

---

# Predicción de Saturación Hospitalaria en España

---

Memoria del proyecto

**Autor:** Francisco Javier Gómez Pulido

Correo electrónico: [frangomezpulido2002@gmail.com](mailto:frangomezpulido2002@gmail.com)

LinkedIn: [linkedin.com/in/frangomezpulido](https://www.linkedin.com/in/frangomezpulido)

GitHub: [github.com/fragompul](https://github.com/fragompul)

## **Resumen**

En este proyecto se desarrolla un modelo predictivo para estimar la saturación hospitalaria en España utilizando técnicas de Ciencia de Datos. Se emplean datos históricos de ocupación hospitalaria, variables epidemiológicas y modelos de aprendizaje automático para anticipar situaciones críticas y ayudar en la planificación sanitaria.

# Índice general

<b>1. Introducción</b>	<b>3</b>
1.1. Contexto y Motivación	3
1.2. Objetivos del Proyecto	3
1.3. Estructura del Documento	4
<b>2. Revisión del Estado del Arte</b>	<b>5</b>
2.1. Saturación Hospitalaria: Definición y Causas	5
2.2. Modelos Predictivos en Salud Pública	5
2.3. Proyectos Similares	6
<b>3. Datos Utilizados</b>	<b>8</b>
3.1. Fuentes de Datos	8
3.2. Descripción de Variables	8
3.3. Preprocesamiento	9
3.4. Observaciones Iniciales	10
<b>4. Metodología</b>	<b>11</b>
4.1. Planteamiento del Problema	11
4.2. Modelos Utilizados	11
4.2.1. Modelos estadísticos clásicos	11
4.2.2. Modelos de machine learning	12
4.2.3. Prophet	12
4.3. Implementación con Prophet	12
4.4. Validación	13
4.5. Visualización de Resultados	14
4.6. Consideraciones y Limitaciones del Modelo	14
<b>5. Resultados</b>	<b>15</b>
5.1. Desempeño de los Modelos	15
5.2. Interpretación de Resultados	15
<b>6. Implementación</b>	<b>16</b>
6.1. Despliegue del Sistema	16
6.2. Estructura del Dashboard	16
6.3. Integración del Modelo Prophet	17
6.4. Visualización Interactiva	17
6.5. Interacción con el Usuario	19

<b>7. Conclusiones y Trabajo Futuro</b>	<b>20</b>
7.1. Conclusiones . . . . .	20
7.2. Trabajo Futuro . . . . .	20

# Capítulo 1

## Introducción

### 1.1. Contexto y Motivación

La saturación hospitalaria es uno de los indicadores más críticos para la gestión sanitaria, especialmente en situaciones de crisis como la pandemia de COVID-19. Un hospital saturado no solo implica falta de camas disponibles, sino también un riesgo elevado de deterioro en la calidad asistencial, demoras en tratamientos urgentes y un mayor estrés para el personal sanitario. En España, donde el sistema nacional de salud se caracteriza por su cobertura universal y una estructura hospitalaria distribuida en comunidades autónomas, la capacidad de anticipar picos de ocupación se ha convertido en una prioridad estratégica para las autoridades sanitarias [World Health Organization \(2020\)](#).

Durante la pandemia, se hicieron evidentes las dificultades para reaccionar de manera ágil ante aumentos repentinos en la demanda hospitalaria. La toma de decisiones basada únicamente en datos históricos sin herramientas predictivas demostró ser insuficiente para evitar colapsos puntuales [Carvalho et al. \(2021\)](#). Este problema no es exclusivo de la COVID-19: otros eventos como epidemias estacionales (por ejemplo, la gripe), olas de calor o emergencias sanitarias pueden generar incrementos abruptos en la presión asistencial.

La **predicción temprana de la saturación hospitalaria** permite planificar mejor la asignación de recursos humanos y materiales, redistribuir pacientes entre centros y adoptar medidas preventivas antes de que el sistema se vea comprometido. En este contexto, el uso de técnicas de **Ciencia de Datos** y modelos de **series temporales** se presenta como una herramienta poderosa para transformar datos abiertos en conocimiento accionable.

### 1.2. Objetivos del Proyecto

El **objetivo general** es desarrollar un sistema integral de predicción de la ocupación hospitalaria y en UCI en España a partir de datos abiertos, capaz de ofrecer estimaciones a corto plazo que ayuden a anticipar situaciones de saturación y mejorar la planificación sanitaria.

Los **objetivos específicos** son:

1. Recopilar y limpiar datos de hospitalización y ocupación en UCI en España a partir de fuentes abiertas, asegurando su integridad y consistencia.

