
This is the **published version** of the bachelor thesis:

Baron Ciriano, Biel. *Rate shopper i predicció de demanda : Plataforma de Revenue Intelligence per Glamping*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/326533>

under the terms of the  license.

Rate shopper i predicció de demanda: Plataforma de Revenue Intelligence per Glamping

Biel Barón Ciriano

28 de gener de 2026

Resum– Aquest treball presenta l'evolució de RateGlamp, un *rate shopper* especialitzat en càmpings, cap a una eina avançada de *Revenue Intelligence*. A partir d'una plataforma ja operativa capaç de recopilar preus i ressenyes de la competència, el projecte incorpora models predictius d'ocupabilitat basats en tècniques d'intel·ligència artificial. S'han desenvolupat i avaluat diferents models de *forecasting*, seleccionant el més adequat segons el seu rendiment amb dades reals. Les prediccions s'integren en un nou *dashboard* web orientat a usuaris Premium, que permet visualitzar ocupació prevista, preus i recomanacions. Tot el sistema s'ha desplegat sobre una arquitectura escalable al núvol utilitzant AWS (Amazon Web Services) i Firebase. L'objectiu final és proporcionar suport a la presa de decisions i maximitzar els ingressos dels càmpings mitjançant informació predictiva adaptada a les seves particularitats.

Paraules clau– Rate shopper, Revenue intelligence, pronòstic, Intel·ligència Artificial, LSTM (Long Short-Term Memory), Vue, Núvol, AWS, Camping

Abstract– This project presents the evolution of RateGlamp, a rate shopper specifically designed for campsites, into an advanced Revenue Intelligence tool. Building on an existing platform capable of collecting competitor prices and reviews, the system incorporates occupancy forecasting models based on artificial intelligence techniques. Several forecasting approaches were developed and evaluated using real data, selecting the most suitable model according to performance metrics. The resulting predictions are integrated into a new web-based dashboard available to Premium users, enabling the visualization of expected occupancy, prices, and recommendations. The entire solution is deployed on a scalable cloud architecture using AWS and Firebase. The final objective is to support data-driven decision making and maximize campsite revenues through predictive insights tailored to the specific characteristics of the camping sector.

Keywords– Rate shopper, Revenue intelligence, forecast, Artificial Intelligence, LSTM, Vue, Cloud, Amazon Web Services (AWS), Camping



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

EN els darrers anys, el sector del càmping ha experimentat una transformació significativa, marcada per una creixent digitalització i per la necessitat de dis-

posar d'eines tecnològiques que permetin prendre decisions més informades. En aquest context, ha sorgit l'oportunitat de desenvolupar un *rate shopper* especialitzat, un tipus de programari que recopila, analitza i compara preus i disponibilitat d'allotjaments amb l'objectiu d'optimitzar la presa de decisions i entendre la posició competitiva al mercat [1]. Aquest projecte s'emmarca en aquesta iniciativa i es construeix a partir de la feina desenvolupada durant les pràctiques curriculars a l'empresa Talenbox.

Les solucions existents estan dissenyades per a hotels i no tenen en compte particularitats com la gran varietat de tipologies d'allotjament al sector càmping, l'alta estacionalitat,

- E-mail de contacte: biel.bar.cir@gmail.com
- Menció: Tecnologies de la Informació
- Treball tutoritzat per: Juan Antonio Martinez Carrascal (Àrea de Ciències de la Computació i Intel·ligència Artificial)
- Curs 2025/26

la forta dependència del clima o els perfils de client diferenciats. Davant d'aquesta situació, s'identifica una clara oportunitat de crear una eina especialitzada que cobreixi realment les necessitats d'aquest segment del sector turístic.

La primera fase desenvolupada durant les pràctiques curriculars a l'empresa Talenbox, s'ha implementat un sistema capaç d'obtenir preus d'un canal de venda tan rellevant com Booking mitjançant *scrapers* desenvolupats a mida. Les dades obtingudes s'emmagatzemen en una base de dades al núvol i es mostren a l'usuari a través d'un *dashboard web* interactiu que facilita la comparació directa amb la competència. Paral·lelament, s'ha desenvolupat un mòdul de recollida i visualització de ressenyes, que permet detectar tendències qualitatives i identificar fortaleeses i debilitats dels competidors des del punt de vista del client.

Tot el programari s'ha construït utilitzant una arquitectura al núvol, emprant AWS per al servidor i Firebase per a la pàgina web, garantint escalabilitat, fiabilitat i facilitat de manteniment.

