Rosa de los vientos del parque de Sierra del Trigo a 20m de altura:
477.
478. WD1= np.array(media_horaria.WD1.values)
479. WD1_t = []
480. WS1_t = []
481. for x in WD1:
482.     if pd.isnull(x)==False:
483.         WD1_t.append(x)
484. WS1= np.array(media_horaria.WS1.values)
485. for x in WS1:
486.     if pd.isnull(x)==False:
487.         WS1_t.append(x)
488.
489. ax = WindroseAxes.from_ax()
490. ax.bar(WD1_t, WS1_t,normed=True, opening=0.8, edgecolor='white', bins=[0.,5.,10.,20.,30.])
491. ax.set_legend(prop = { 'size': 18})
492. ax.set_title('WS1', fontsize = 18)
493.
494.
495. #Rosa de los vientos del parque de Sierra del Trigo a 40 metros de altura:
496.
497. WD2= np.array(media_horaria.WD2.values)
498. WD2_t = []
499. WS2_t = []
500. for x in WD2:
501.     if pd.isnull(x)==False:
502.         WD2_t.append(x)
503. WS2= np.array(media_horaria.WS2.values)
504. for x in WS2:
505.     if pd.isnull(x)==False:
506.         WS2_t.append(x)
507.
508. ax = WindroseAxes.from_ax()
509. ax.bar(WD2_t, WS2_t,normed=True, opening=0.8, edgecolor='white', bins=[0.,5.,10.,20.,30.])
510. ax.set_legend( prop = { 'size': 18})
511. ax.set_title('WS2', fontsize=18)
512.
513. #.....
514. #Potencia producida con aerogenerador Gamesa 660
515. #.....
516.
517. #A partir de la velocidad del viento, conseguir la potencia en Sierra del Trigo a 20 m de
altura:

```

```

518.
519. datos_20m = pd.DataFrame(df.WS1).copy()
520.
521. potencia_datos_20m = np.zeros(len(datos_20m))
522.
523. for i in range(len(datos_20m)):
524.     d = datos_20m['WS1'][i]
525.     pot = GamesaGeneracion(d)
526.     potencia_datos_20m[i] = pot
527.
528. datos_20m['potencia_datos']=potencia_datos_20m.copy()
529.
530.
531.
532. #A partir de la velocidad del viento, conseguir la potencia en Sierra del Trigo a 40 m de
    altura:
533.
534. datos_40m = pd.DataFrame(df.WS2).copy()
535.
536. potencia_datos_40m = np.zeros(len(datos_40m))
537.
538. for i in range(len(datos_40m)):
539.     d = datos_40m['WS2'][i]
540.     pot = GamesaGeneracion(d)
541.     potencia_datos_40m[i] = pot
542.
543. datos_40m['potencia_datos']=potencia_datos_40m.copy()
544.
545.
546. #.....
547. #Media horaria de los datos de torre
548. #.....
549.
550.
551. Sierra_trigo_horario_20m = datos_20m.resample('1H').mean()
552. Sierra_trigo_horario_20m = Sierra_trigo_horario_20m.dropna()
553. Sierra_trigo_horario_40m = datos_40m.resample('1H').mean()
554. Sierra_trigo_horario_40m = Sierra_trigo_horario_40m.dropna()
555.
556.
557.
558.
559. #.....
560. #Fichero ERA5:
561. #.....
562.
563. archivo_ERA5= 'ModuloViento10m100mERA5_SierraTrigo.csv'
564.
565. df_ERA5 = pd.read_csv(archivo_ERA5, sep=';', index_col='Date', parse_dates = True, dtype =
    {'Date': object, 'v10m': float, 'v100m': float})
566.
567. ERA5 = df_ERA5.copy()
568.
569.
570.
571. #.....
572. #Separación de modelo y corrección:
573. #.....
574.
575. #Se llama a la función dias_datos con la finalidad de obtener los días de modelo y corrección
    del mismo:
576.
577.
578. #Sierra del Trigo:
579. # Debido a que no podemos trabajar con los índices, se lo quitamos y luego se vuelven a
    añadir:
580.