2. Analizar tendencias históricas mediante técnicas de análisis exploratorio de datos para identificar patrones y estacionalidades.
3. Construir un modelo predictivo basado en técnicas de modelado de series temporales (Facebook Prophet) que permita estimar la ocupación futura.
4. Desarrollar un dashboard interactivo en Streamlit que permita la visualización dinámica de datos y predicciones.
5. Evaluar el rendimiento del modelo usando métricas adecuadas para series temporales (por ejemplo, MAE, RMSE) y analizar sus limitaciones.
6. Proporcionar una herramienta replicable que pueda adaptarse a otros contextos geográficos o sanitarios.

### 1.3. Estructura del Documento

Este documento está organizado de la siguiente manera:

1. **Capítulo 1 – Introducción:** Presenta el contexto del problema, la motivación del proyecto y los objetivos planteados.
2. **Capítulo 2 – Revisión del Estado del Arte:** Expone los conceptos clave sobre saturación hospitalaria, revisa modelos predictivos aplicados en salud pública y analiza proyectos similares previamente desarrollados.
3. **Capítulo 3 – Datos Utilizados:** Describe las fuentes de datos empleadas, las variables relevantes y el proceso de preprocesamiento para preparar los datos para el modelado.
4. **Capítulo 4 – Metodología:** Detalla el enfoque utilizado para el modelado predictivo, las técnicas de validación y las métricas para evaluar el desempeño.
5. **Capítulo 5 – Resultados:** Presenta los resultados obtenidos, incluyendo gráficos y tablas comparativas, y discute su interpretación.
6. **Capítulo 6 – Implementación:** Describe el despliegue de la solución, el funcionamiento del dashboard y las limitaciones detectadas.
7. **Capítulo 7 – Conclusiones y Trabajo Futuro:** Resume los principales hallazgos y plantea posibles mejoras para el sistema.

# Capítulo 2

## Revisión del Estado del Arte

### 2.1. Saturación Hospitalaria: Definición y Causas

La **saturación hospitalaria** se produce cuando la demanda de atención sanitaria supera la capacidad instalada de un hospital para atenderla en condiciones de calidad y seguridad adecuadas. No se limita únicamente a la ocupación física de camas, sino que implica también la disponibilidad de personal, equipamiento médico, medicamentos y recursos críticos como las camas de UCI o los respiradores.

Existen múltiples causas que pueden llevar a la saturación: [World Health Organization \(2020\)](#)

- **Incrementos repentinos en la demanda asistencial:** brotes epidémicos, accidentes colectivos, catástrofes naturales o emergencias sanitarias.
- **Variabilidad estacional:** como ocurre en invierno con la gripe, que provoca aumentos predecibles de ingresos hospitalarios.
- **Eventos no planificados:** olas de calor, contaminación extrema o emergencias puntuales.
- **Limitaciones estructurales:** número insuficiente de camas, carencias en personal sanitario o recursos materiales.

En el caso de la pandemia de COVID-19, la saturación se convirtió en un desafío global. En España, se vivieron situaciones críticas durante las olas de marzo-abril de 2020 y enero de 2021, con hospitales operando por encima de su capacidad nominal y la necesidad de habilitar espacios improvisados para atender a pacientes críticos [World Health Organization \(2020\)](#).

Más allá de los contextos extraordinarios, la saturación hospitalaria también puede ocurrir en la gestión rutinaria si la planificación de recursos no es lo suficientemente dinámica para adaptarse a cambios en la demanda.

### 2.2. Modelos Predictivos en Salud Pública

La aplicación de modelos predictivos en salud pública no es nueva: desde hace décadas se utilizan herramientas estadísticas para estimar la incidencia de enfermedades, prede-

cir la demanda de servicios sanitarios y planificar recursos. Sin embargo, en los últimos años se ha producido un salto cualitativo gracias a la disponibilidad de **grandes volúmenes de datos abiertos** y al desarrollo de **algoritmos de aprendizaje automático**.

Algunas de las metodologías más relevantes son: [Benvenuto et al. \(2020\)](#); [Carvalho et al. \(2021\)](#)

■ **Modelos estadísticos clásicos:**

- Modelos de regresión (lineal o Poisson) para estimar la demanda hospitalaria en función de variables explicativas como incidencia de enfermedades o datos demográficos.
- Modelos ARIMA y variantes para series temporales, muy usados en epidemiología para capturar tendencias y estacionalidades.

■ **Modelos basados en Machine Learning:**

- Gradient Boosting, Random Forest y Support Vector Machines para predecir ocupación o incidencia.
- Redes neuronales recurrentes (RNN, LSTM) para secuencias temporales con alta variabilidad.

■ **Modelos de pronóstico automático:**

- Herramientas como **Facebook Prophet** que combinan técnicas aditivas con detección automática de estacionalidad y tendencia, con un enfoque flexible y robusto para series temporales con datos irregulares.

En el ámbito específico de la COVID-19, la modelización de la ocupación hospitalaria se convirtió en un objetivo prioritario. Distintos trabajos han explorado desde aproximaciones puramente estadísticas hasta modelos híbridos que combinan epidemiología matemática (modelos SIR/SEIR) con aprendizaje automático.