El present treball parteix d'aquesta base ja desenvolupada i planteja la seva evolució cap a una eina completa de *Revenue Intelligence* [2]. L'objectiu final és incorporar-hi models d'intel·ligència artificial capaços de predir la demanda, anticipar l'evolució dels preus i oferir recomanacions que ajudin a maximitzar els ingressos. Aquesta transició transformarà el *rate shopper* inicial en una plataforma d'anàlisi avançada, alineada amb les tendències més recents del sector i adaptada específicament a la realitat dels càmpings.

2 OBJECTIUS

L'objectiu principal del treball és evolucionar el *rate shopper* cap a una solució de *Revenue Intelligence* que incorpori prediccions d'ocupabilitat empaquetades en un nou pla *Premium*.

Aquest objectiu general es concreta en tres línies principals d'actuació, primer el desenvolupament d'un model predictiu de demanda, ampliar la web amb un *dashboard* de prediccions i finalment integrar les prediccions en la presa de decisions. A continuació es descriuen els objectius clau desglossats en requisits funcionals (RF), obligatoris (O) o desitjables (D). A l'apèndix A.1 es pot trobar una taula resum amb tots els requisits funcionals a més dels requisits no funcionals.

2.1 Desenvolupament d'un model predictiu d'ocupabilitat

- **[RF1 – O]** Desenvolupar un prototip de model predictiu d'ocupabilitat a partir de les dades del càmping de referència.
- **[RF2 – D]** Avaluar diferents models predictius: ARIMA (AutoRegressive Integrated Moving Average), Prophet i LSTM (Long Short-Term Memory).
- **[RF3 – O]** Mesurar el rendiment mitjançant MAE (Error Absolut Mitjà) [3], RMSE (Arrel de l'Error Quadràtic Mitjà) [4] i regressió lineal [5] amb dades de *train* i *test*.
- **[RF4 – O]** Seleccionar el model que millor s'ajusti a les dades segons les mètriques obtingudes.

- **[RF5 – D]** Explorar la incorporació de variables externes (temps atmosfèric, preu mitjà de l'allotjament, ...).

2.2 Ampliació de la web amb un *dashboard* de prediccions

- **[RF6 – O]** Implementar filtres al *dashboard*: tipus d'allotjament, horitzó temporal i data d'inici.
- **[RF7 – O]** Desenvolupar un calendari amb: ocupació real, ocupació predita, preu actual, preu suggerit i preu mitjà dels competidors.
- **[RF8 – O]** Implementar una gràfica comparativa de prediccions.
- **[RF9 – O]** Afegir al *dashboard* d'administració una opció per assignar el pla *Premium* als clients.

2.3 Adaptació del back end per donar suport al pla *Premium*

- **[RF10 – O]** Ampliar la funcionalitat del *dashboard* administratiu per permetre l'assignació del pla *Premium*.
- **[RF11 – O]** Crear els endpoints necessaris per al *dashboard* predictiu (filtres, calendari i gràfiques).
- Preparar una arquitectura escalable perquè les prediccions es generin setmanalment mitjançant AWS:
 - **[RF12 – O]** Fer servir contenidors ECS (Elastic Container Service), un per usuari, per generar les prediccions i guardar-les a la base de dades RDS (Relational Database Service).
 - **[RF13 – O]** Implementar una Lambda Function per crear els *batches* de treball i iniciar els contenidors ECS.

Aquests objectius permeten transformar l'actual aplicació en una eina de *Revenue Intelligence* capaç de proporcionar informació predictiva, suport a la presa de decisions i un nou servei *Premium* orientat a incrementar el valor aportat als càmpings.

3 ESTAT DE L'ART

Aquesta revisió permet contextualitzar el projecte dins del panorama tecnològic actual i identificar tant les limitacions del programari existent com les oportunitats d'innovació que han motivat la proposta presentada en aquest treball.

3.1 Solucions de *rate shopping* i *revenue management* al sector hotelier

El sector hotelier compta amb una gran maduresa digital pel que fa a eines de comparació de preus, presa de decisions i forecasting. Els *rate shoppers* tradicionals permeten obtenir i comparar preus de la competència, analitzar disponibilitat i estudiar tendències del mercat, integrant-se habitualment amb plataformes de reserva d'ús global.

Tot i això, aquestes eines estan orientades exclusivament al sector hotelier i no tenen en compte les característiques

específiques del món del càmping, on les tipologies d'allotjament, l'alta estacionalitat, la sensibilitat al clima i la diversitat de perfils de client generen models de demanda més complexos.