581. modelo_sierra_20m, validacion_sierra_20m = modelo_validacion(datos_20m, 'Date_Time')
582. modelo_sierra_20m, validacion_sierra_20m = modelo_sierra_20m.resample('1H').mean(),
    validacion_sierra_20m.resample('1H').mean()
583. modelo_sierra_20m, validacion_sierra_20m = modelo_sierra_20m.dropna(),
    validacion_sierra_20m.dropna()
584.
585. modelo_sierra_40m, validacion_sierra_40m = modelo_validacion(datos_40m, 'Date_Time')
586. modelo_sierra_40m, validacion_sierra_40m = modelo_sierra_40m.resample('1H').mean(),
    validacion_sierra_40m.resample('1H').mean()
587. modelo_sierra_40m, validacion_sierra_40m = modelo_sierra_40m.dropna(),
    validacion_sierra_40m.dropna()
588.

```

```

589.
590.
591. #.....
592. #Interpolación potencial:
593. #.....
594.
595.
596.
597. #Ley potencial:
598. #         v2 = v1*(z2/z1)^a
599.
600. #          $\alpha = (\ln(v2)-\ln(v1)) / (\ln(z2)-\ln(z1))$ 
601.
602. #         Para sacar a, usar las velocidades (v1 y v2) a z1=20m y z2= 40m respectivamente
        (z1 y z2 son las alturas de Sierra del Trigo).
603.
604.
605.
606. #.....
607. # Cálculo de  $\alpha$ 
608. #.....
609.
610. # Se sacaron los valores alfa teniendo en cuenta los días del 1 al 20 de cada mes (modelo):
611.
612.  $\alpha\_sierra = (\text{np.log}(\text{modelo\_sierra\_40m.WS2}) - \text{np.log}(\text{modelo\_sierra\_20m.WS1})) / (\text{np.log}(40) - \text{np.log}(20))$ 
613.
614.  $\alpha\_anual = \text{np.mean}(\alpha\_sierra)$ 
615.
616.  $\alpha\_dia, \alpha\_noche = \alpha\_dia\_noche(\alpha\_sierra)$ 
617.
618. invierno, primavera, verano, otoño = estaciones( $\alpha\_sierra$ , 'Date_Time')
619.
620.  $\alpha\_estaciones = \alpha\_estacional(\text{invierno}, \text{primavera}, \text{verano}, \text{otoño})$ 
621.
622.
623.
624. #.....
625. #Extrapolación de viento en altura a 40 metros
626. #.....
627.
628.  $v2\_40m = \text{ley\_potencial}(\text{validacion\_sierra\_20m.WS1}, \text{'Date\_Time'}, \alpha\_estaciones, 20., 40.)$ 
629.  $v2\_40m.columns = ['v2\_40m']$ 
630.
631.
632. #.....
633. #Paso previo para calcular los estadísticos
634. #.....
635.
636.  $df\_40m = \text{pd.merge}(v2\_40m, \text{validacion\_sierra\_40m}, \text{left\_index=True}, \text{right\_index=True})$ 
637.
638. # Comprobar que no hay valores NaN.
639.  $\text{if}(df\_40m.v2\_40m[df\_40m.v2\_40m.isna()].shape[0] != 0) \text{ and}$ 
        ( $df\_40m.WS2[df\_40m.WS2.isna()].shape[0] != 0$ ):
640.     print('Hay valores NaN, ¡Cuidado!')
641. else:
642.     print('No hay valores NaN en los datos a evaluar')
643.
644.
645.  $y\_true\_validacion = df\_40m.WS2.values.copy()$ 
646.  $y\_pred\_validacion = df\_40m.v2\_40m.values.copy()$ 
647.
648.
649. #.....
650. # RMSE de validación
651. #.....
652.
653.  $\text{RMSE}(y\_true\_validacion, y\_pred\_validacion)$ 
654.
655. # RMSE de la validación: 1.1047503588297638 m/s
656. # Debido a los bajos valores de  $\alpha$  (casi 0), el RMSE a 40m se extrapola a 100m, por lo que el
        RMSE a 100m es también 1.1047503588297638 m/s.
657.
658.
659. #.....
660. #Extrapolación de viento en altura a 100 metros
661. #.....