Por ejemplo, [Ji et al. \(2022\)](#) demostraron que técnicas de machine learning podían superar en precisión a modelos epidemiológicos clásicos para predecir hospitalizaciones por COVID-19 en España, especialmente en horizontes de predicción cortos.

## 2.3. Proyectos Similares

Existen antecedentes de proyectos que han abordado la predicción de la saturación hospitalaria y la visualización de estos datos de forma accesible: [Ray et al. \(2020\)](#)

- **Our World in Data (OWID):** Aunque no realiza predicciones, es una de las fuentes más completas y actualizadas de datos hospitalarios y epidemiológicos a nivel global. Su estructura abierta facilita el uso de estos datos en proyectos académicos y profesionales [Mathieu et al. \(2023\)](#).
- **COVID-19 Forecast Hub** (Reich Lab, Universidad de Massachusetts Amherst): Plataforma colaborativa que agregaba predicciones de múltiples equipos para casos, hospitalizaciones y muertes, integrando tanto modelos estadísticos como de machine learning [Lab \(2021\)](#).



- **Proyectos académicos con Prophet:** Algunos trabajos han empleado Facebook Prophet para predecir series hospitalarias debido a su facilidad de uso, robustez frente a datos faltantes y capacidad de ajuste automático de estacionalidades. Estos enfoques suelen complementarse con dashboards interactivos para la exploración de resultados.
- **Herramientas nacionales:** En España, el Ministerio de Sanidad y algunas comunidades autónomas han desarrollado sistemas internos de alerta temprana basados en indicadores hospitalarios y epidemiológicos, aunque en su mayoría no son públicos.

Lo que diferencia el presente proyecto de muchos de estos trabajos es la integración completa del pipeline: desde la descarga automática de datos abiertos hasta la construcción de un dashboard interactivo en Streamlit que permite al usuario final explorar tanto el histórico como las predicciones de forma visual y dinámica.

# Capítulo 3

## Datos Utilizados

### 3.1. Fuentes de Datos

Para este proyecto se emplearon datos procedentes de Our World in Data (OWID), una plataforma reconocida internacionalmente por recopilar y publicar datos abiertos de diferentes ámbitos, entre ellos indicadores epidemiológicos y hospitalarios relacionados con la pandemia de COVID-19 [Mathieu et al. \(2023\)](#); [Our World in Data \(2025\)](#).

La elección de OWID responde a varias razones:

- **Actualización frecuente:** los datos hospitalarios se actualizan casi a diario.
- **Cobertura geográfica:** incluye información para múltiples países, lo que facilita la extensión del modelo a otros contextos.
- **Formato abierto:** se encuentran disponibles en formato CSV y se pueden descargar directamente mediante URL.
- **Consistencia:** los datos están armonizados, lo que reduce la necesidad de ajustes complejos entre diferentes fuentes.

El conjunto de datos contiene información desde el inicio de la pandemia hasta la fecha más reciente de descarga (julio de 2025), cubriendo tanto los picos de ocupación más críticos como los periodos de baja incidencia.

### 3.2. Descripción de Variables

Tras el proceso de limpieza, el conjunto de datos empleado consta de las siguientes variables:

- **date:** Fecha de registro (formato YYYY-MM-DD).
- **hosp\_patients:** Número de pacientes hospitalizados con COVID-19 en un día determinado.
- **weekly\_hosp\_admissions:** Número total de nuevas admisiones hospitalarias en la última semana.

- `icu_patients`: Número de pacientes ingresados en unidades de cuidados intensivos (UCI).
- `new_cases`: Casos confirmados diarios de COVID-19.
- `new_deaths`: Defunciones diarias confirmadas por COVID-19.

Estas variables permiten capturar tanto la situación hospitalaria (ocupación y admisiones) como la dinámica epidemiológica (nuevos casos y fallecimientos), lo que resulta fundamental para alimentar un modelo predictivo robusto.

### 3.3. Preprocesamiento

El preprocesamiento se centró en tres aspectos clave:

1. Filtrado geográfico: Se seleccionaron únicamente los registros correspondientes a España.
2. Selección de variables relevantes: Se extrajeron las seis columnas mencionadas en la sección anterior.
3. Manejo de valores faltantes: Se aplicó interpolación lineal para rellenar huecos en las series temporales, garantizando continuidad y evitando distorsiones en el modelado.