Actualment, destaquen diverses solucions consolidades en el mercat:

- **BEONx** [6]: plataforma especialitzada en forecasting anual amb més de 15 *inputs* i 80 variables, basada en tècniques d'intel·ligència artificial i *machine learning*. Ofereix prediccions de demanda molt completes, però està dissenyada per a entorns hotelers tradicionals.
- **Duetto** [7]: incorpora un mòdul de forecasting avançat que utilitza algoritmes d'IA per prediccions d'ocupació i rendiment. És una de les eines més potents del sector hotelier, però no contempla tipologies d'allotjament diferents ni forta dependència de condicions externes.
- **IDEAS RevPlan** [8]: se centra en la planificació estratègica i la unificació de dades de revenue. Aporta valor a grans cadenes hoteleres, però no ofereix flexibilitat per a mercats més heterogenis com el càmping.
- **Lighthouse** [9]: eina centrada en l'anàlisi de mercat i la predicció de demanda mitjançant dades massives, amb una clara orientació al sector hotelier urbà.
- **Actabl** [10]: software dedicat al pressupost i forecasting financer, integrat amb dades en temps real. Tot i la seva potència, l'abast és eminentment hotelier.
- **Amadeus Demand360** [11]: un dels sistemes més estesos en el sector. Aporta dades comparatives de demanda i ocupació, però no té en compte la variabilitat d'allotjaments ni la forta influència del clima en els càmpings.

3.2 Limitacions de les solucions existents per al sector càmping

Malgrat l'alta especialització d'aquestes eines, totes comparteixen un element comú: estan dissenyades per a hotels. Els càmpings presenten necessitats diferenciades que dificulten una aplicació directa d'aquests models:

- elevada estacionalitat i dependència meteorològica.
- heterogeneïtat d'allotjaments (tendes, bungalows, parcel·les, *glamping*).
- major sensibilitat del client a variacions de preu.
- diferents perfils de client segons temporada.

Aquestes diferències provoquen que els models de forecasting hotelers generin prediccions poc fiables quan s'apliquen al sector càmping. Això crea un espai d'oportunitat clar per desenvolupar eines específicament adaptades.

3.3 Aportació del projecte dins l'estat de l'art

En el context esmentat, el projecte desenvolupat resol una primera part del buit existent mitjançant la implementació d'un *dashboard* capaç de comparar preus amb competidors en canals de reserva especialitzats, així com d'un sistema d'extracció i visualització de ressenyes en aquests canals.

A partir d'aquesta base tecnològica ja operativa, el projecte planteja una segona fase centrada en la integració de models d'intel·ligència artificial per al *forecasting* de demanda i d'evolució de preus. L'objectiu és proporcionar una capa d'anàlisi predictiva que permeti anticipar comportaments del mercat i reforçar la presa de decisions estratègiques. Aquesta fase inclourà també la incorporació de noves funcionalitats al *dashboard*, per tal de visualitzar de manera clara i concisa les prediccions generades per aquests models i altres indicadors rellevants.

D'aquesta manera, el projecte no només cobreix mancances actuals en el sector, sinó que estableix les bases per a una eina integral de *revenue intelligence* adaptada específicament als càmpings, aportant innovació i diferenciació en un àmbit on encara no existeixen solucions tecnològiques equivalents.

4 METODOLOGIA

El desenvolupament del projecte s'ha estructurat seguint una metodologia seqüencial o *waterfall* [12], organitzada en dues fases principals: en primer lloc, el disseny, entrenament i validació dels models predictius; i, en segon lloc, la construcció del sistema necessari per a la seva explotació en un entorn real, incloent-hi el *back end*, l'API (*application programming interface*) i la integració amb el *front end*. Aquest enfocament és especialment adequat per a un projecte individual on les dependències són estrictament lineals: primer cal disposar d'un model funcional i estable, i només després és possible desplegar-lo i dissenyar una interfície que presenti la informació de manera clara i útil.

Tot i considerar metodologies més iteratives, es va optar per un esquema seqüencial perquè els requisits eren clars des de l'inici, el projecte no depenia d'agents externs i les entregues acadèmiques estaven estructurades en blocs tancats que encaixaven amb aquest paradigma. Malgrat això, s'ha mantingut certa flexibilitat per reajustar tasques i incorporar millores puntuals al llarg del desenvolupament.

A continuació es descriuen les dues fases principals de la metodologia.

4.1 Fase 1: Desenvolupament i validació dels models predictius

La primera fase s'ha centrat en la creació dels models de *forecasting*. A partir de dades reals es van preparar els conjunts d'entrenament i test, es van provar diferents arquitectures i es van aplicar mètriques d'avaluació com el MAE i el RMSE per identificar el model òptim. Aquesta fase va incloure processos iteratius d'ajust de paràmetres, validació creuada i anàlisi de resultats amb l'objectiu d'obtenir un model robust i fiable.