662.
663. # A partir de datos de 20m:
664.  $v\_20m\_100m = \text{ley\_potencial}(\text{Sierra\_trigo\_horario\_20m.WS1}, \text{'Date\_Time'}, \alpha\_estaciones, 20., 100.)$ 

```

[illegible]

```

709. potencia_100m_ERA5_ventas = Generacion(ERA5['v100m'].loc['2020-01-01 00:00:00' : '2020-12-31
23:00:00'], v_min=3.5, v_max=25., v_corte=25., v_viento=v_viento1, potencia=p_ventas)
710. potencia_100m_ERA5_gamesa = Generacion(ERA5['v100m'].loc['2020-01-01 00:00:00' : '2020-12-31
23:00:00'], v_min=2., v_max=27., v_corte=18.5, v_viento=v_viento1, potencia=p_gamesa)
711. potencia_100m_ERA5_sg = Generacion(ERA5['v100m'].loc['2020-01-01 00:00:00' : '2020-12-31
23:00:00'], v_min=2.5, v_max=25.5, v_corte=21.5, v_viento=v_viento1, potencia=p_sg)
712.
713.
714. #.....
715. # Capacity Factor horario del año 2020
716. #.....
717.
718. CF_sierra_enercon = potencia_100m_sierra_enercon['potencia_datos'].loc['2020-01-01 00:00:00' :
'2020-12-31 23:00:00']/2300
719. CF_sierra_nordex = potencia_100m_sierra_nordex['potencia_datos'].loc['2020-01-01 00:00:00' :
'2020-12-31 23:00:00']/2500
720. CF_sierra_repower = potencia_100m_sierra_repower['potencia_datos'].loc['2020-01-01 00:00:00' :
'2020-12-31 23:00:00']/2000
721. CF_sierra_ventas = potencia_100m_sierra_ventas['potencia_datos'].loc['2020-01-01 00:00:00' :
'2020-12-31 23:00:00']/3000
722. CF_sierra_gamesa = potencia_100m_sierra_gamesa['potencia_datos'].loc['2020-01-01 00:00:00' :
'2020-12-31 23:00:00']/4500
723. CF_sierra_sg = potencia_100m_sierra_sg['potencia_datos'].loc['2020-01-01 00:00:00' : '2020-12-
31 23:00:00']/2100
724.
725. CF_ERA5_enercon = potencia_100m_ERA5_enercon['potencia_datos'].loc['2020-01-01 00:00:00' :
'2020-12-31 23:00:00']/2300
726. CF_ERA5_nordex = potencia_100m_ERA5_nordex['potencia_datos'].loc['2020-01-01 00:00:00' :
'2020-12-31 23:00:00']/2500
727. CF_ERA5_repower = potencia_100m_ERA5_repower['potencia_datos'].loc['2020-01-01 00:00:00' :
'2020-12-31 23:00:00']/2000
728. CF_ERA5_ventas = potencia_100m_ERA5_ventas['potencia_datos'].loc['2020-01-01 00:00:00' :
'2020-12-31 23:00:00']/3000
729. CF_ERA5_gamesa = potencia_100m_ERA5_gamesa['potencia_datos'].loc['2020-01-01 00:00:00' :
'2020-12-31 23:00:00']/4500
730. CF_ERA5_sg = potencia_100m_ERA5_sg['potencia_datos'].loc['2020-01-01 00:00:00' : '2020-12-31
23:00:00']/2100
731.
732.
733. #.....
734. # Capacity Factor medio del año 2020
735. #.....
736.
737. CF_sierra_enercon_medio = CF_sierra_enercon.mean()
738. CF_sierra_nordex_medio = CF_sierra_nordex.mean()
739. CF_sierra_repower_medio = CF_sierra_repower.mean()
740. CF_sierra_ventas_medio = CF_sierra_ventas.mean()
741. CF_sierra_gamesa_medio = CF_sierra_gamesa.mean()
742. CF_sierra_sg_medio = CF_sierra_sg.mean()
743.