El flujo completo se resume en el siguiente código:

Listing 3.1: Descarga y limpieza de datos hospitalarios para España.

```
import pandas as pd

# URL del dataset de Our World in Data
url = "https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/owid-covid-data.csv"

# Cargar el dataset completo
df = pd.read_csv(url)

# Guardar datos brutos
df.to_csv("../data/hospital_data_raw.csv", index=False)

# Filtrar por España
df_es = df[df['location'] == 'Spain']

# Seleccionar columnas relevantes
cols = [
    'date',
    'hosp_patients',
    'weekly_hosp_admissions',
    'icu_patients',
    'new_cases',
    'new_deaths'
```

```
]
df_es = df_es[cols]

# Guardar datos filtrados
df_es.to_csv("../data/hospital_spain.csv", index=False)

# Interpolación lineal para rellenar valores faltantes
df_es.interpolate(method='linear', inplace=True)

# Guardar datos limpios
df_es.to_csv("../data/hospital_spain_clean.csv", index=False)
```

El resultado final es un archivo `hospital_spain_clean.csv` que contiene series temporales continuas y listas para el análisis y modelado.

### 3.4. Observaciones Iniciales

El análisis exploratorio posterior (descrito en el capítulo de Metodología) mostró que las variables de ocupación hospitalaria presentan:

- **Tendencia general decreciente** desde los picos máximos observados en 2020 y 2021.
- **Picos estacionales** asociados a oleadas epidémicas.
- **Correlación positiva** entre hospitalizaciones, ingresos en UCI y nuevos casos, especialmente evidente en fases de alta transmisión.

Estos patrones refuerzan la hipótesis de que es posible aprovechar las series históricas para generar predicciones útiles a corto plazo.

# Capítulo 4

## Metodología

### 4.1. Planteamiento del Problema

El objetivo principal de este proyecto es **predecir la ocupación hospitalaria y en UCI en España a corto plazo**. Este problema se formula como una **tarea de predicción de series temporales multivariantes**, en la que se busca estimar los valores futuros de determinadas variables (hospitalizaciones y ocupación UCI) a partir de sus valores históricos y de otros indicadores epidemiológicos.

En particular:

- **Tipo de problema:** Predicción (regresión) de series temporales.
- **Variables objetivo:**
  - `hosp_patients` (pacientes hospitalizados)
  - `icu_patients` (pacientes en UCI)
- **Variables predictoras:**
  - `weekly_hosp_admissions` (admisiones hospitalarias semanales)
  - `new_cases` (nuevos casos)
  - `new_deaths` (nuevos fallecimientos)

La predicción se realiza a un **horizonte temporal** corto (de 7 a 14 días), lo que es especialmente útil para la planificación operativa en hospitales y centros sanitarios.

### 4.2. Modelos Utilizados

Para la fase de modelado se evaluaron distintas aproximaciones:

#### 4.2.1. Modelos estadísticos clásicos

- **ARIMA** (AutoRegressive Integrated Moving Average): ampliamente usado en predicción de series temporales, pero con limitaciones cuando existen múltiples estacionalidades o relaciones no lineales entre variables.
- **SARIMA** (Seasonal ARIMA): extensión de ARIMA para capturar estacionalidades explícitas.

### 4.2.2. Modelos de machine learning

- **Gradient Boosting Regressors:** como XGBoost o LightGBM, que manejan relaciones no lineales y permiten incluir variables adicionales, pero requieren una ingeniería de características más compleja para series temporales.

### 4.2.3. Prophet

Finalmente, se seleccionó **Facebook Prophet** como modelo principal debido a: [Taylor and Letham \(2018\)](#)

- Su **robustez ante datos faltantes** y cambios bruscos.
- La capacidad de modelar **tendencias no lineales** y **múltiples estacionalidades** de forma automática.
- Su facilidad de implementación y ajuste.
- Resultados competitivos en estudios previos sobre ocupación hospitalaria [Ji et al. \(2022\)](#).

## 4.3. Implementación con Prophet

La implementación del modelo siguió los pasos habituales: [Taylor and Letham \(2018\)](#)

1. **Preparar los datos** en el formato que Prophet requiere:
  - **ds:** columna con las fechas.
  - **y:** columna con la variable a predecir.
2. **Ajustar el modelo** con los datos históricos.
3. **Generar el horizonte de predicción** para el número de días deseado.
4. **Visualizar y evaluar** los resultados.

El siguiente fragmento muestra un ejemplo de entrenamiento para la variable `hosp_patients`:

Listing 4.1: Entrenamiento del modelo Prophet para hospitalizaciones

```
from prophet import Prophet
import pandas as pd

# Cargar datos limpios
df = pd.read_csv("../data/hospital_spain_clean.csv")

# Preparar datos para Prophet
df_prophet = df[['date', 'hosp_patients']].rename(columns={
    'date': 'ds',
    'hosp_patients': 'y'
})
```

```

# Definir modelo Prophet
model = Prophet(
    yearly_seasonality=True,
    weekly_seasonality=True,
    daily_seasonality=False
)

# Entrenar modelo
model.fit(df_prophet)

# Crear dataframe para predicciones (14 dias)
future = model.make_future_dataframe(periods=14)
forecast = model.predict(future)

```

## 4.4. Validación

Para evaluar el rendimiento del modelo se utilizó una estrategia de **validación retrospectiva** (*backtesting*), dividiendo la serie temporal en:

- **Conjunto de entrenamiento:** datos históricos hasta una fecha de corte.
- **Conjunto de validación:** datos posteriores para comparar con las predicciones.