4.2 Fase 2: Desenvolupament del sistema complet: *back end*, API i *front end*

Un cop els models van quedar validats, es va iniciar la segona fase, orientada a la seva integració en un entorn productiu. En primer lloc, es va desenvolupar el *back end* i la infraestructura al núvol: els models es van contenidoritzar amb Docker per garantir la consistència entre entorns, es van desplegar els serveis a AWS mitjançant ECS, i es va dissenyar una API per exposar els resultats i facilitar la integració amb la resta del sistema.

Posteriorment es va implementar el *front end*. Prèviament s'havia definit el disseny preliminar del *dashboard* amb Figma per validar-ne la distribució i la coherència visual amb l'eina existent. Aquest treball previ va permetre una implementació més eficient dels components web i una integració progressiva amb la nova API.

5 PLANIFICACIÓ

Seguint la metodologia anteriorment descrita, la planificació inicial del projecte es va realitzar utilitzant Microsoft Project, eina que va permetre definir les tasques, les dependències i el calendari complet de desenvolupament. Aquest instrument també es va utilitzar per al seguiment continu del projecte, actualitzant l'estat de cada tasca i identificant possibles desviacions respecte a la planificació inicial. Un exemple d'aquest seguiment es pot observar en el diagrama de Gantt inclòs a l'Apèndix A.2.

Per garantir una comunicació fluida i una validació constant dels avenços, es van dur a terme dues reunions setmanals amb l'empresa Talenbox. En aquestes sessions s'anализava l'estat del projecte, es revisaven els objectius assolits i es definien les accions i prioritats per a les següents setmanes. Aquest mecanisme de coordinació va ser clau per mantenir l'alineació amb les necessitats reals de l'eina i assegurar la coherència dels resultats obtinguts.

Les fites principals del projecte han estat:

- **Setmana 10:** Models completament entrenats i validats.
- **Setmana 14:** Sistema complet integrat: *back end*, API i *front end*.

5.1 Incidències

La principal incidència del projecte es va produir durant la Fase 1, a causa de la complexitat associada a la preparació de les dades, l'entrenament dels models i els processos d'ajust i validació. Aquesta situació va provocar una desviació temporal respecte a la planificació inicial, de manera que la primera entrega de seguiment es va realitzar a la setmana 10 en lloc de la setmana 8 prevista.

Tot i aquest endarreriment parcial, l'entrega de l'informe de seguiment també es va ajustar al mateix període, fet que va evitar un impacte negatiu en el desenvolupament global del projecte i en les fites posteriors. Un cop superada aquesta fase, la Fase 2 es va desenvolupar amb més rapidesa de la prevista, permetent dedicar temps addicional a tasques de proves, optimització i redacció final de la documentació.

En conjunt, l'ús d'una metodologia seqüencial, combinada amb eines de planificació com Microsoft Project i reunions periòdiques de seguiment, va permetre mantenir el control del projecte, garantir la traçabilitat de les tasques i assolir els objectius dins dels terminis establerts. El Gantt de seguiment complet es pot consultar a l'Apèndix A.3.

6 ARQUITECTURA TECNOLÒGICA

Actualment, l'eina *Rateglamp* permet recopilar preus i ressenyes de més de 120 càmpings competidors a Catalunya mitjançant Booking. A continuació es detallen les tecnologies, arquitectures i eines emprades, així com la seva evolució durant el desenvolupament del projecte.

6.1 Infraestructura transversal

Durant el projecte s'han establert diversos components transversals que sustenten el projecte:

- **VPC (Virtual Private Cloud):** una infraestructura de xarxa aïllada per garantir seguretat i control del trànsit intern allotjada a AWS [13].
- **Base de dades Amazon RDS (PostgreSQL):** repositori centralitzat i escalable per a l'emmagatzematge de preus, ressenyes, prediccions i dades d'usuari [14].
- **Docker:** contenidors per garantir consistència entre entorns de desenvolupament i producció [15].

6.2 Antecedents del projecte

Abans de l'inici del projecte, l'eina constava de tres components principals: els *scrapers* per a l'extracció de dades, una API RESTful per al *back end* i un *front end* web per a la visualització de la informació. A continuació es detallen les tecnologies emprades en cadascun d'aquests components.

6.2.1 Scrapers

Els *scrapers* es van desenvolupar en Python emprant les següents llibreries principals: `requests`, `psycopg2-binary`, `playwright`, `boto3`.