744. CF_ERA5_enercon_medio = CF_ERA5_enercon.mean()
745. CF_ERA5_nordex_medio = CF_ERA5_nordex.mean()
746. CF_ERA5_repower_medio = CF_ERA5_repower.mean()
747. CF_ERA5_ventas_medio = CF_ERA5_ventas.mean()
748. CF_ERA5_gamesa_medio = CF_ERA5_gamesa.mean()
749. CF_ERA5_sg_medio = CF_ERA5_sg.mean()
750.
751.
752. #.....
753. # Desviación estándar del Capacity Factor horario del año 2020
754. #.....
755.
756. dev_CF_sierra_enercon = np.std(CF_sierra_enercon)
757. dev_CF_sierra_nordex = np.std(CF_sierra_nordex)
758. dev_CF_sierra_repower = np.std(CF_sierra_repower)
759. dev_CF_sierra_ventas = np.std(CF_sierra_ventas)
760. dev_CF_sierra_gamesa = np.std(CF_sierra_gamesa)
761. dev_CF_sierra_sg = np.std(CF_sierra_sg)
762.
763. dev_CF_ERA5_enercon = np.std(CF_ERA5_enercon)
764. dev_CF_ERA5_nordex = np.std(CF_ERA5_nordex)
765. dev_CF_ERA5_repower = np.std(CF_ERA5_repower)
766. dev_CF_ERA5_ventas = np.std(CF_ERA5_ventas)
767. dev_CF_ERA5_gamesa = np.std(CF_ERA5_gamesa)
768. dev_CF_ERA5_sg = np.std(CF_ERA5_sg)
769.
770.
771. #.....
772. #Potencia real producida horaria del año 2020

```

```

773. #.....
774.
775. P_sierra_enercon = potencia_100m_sierra_enercon['potencia_datos'] * CF_sierra_enercon
776. P_sierra_nordex = potencia_100m_sierra_nordex['potencia_datos'] * CF_sierra_nordex
777. P_sierra_repower = potencia_100m_sierra_repower['potencia_datos'] * CF_sierra_repower
778. P_sierra_vestas = potencia_100m_sierra_vestas['potencia_datos'] * CF_sierra_vestas
779. P_sierra_gamesa = potencia_100m_sierra_gamesa['potencia_datos'] * CF_sierra_gamesa
780. P_sierra_sg = potencia_100m_sierra_sg['potencia_datos'] * CF_sierra_sg
781.
782. P_ERA5_enercon = potencia_100m_ERA5_enercon['potencia_datos'] * CF_ERA5_enercon
783. P_ERA5_nordex = potencia_100m_ERA5_nordex['potencia_datos'] * CF_ERA5_nordex
784. P_ERA5_repower = potencia_100m_ERA5_repower['potencia_datos'] * CF_ERA5_repower
785. P_ERA5_vestas = potencia_100m_ERA5_vestas['potencia_datos'] * CF_ERA5_vestas
786. P_ERA5_gamesa = potencia_100m_ERA5_gamesa['potencia_datos'] * CF_ERA5_gamesa
787. P_ERA5_sg = potencia_100m_ERA5_sg['potencia_datos'] * CF_ERA5_sg
788.
789.
790.
791.
792. #*****
793. # Estadísticos de CF sierra y ERA5:
794. #*****
795.
796. df_conjunto_CF_enercon= pd.merge(CF_sierra_enercon, CF_ERA5_enercon, left_index=True,
right_index=True)
797. df_conjunto_CF_nordex= pd.merge(CF_sierra_nordex, CF_ERA5_nordex, left_index=True,
right_index=True)
798. df_conjunto_CF_repower= pd.merge(CF_sierra_repower, CF_ERA5_repower, left_index=True,
right_index=True)
799. df_conjunto_CF_vestas= pd.merge(CF_sierra_vestas, CF_ERA5_vestas, left_index=True,
right_index=True)
800. df_conjunto_CF_gamesa= pd.merge(CF_sierra_gamesa, CF_ERA5_gamesa, left_index=True,
right_index=True)
801. df_conjunto_CF_sg= pd.merge(CF_sierra_sg, CF_ERA5_sg, left_index=True, right_index=True)
802.
803.
804.