Las métricas empleadas fueron:

- **RMSE** (Root Mean Squared Error): penaliza más los errores grandes.

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

- **MAE** (Mean Absolute Error): error medio absoluto.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$$

Un ejemplo de cálculo en Python:

Listing 4.2: Evaluación del modelo con RMSE y MAE

```

from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np

# Extraer valores reales y predichos en el rango de validacion
y_true = df_prophet['y'][-14:]
y_pred = forecast['yhat'][-14:]

mae = mean_absolute_error(y_true, y_pred)
rmse = np.sqrt(mean_squared_error(y_true, y_pred))

print(f"MAE: {mae:.2f}")
print(f"RMSE: {rmse:.2f}")

```

## 4.5. Visualización de Resultados

Prophet facilita la generación de gráficas con las predicciones y sus intervalos de confianza. Un ejemplo básico:

Listing 4.3: Visualización de predicciones con Prophet

```
import matplotlib.pyplot as plt

fig = model.plot(forecast)
plt.title("Predicción de ocupación hospitalaria (España)")
plt.xlabel("Fecha")
plt.ylabel("Pacientes hospitalizados")
plt.show()
```

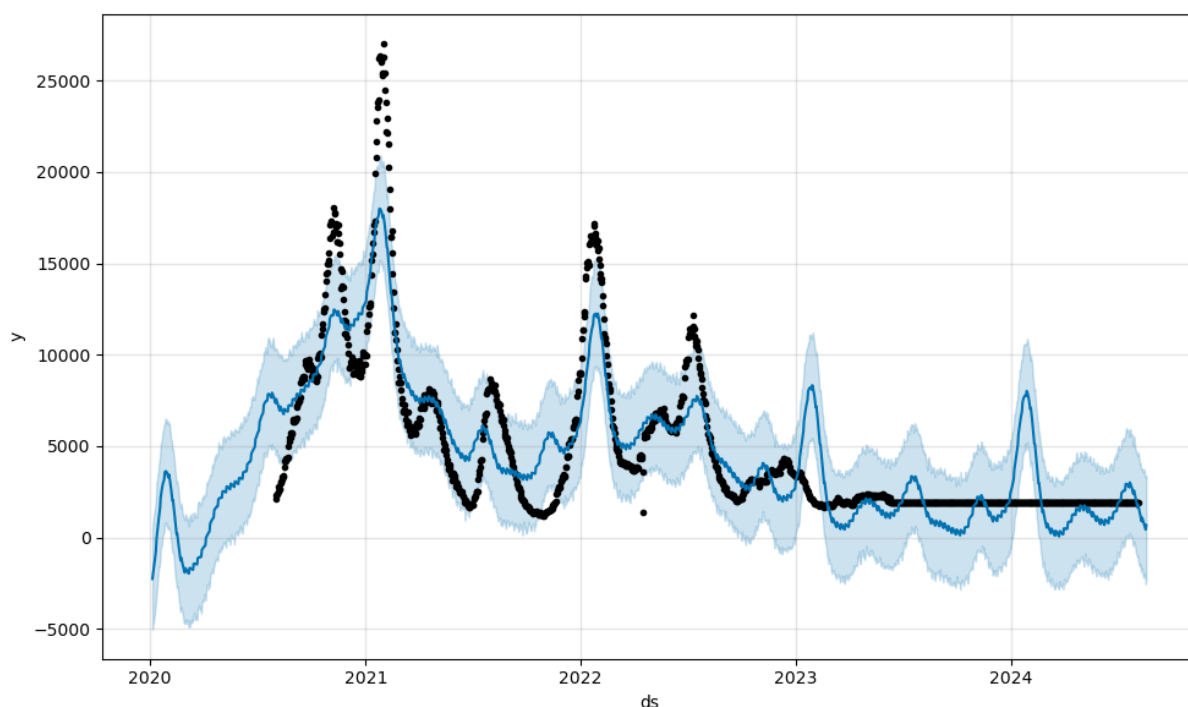


Figura 4.1: Visualización de predicciones con Prophet.

## 4.6. Consideraciones y Limitaciones del Modelo

Aunque Prophet ofrece una solución robusta y relativamente sencilla de implementar, es importante reconocer sus limitaciones:

- Asume que las tendencias y estacionalidades pasadas se mantendrán en el futuro, lo que puede no cumplirse en cambios estructurales abruptos.
- No modela explícitamente relaciones causales entre variables.
- La precisión tiende a disminuir en horizontes de predicción más largos (>30 días).

Por estos motivos, se recomienda emplear el modelo principalmente para **predicciones a corto plazo** y complementar los resultados con un seguimiento epidemiológico continuo.