805. # y_true_CF = df_conjunto_CF_XXXXXX['potencia_datos_x'].value.copy()
806. # y_pred_CF = df_conjunto_CF_XXXXXX['potencia_datos_y'].value.copy()
807.
808. # RMSE anual:
809.
810. RMSE_CF_enercon = RMSE(df_conjunto_CF_enercon['potencia_datos_x'].copy(),
df_conjunto_CF_enercon['potencia_datos_y'].copy())
811. RMSE_CF_nordex = RMSE(df_conjunto_CF_nordex['potencia_datos_x'].copy(),
df_conjunto_CF_nordex['potencia_datos_y'].copy())
812. RMSE_CF_repower = RMSE(df_conjunto_CF_repower['potencia_datos_x'].copy(),
df_conjunto_CF_repower['potencia_datos_y'].copy())
813. RMSE_CF_vestas = RMSE(df_conjunto_CF_vestas['potencia_datos_x'].copy(),
df_conjunto_CF_vestas['potencia_datos_y'].copy())
814. RMSE_CF_gamesa = RMSE(df_conjunto_CF_gamesa['potencia_datos_x'].copy(),
df_conjunto_CF_gamesa['potencia_datos_y'].copy())
815. RMSE_CF_sg = RMSE(df_conjunto_CF_sg['potencia_datos_x'].copy(),
df_conjunto_CF_sg['potencia_datos_y'].copy())
816.
817.
818. # RMSE día-noche:
819. #Enercon:
820. dia_CF_enercon, noche_CF_enercon = dia_noche(df_conjunto_CF_enercon)
821. dia_RMSE_CF_enercon = RMSE(dia_CF_enercon['potencia_datos_x'].copy(),
dia_CF_enercon['potencia_datos_y'].copy())
822. noche_RMSE_CF_enercon = RMSE(noche_CF_enercon['potencia_datos_x'].copy(),
noche_CF_enercon['potencia_datos_y'].copy())
823.
824. #Nordex:
825. dia_CF_nordex, noche_CF_nordex = dia_noche(df_conjunto_CF_nordex)
826. dia_RMSE_CF_nordex = RMSE(dia_CF_nordex['potencia_datos_x'].copy(),
dia_CF_nordex['potencia_datos_y'].copy())
827. noche_RMSE_CF_nordex = RMSE(noche_CF_nordex['potencia_datos_x'].copy(),
noche_CF_nordex['potencia_datos_y'].copy())
828.
829. #Repower:
830. dia_CF_repower, noche_CF_repower = dia_noche(df_conjunto_CF_repower)
831. dia_RMSE_CF_repower = RMSE(dia_CF_repower['potencia_datos_x'].copy(),
dia_CF_repower['potencia_datos_y'].copy())
832. noche_RMSE_CF_repower = RMSE(noche_CF_repower['potencia_datos_x'].copy(),
noche_CF_repower['potencia_datos_y'].copy())
833.
834. #Vestas:

```



```

835. dia_CF_vestas, noche_CF_vestas = dia_noche(df_conjunto_CF_vestas)
836. dia_RMSE_CF_vestas = RMSE(dia_CF_vestas['potencia_datos_x'].copy(),
    dia_CF_vestas['potencia_datos_y'].copy())
837. noche_RMSE_CF_vestas = RMSE(noche_CF_vestas['potencia_datos_x'].copy(),
    noche_CF_vestas['potencia_datos_y'].copy())
838.
839.     #Gamesa:
840. dia_CF_gamesa, noche_CF_gamesa = dia_noche(df_conjunto_CF_gamesa)
841. dia_RMSE_CF_gamesa = RMSE(dia_CF_gamesa['potencia_datos_x'].copy(),
    dia_CF_gamesa['potencia_datos_y'].copy())
842. noche_RMSE_CF_gamesa = RMSE(noche_CF_gamesa['potencia_datos_x'].copy(),
    noche_CF_gamesa['potencia_datos_y'].copy())
843.
844.     #Sg:
845. dia_CF_sg, noche_CF_sg = dia_noche(df_conjunto_CF_sg)
846. dia_RMSE_CF_sg = RMSE(dia_CF_sg['potencia_datos_x'].copy(),
    dia_CF_sg['potencia_datos_y'].copy())
847. noche_RMSE_CF_sg = RMSE(noche_CF_sg['potencia_datos_x'].copy(),
    noche_CF_sg['potencia_datos_y'].copy())
848.
849.
850. # RMSE estaciones:
851.     #Enercon:
852. invierno_CF_enercon, primavera_CF_enercon, verano_CF_enercon, otoño_CF_enercon =
    estaciones(df_conjunto_CF_enercon, 'index')
853. invierno_RMSE_CF_enercon = RMSE(invierno_CF_enercon['potencia_datos_x'].copy(),
    invierno_CF_enercon['potencia_datos_y'].copy())
854. primavera_RMSE_CF_enercon = RMSE(primavera_CF_enercon['potencia_datos_x'].copy(),
    primavera_CF_enercon['potencia_datos_y'].copy())
855. verano_RMSE_CF_enercon = RMSE(verano_CF_enercon['potencia_datos_x'].copy(),
    verano_CF_enercon['potencia_datos_y'].copy())
856. otoño_RMSE_CF_enercon = RMSE(otoño_CF_enercon['potencia_datos_x'].copy(),
    otoño_CF_enercon['potencia_datos_y'].copy())
857.
858.     #Nordex:
859. invierno_CF_nordex, primavera_CF_nordex, verano_CF_nordex, otoño_CF_nordex =
    estaciones(df_conjunto_CF_nordex, 'index')
860. invierno_RMSE_CF_nordex = RMSE(invierno_CF_nordex['potencia_datos_x'].copy(),
    invierno_CF_nordex['potencia_datos_y'].copy())
861. primavera_RMSE_CF_nordex = RMSE(primavera_CF_nordex['potencia_datos_x'].copy(),
    primavera_CF_nordex['potencia_datos_y'].copy())
862. verano_RMSE_CF_nordex = RMSE(verano_CF_nordex['potencia_datos_x'].copy(),
    verano_CF_nordex['potencia_datos_y'].copy())
863. otoño_RMSE_CF_nordex = RMSE(otoño_CF_nordex['potencia_datos_x'].copy(),
    otoño_CF_nordex['potencia_datos_y'].copy())
864.
865.     #Repower:
866. invierno_CF_repower, primavera_CF_repower, verano_CF_repower, otoño_CF_repower =
    estaciones(df_conjunto_CF_repower, 'index')
867. invierno_RMSE_CF_repower = RMSE(invierno_CF_repower['potencia_datos_x'].copy(),
    invierno_CF_repower['potencia_datos_y'].copy())
868. primavera_RMSE_CF_repower = RMSE(primavera_CF_repower['potencia_datos_x'].copy(),
    primavera_CF_repower['potencia_datos_y'].copy())
869. verano_RMSE_CF_repower = RMSE(verano_CF_repower['potencia_datos_x'].copy(),
    verano_CF_repower['potencia_datos_y'].copy())
870. otoño_RMSE_CF_repower = RMSE(otoño_CF_repower['potencia_datos_x'].copy(),
    otoño_CF_repower['potencia_datos_y'].copy())
871.
872.     #Vestas:
873. invierno_CF_vestas, primavera_CF_vestas, verano_CF_vestas, otoño_CF_vestas =
    estaciones(df_conjunto_CF_vestas, 'index')
874. invierno_RMSE_CF_vestas = RMSE(invierno_CF_vestas['potencia_datos_x'].copy(),
    invierno_CF_vestas['potencia_datos_y'].copy())
875. primavera_RMSE_CF_vestas = RMSE(primavera_CF_vestas['potencia_datos_x'].copy(),
    primavera_CF_vestas['potencia_datos_y'].copy())
876. verano_RMSE_CF_vestas = RMSE(verano_CF_vestas['potencia_datos_x'].copy(),
    verano_CF_vestas['potencia_datos_y'].copy())
877. otoño_RMSE_CF_vestas = RMSE(otoño_CF_vestas['potencia_datos_x'].copy(),
    otoño_CF_vestas['potencia_datos_y'].copy())
878.