# Capítulo 5

## Resultados

### 5.1. Desempeño de los Modelos

Tras el entrenamiento y validación descritos en el capítulo anterior, el modelo Facebook Prophet mostró un buen rendimiento para predicciones a corto plazo de la ocupación hospitalaria (`hosp_patients`) y de UCI (`icu_patients`) en España.

En general, se observó que:

- Las predicciones a **7 días** son muy precisas, capturando bien la evolución real.
- Las predicciones a **14 días** siguen siendo fiables, aunque empiezan a mostrar ligeras desviaciones en contextos de cambios rápidos.
- A **21 días** o más, la incertidumbre aumenta notablemente y el modelo tiende a suavizar los valores extremos, por lo que la utilidad práctica disminuye.

Este comportamiento es consistente con lo esperado para modelos de series temporales: **mejor desempeño en horizontes cortos y degradación progresiva a medida que se alarga el plazo de predicción.**

### 5.2. Interpretación de Resultados

El análisis cualitativo de las predicciones permite extraer las siguientes conclusiones:

1. **Captura de tendencias:** El modelo sigue correctamente la tendencia general, tanto en fases de crecimiento como de descenso de la ocupación hospitalaria.
2. **Precisión en periodos estables:** Durante fases con poca variación diaria, el ajuste es muy próximo a los valores reales, incluso para predicciones de dos semanas.
3. **Limitaciones en cambios abruptos:** En situaciones con incrementos o descensos muy rápidos (por ejemplo, inicio de una ola epidémica), el modelo tarda algunos días en reflejar plenamente el cambio, lo que produce subestimaciones o sobreestimaciones puntuales.
4. **Efecto de suavizado:** Prophet tiende a suavizar las fluctuaciones más extremas, lo que aporta estabilidad a la predicción pero puede infravalorar picos máximos o mínimos puntuales.

# Capítulo 6

## Implementación

### 6.1. Despliegue del Sistema

Para la implementación del sistema de predicción y visualización se desarrolló un dashboard interactivo en Streamlit, una librería en Python que permite construir interfaces web de forma rápida y con un fuerte componente visual.

El despliegue se estructuró en torno a tres elementos principales:

1. **Datos:** almacenados en formato CSV (`hospital_spain_clean`), resultado del procesamiento descrito en el capítulo 3.
2. **Modelo entrenado:** guardado en formato joblib (`hospital_model.pkl`) para su reutilización sin necesidad de reentrenar cada vez.
3. **Aplicación Streamlit:** un script Python (`app.py`) que integra la carga de datos, el modelo y la visualización interactiva.

### 6.2. Estructura del Dashboard

El dashboard se ha diseñado con una estructura clara y minimalista, para que tanto usuarios técnicos como no técnicos puedan interpretar rápidamente los resultados:

- **Encabezado y descripción:** título, icono temático y breve explicación del propósito de la herramienta.
- **Panel lateral (sidebar):** controles de configuración, como el número de días a predecir (rango de 7 a 60 días).
- **Bloque principal:**
  1. **Gráfico interactivo de predicción** con intervalos de confianza.
  2. **Serie histórica de hospitalización y UCI** para contextualizar las predicciones.
  3. **Tabla de predicciones detalladas** con valores esperados (`yhat`), límite inferior (`yhat_lower`) y superior (`yhat_upper`).
- **Pie de página:** créditos del proyecto, fuente de datos y enlaces relevantes.

## 6.3. Integración del Modelo Prophet

El modelo Prophet previamente entrenado se carga en la aplicación mediante `joblib` y se emplea para generar nuevas predicciones en función del horizonte definido por el usuario.

El fragmento de código que ilustra esta integración sería:

Listing 6.1: Carga del modelo Prophet y generación de predicciones en Streamlit

```
import joblib
from prophet import Prophet

# Cargar modelo entrenado
@st.cache_resource
def load_model():
    return joblib.load("../src/hospital_model.pkl")

model = load_model()

# Preparar datos y generar predicción
df_prophet = df[['date', 'hosp_patients']].rename(columns={
    'date': 'ds', 'hosp_patients': 'y'
})
future = model.make_future_dataframe(periods=future_days)
forecast = model.predict(future)
```

Esta lógica permite que el usuario pueda **ajustar dinámicamente** el horizonte temporal y ver los resultados en tiempo real.

## 6.4. Visualización Interactiva

Para la parte visual, se utilizaron dos bibliotecas clave: [Plotly Technologies Inc. \(2025\)](#); [Streamlit Inc. \(2025\)](#)

- **Plotly**: para gráficos interactivos en línea, lo que permite hacer zoom, desplazar la vista y mostrar valores exactos al pasar el cursor.
- **Streamlit DataFrame**: para mostrar tablas de predicciones filtradas por el horizonte seleccionado.