879.     #Gamesa:
880. invierno_CF_gamesa, primavera_CF_gamesa, verano_CF_gamesa, otoño_CF_gamesa =
    estaciones(df_conjunto_CF_gamesa, 'index')
881. invierno_RMSE_CF_gamesa = RMSE(invierno_CF_gamesa['potencia_datos_x'].copy(),
    invierno_CF_gamesa['potencia_datos_y'].copy())
882. primavera_RMSE_CF_gamesa = RMSE(primavera_CF_gamesa['potencia_datos_x'].copy(),
    primavera_CF_gamesa['potencia_datos_y'].copy())
883. verano_RMSE_CF_gamesa = RMSE(verano_CF_gamesa['potencia_datos_x'].copy(),
    verano_CF_gamesa['potencia_datos_y'].copy())

```

```

884. otoño_RMSE_CF_gamesa = RMSE(otoño_CF_gamesa['potencia_datos_x'].copy(),
    otoño_CF_gamesa['potencia_datos_y'].copy())
885.
886.     #Sg:
887. invierno_CF_sg, primavera_CF_sg, verano_CF_sg, otoño_CF_sg = estaciones(df_conjunto_CF_sg,
    'index')
888. invierno_RMSE_CF_sg = RMSE(invierno_CF_sg['potencia_datos_x'].copy(),
    invierno_CF_sg['potencia_datos_y'].copy())
889. primavera_RMSE_CF_sg = RMSE(primavera_CF_sg['potencia_datos_x'].copy(),
    primavera_CF_sg['potencia_datos_y'].copy())
890. verano_RMSE_CF_sg = RMSE(verano_CF_sg['potencia_datos_x'].copy(),
    verano_CF_sg['potencia_datos_y'].copy())
891. otoño_RMSE_CF_sg = RMSE(otoño_CF_sg['potencia_datos_x'].copy(),
    otoño_CF_sg['potencia_datos_y'].copy())
892.
893.
894. # BIAS anual:
895.
896. BIAS_CF_enercon =BIAS(df_conjunto_CF_enercon['potencia_datos_x'].copy(),
    df_conjunto_CF_enercon['potencia_datos_y'].copy())
897. BIAS_CF_nordex = BIAS(df_conjunto_CF_nordex['potencia_datos_x'].copy(),
    df_conjunto_CF_nordex['potencia_datos_y'].copy())
898. BIAS_CF_repower = BIAS(df_conjunto_CF_repower['potencia_datos_x'].copy(),
    df_conjunto_CF_repower['potencia_datos_y'].copy())
899. BIAS_CF_vestas = BIAS(df_conjunto_CF_vestas['potencia_datos_x'].copy(),
    df_conjunto_CF_vestas['potencia_datos_y'].copy())
900. BIAS_CF_gamesa = BIAS(df_conjunto_CF_gamesa['potencia_datos_x'].copy(),
    df_conjunto_CF_gamesa['potencia_datos_y'].copy())
901. BIAS_CF_sg = BIAS(df_conjunto_CF_sg['potencia_datos_x'].copy(),
    df_conjunto_CF_sg['potencia_datos_y'].copy())
902.
903.
904. #####
905. # Energía 2020 horaria:
906. #####
907. #      E anual = CF*8760*Pn --> Horaria =CF*Pn      La energía está en kWh
908.
909. E_sierra_enercon = P_sierra_enercon*1
910. E_sierra_nordex = P_sierra_nordex *1
911. E_sierra_repower = P_sierra_repower *1
912. E_sierra_vestas = P_sierra_vestas *1
913. E_sierra_gamesa = P_sierra_gamesa *1
914. E_sierra_sg = P_sierra_sg *1
915.
916. E_ERA5_enercon = P_ERA5_enercon *1
917. E_ERA5_nordex = P_ERA5_nordex *1
918. E_ERA5_repower = P_ERA5_repower *1
919. E_ERA5_vestas = P_ERA5_vestas *1
920. E_ERA5_gamesa = P_ERA5_gamesa *1
921. E_ERA5_sg = P_ERA5_sg *1
922.
923.