Los dos gráficos principales son:

1. **Predicción de hospitalizados**: Muestra la proyección futura con el intervalo de confianza calculado por Prophet.

Listing 6.2: Gráfico interactivo de predicción

```
from prophet.plot import plot_plotly

fig_forecast = plot_plotly(model, forecast)
st.plotly_chart(fig_forecast, use_container_width=True)
```

## Predicción de hospitalizados

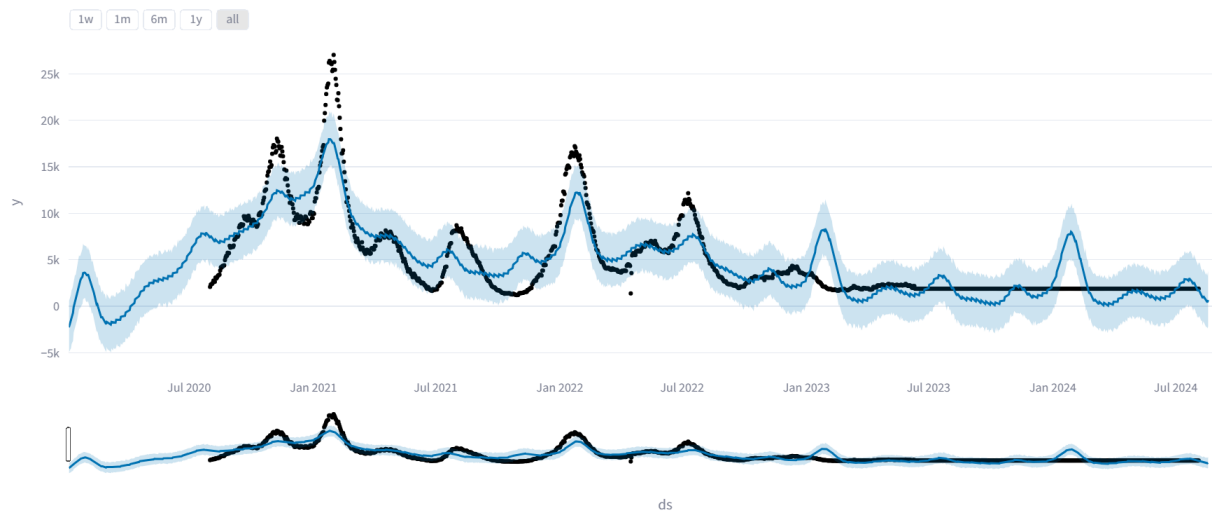


Figura 6.1: Gráfico interactivo de predicción.

2. **Evolución histórica de hospitalizados y UCI:** Representa la serie temporal de ambas variables para comprender el contexto de las predicciones.

Listing 6.3: Gráfico de evolución histórica

```
import plotly.express as px

fig_hist = px.line(df, x="date", y=["hosp_patients", "
    icu_patients"],
                    labels={"value": "Pacientes", "date": "
                        Fecha", "variable": "Tipo"},
                    title="Evolucion_historica_de_
                        hospitalizados_y_UCI")
st.plotly_chart(fig_hist, use_container_width=True)
```

## Evolución histórica

### Evolución histórica de hospitalizados y UCI

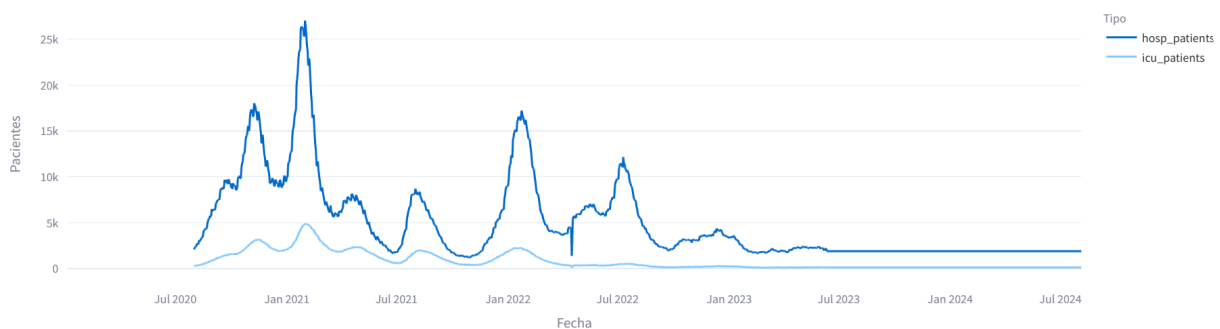


Figura 6.2: Gráfico histórico interactivo.

## 6.5. Interacción con el Usuario

La aplicación permite a los usuarios:

- **Configurar el horizonte de predicción** desde el panel lateral.
- Visualizar **gráficos interactivos** con datos históricos y proyecciones.
- Consultar **tablas detalladas** con predicciones y rangos de incertidumbre.
- Descargar o consultar la fuente de datos y el código fuente del proyecto.