924. # Energía anual MWh:
925. E_anual_sierra_enercon = (E_sierra_enercon.sum())/1000
926. E_anual_sierra_nordex = (E_sierra_nordex.sum())/1000
927. E_anual_sierra_repower = (E_sierra_repower.sum())/1000
928. E_anual_sierra_vestas = (E_sierra_vestas.sum())/1000
929. E_anual_sierra_gamesa = (E_sierra_gamesa.sum())/1000
930. E_anual_sierra_sg = (E_sierra_sg.sum())/1000
931.
932. E_anual_ERA5_enercon = (E_ERA5_enercon.sum())/1000
933. E_anual_ERA5_nordex = (E_ERA5_nordex.sum())/1000
934. E_anual_ERA5_repower = (E_ERA5_repower.sum())/1000
935. E_anual_ERA5_vestas = (E_ERA5_vestas.sum())/1000
936. E_anual_ERA5_gamesa = (E_ERA5_gamesa.sum())/1000
937. E_anual_ERA5_sg = (E_ERA5_sg.sum())/1000
938.
939.
940.
941. #.....
942. # Gráficos adicionales
943. #.....
944.
945. plt.plot (E_sierra_enercon, label ='Energía Sierra del Trigo')
946. plt.plot (E_ERA5_enercon, label = 'Energía ERA5')
947. plt.title ('Energía Enercon')
948. plt.xlabel('Tiempo')
949. plt.ylabel('Energía (kw)')
950. plt.legend(loc=0)

```

```

951. plt.show()
952.
953.
954.
955. #Histograma viento Torre vs. a ERA-5:
956.
957. pyplot.hist(ERA5['v100m'], bins=np.arange(0, 32.5, 0.1), alpha = 1 ,density=True, label =
    'ERA5')
958. pyplot.hist(v_40m_100m['v_100m'], bins=np.arange(0, 32.5, 0.1), alpha = 0.8, density=True,
    label = 'Datos torre')
959. pyplot.title('Viento a 100m Torre - ERA5')
960. pyplot.xlabel('v (velocidad viento)')
961. pyplot.ylabel('Frecuencia')
962. pyplot.legend(loc = 'upper right')
963. pyplot.show()
964.
965.
966. #Caja bigotes Sierra -ERA5:
967. caja_100m = pd.merge(ERA5['v100m'], v_40m_100m['v_100m'], left_index=True, right_index=True)
968. caja_100m.columns=['ERA5', 'Torre']
969. boxplot = caja_100m.boxplot(column=['ERA5', 'Torre'])
970. boxplot.set_ylabel('v (m/s)')
971.
972.
973. formatter1=mdates.DateFormatter('%H')
974.
975. fig, ax =plt.subplots()
976.
977. ax.plot(media_horaria['WS2'].loc['2020-05-15'].index,media_horaria['WS2'].loc['2020-05-15'],
    label = 'Datos horarios')
978. ax.set_title('Viento el día 15-05-2020 ')
979. ax.legend(loc=0)
980. ax.set_xlabel('Hora')
981. ax.set_ylabel('Viento (m/s)')
982. plt.xticks(rotation=60)
983. ax.xaxis.set_major_formatter(formatter1)
984.
985.
986. formatter=mdates.DateFormatter('%H:%M')
987. fig, ax =plt.subplots()
988. ax.plot(df['WS2'].loc['2020-05-15 00:00:00': '2020-05-15 23:50:00'], label = 'Datos diez
    minutales')
989. ax.plot(media_horaria['WS2'].loc['2020-05-15'].index,media_horaria['WS2'].loc['2020-05-15'],
    label = 'Datos horarios')
990. ax.set_title('Viento el día 15-05-2020 ')
991. ax.legend(loc=0)
992. ax.set_xlabel('Horas')
993. ax.set_ylabel('Viento (m/s)')
994. plt.xticks(rotation=60)
995. ax.xaxis.set_major_formatter(formatter)
996.
997.
998.
999.
1000. plt.plot(CF_sierra_enercon, label='Sierra')
1001. plt.plot(CF_ERA5_enercon,
    label='ERA5')
1002.
1003. plt.title('Potencia generada en Jaén (20m)')
1004. plt.xlabel('Fecha')
1005. plt.ylabel('Generación (MWh)')
1006. plt.legend(loc=0)
1007.
1008. plt.show()

```