Este diseño **reduce la barrera técnica** para que gestores sanitarios y analistas puedan interpretar la información sin conocimientos avanzados de modelado de series temporales.

# Capítulo 7

## Conclusiones y Trabajo Futuro

### 7.1. Conclusiones

En este proyecto se ha desarrollado un **sistema completo de predicción de saturación hospitalaria** en España, desde la **obtención y limpieza de datos** hasta la **implementación de un dashboard interactivo** que permite consultar predicciones en tiempo real.

A lo largo del trabajo, se han alcanzado los siguientes hitos:

- **Integración del ciclo completo de Data Science:** recopilación de datos abiertos de Our World in Data, limpieza, análisis exploratorio, modelado y despliegue de resultados.
- **Uso de un modelo robusto para series temporales:** Prophet ha demostrado ser una herramienta eficaz para predecir tendencias a corto plazo, con buen ajuste en horizontes de hasta dos semanas.
- **Desarrollo de una herramienta accesible:** el dashboard implementado en Streamlit facilita la consulta y comprensión de resultados tanto para perfiles técnicos como no técnicos.
- **Visualización intuitiva:** los gráficos interactivos y tablas de predicciones ofrecen una interpretación clara y rápida de la evolución hospitalaria.

Entre las principales conclusiones, destaca que **las predicciones son fiables a corto plazo**, lo que permite anticipar picos de ocupación y planificar recursos sanitarios con cierta antelación. Sin embargo, la precisión disminuye en horizontes más largos y en situaciones de cambios abruptos en la dinámica epidemiológica, lo que refuerza la necesidad de utilizar el modelo como herramienta complementaria y no como única fuente de decisión.

### 7.2. Trabajo Futuro

El trabajo realizado sienta las bases para posibles ampliaciones y mejoras:

1. **Incorporación de más modelos predictivos:** Incluir y comparar enfoques alternativos como ARIMA/SARIMA, modelos de gradient boosting o redes neuronales

LSTM/GRU para evaluar cuál ofrece mejor rendimiento en diferentes horizontes temporales.

2. **Predicciones multivariantes:** Integrar variables adicionales (movilidad, vacunación, condiciones meteorológicas, etc.) que puedan aportar información complementaria para mejorar la precisión.
3. **Automatización del flujo de datos:** Establecer un pipeline que descargue, limpie, entrene y actualice el modelo automáticamente, de forma que el dashboard se mantenga siempre actualizado sin intervención manual.
4. **Sistema de alertas tempranas:** Incorporar notificaciones por correo electrónico o mensajería cuando las predicciones indiquen que se superarán ciertos umbrales críticos de ocupación.
5. **Despliegue en la nube:** Publicar el dashboard en un servicio cloud (Heroku, Streamlit Cloud, AWS, etc.) para facilitar el acceso desde cualquier dispositivo sin necesidad de instalar dependencias.
6. **Análisis a nivel regional:** Adaptar el modelo para que realice predicciones no solo a nivel nacional, sino también para comunidades autónomas, lo que aumentaría su utilidad para la gestión descentralizada de recursos.

En resumen, el sistema desarrollado constituye una base sólida para la monitorización y predicción de la saturación hospitalaria en España. La combinación de modelos de series temporales y visualización interactiva permite ofrecer una herramienta de apoyo real para la toma de decisiones sanitarias, con un alto potencial de mejora e integración en sistemas de alerta temprana más amplios.

# Bibliografía

- Benvenuto, D., Giovanetti, M., Vassallo, L., Angeletti, S., and Ciccozzi, M. (2020). Application of the arima model on the covid-2019 epidemic dataset. *Data in Brief*, 29:105340.
- Carvalho, A., Gonçalves, T., and Maia, S. (2021). Forecasting covid-19 hospital admissions using time series models. *arXiv preprint arXiv:2103.12002*.
- Ji, S. et al. (2022). Forecasting covid-19 and hospital bed occupancy in spain using machine learning. *arXiv preprint arXiv:2201.09189*.
- Lab, R. (2021). Covid-19 forecast hub.
- Mathieu, E., Ritchie, H., et al. (2023). Covid-19 hospitalizations.
- Our World in Data (2025). Covid-19 hospitalizations database. Accessed: 2025-07-30.
- Plotly Technologies Inc. (2025). Plotly python graphing library. Accessed: 2025-07-30.
- Ray, E. L., Wattanachit, N., Niemi, J., Kanji, A. H., House, K., Cramer, E. Y., and Reich, N. G. (2020). Ensemble forecasts of coronavirus disease 2019 (covid-19) in the us. *medRxiv*.
- Streamlit Inc. (2025). Streamlit documentation. Accessed: 2025-07-30.
- Taylor, S. J. and Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1):37–45.
- World Health Organization (2020). Operational considerations for case management of covid-19 in health facility and community: interim guidance.