Playwright s'utilitza per realitzar *scraping* a Booking mitjançant una navegació realista que imita el comportament d'un usuari, incloent-hi rotació de *user agents* per evitar bloquejos.

L'arquitectura d'execució a AWS és la següent: un *scheduler* d'Amazon EventBridge [16] inicia una Lambda Function [17] que genera lots de treball amb els URL (Uniform Resource Locator) provinents de la base de dades. Aquestes tasques s'encuen a SQS (Simple Queue Service) [18], i mitjançant una AWS Pipe [19] es desencadena l'execució d'un contenidor ECS per missatge. Cada contenidor processa el seu conjunt d'URL i desa les dades extretes a Amazon RDS.

Aquesta arquitectura desacobla la planificació (EventBridge), la generació de lots (Lambda), el buffer de missatges (SQS) i l'execució (ECS), aportant resiliència, reintents automàtics i escalabilitat horitzontal.

6.2.2 API

La API també està desenvolupada en Python, amb les següents llibreries: Flask, psycopg2-binary, flask-cors, firebase-admin, boto3, openpyxl, reportlab, matplotlib, gunicorn

S'ha construït com una API RESTful basada en Flask, que proporciona serveis per al *dashboard*, generació d'informes, gestió de ressenyes i administració d'usuaris. La comunicació amb Amazon RDS es realitza mitjançant psycopg2-binary.

La implementació segueix una arquitectura de microserveis interns, on funcionalitats com ressenyes, dades de preus o gestió d'usuaris es troben modularitzades. Aquesta organització permet interoperabilitat, mantenibilitat i la possibilitat de desplegar mòduls de forma independent en el futur.

Per al desplegament a AWS, s'ha optat per contenidors ECS gestionats mitjançant Fargate, que simplifica la infraestructura i permet escalar segons demanda. Gunicorn s'utilitza com a servidor d'aplicacions per gestionar múltiples connexions simultànies.

Per provar l'API s'ha utilitzat Postman, que facilita la creació i execució de peticions HTTP per validar els diferents endpoints desenvolupats.

6.2.3 Web

El *front end* està desenvolupat en Vue.js, amb les següents llibreries principals: chart.js, chartjs-plugin-zoom, firebase, leaflet, leaflet.heat, vue-chartjs, vue-router

Abans de començar el desenvolupament, es va dissenyar el *layout* i els components principals del *dashboard* amb Figma per validar la distribució i coherència visual amb l'eina existent. També es va fer una prova d'usabilitat amb usuaris reals per recollir feedback i ajustar detalls abans de la implementació.

Vue permet crear una interfície interactiva i responsiva a partir de components separats. La visualització de dades es realitza amb Chart.js per als gràfics i leaflet.heat permet generar mapes de calor basats en els preus del dia seleccionat dels càmpings competidors. Es pot veure a la Figura 1 una captura de pantalla del *dashboard* amb els tres components principals; calendari de preus, gràfica de preus i mapa de calor.

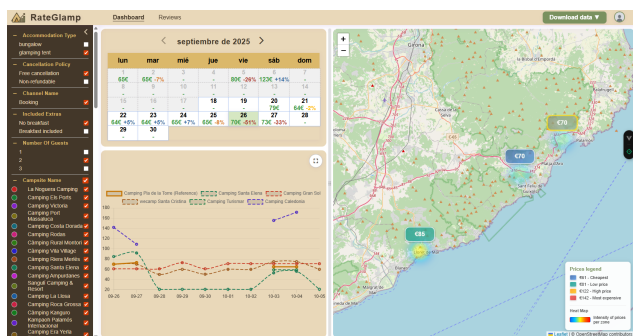


Fig. 1: Captura de pantalla del *dashboard* principal del rate shopper per glamping.

L'arquitectura de components permet mantenir vistes independents (*dashboard*, ressenyes, configuració) carrega-

des dinàmicament a través de *vue-router*. Aquest últim, a més, facilita la navegació sense recàrregues i la protecció de rutes restringides.

Firebase gestiona l'autenticació i permisos d'usuari, juga un paper clau en la seguretat: permet delegar la gestió d'autenticació, permisos (*claims*), recuperació de contrasenyes i altres operacions sensibles a la infraestructura de Google, simplificant considerablement el desenvolupament [20].

6.3 Evolució del projecte

Durant el desenvolupament del projecte s'han afegit nous components i funcionalitats per complir amb els objectius plantejats. A continuació es detallen les principals novetats introduïdes.

6.3.1 Model predictiu d'ocupabilitat

S'ha desenvolupat un model de *forecasting* d'ocupabilitat utilitzant Python en un Jupyter Notebook que després s'ha adaptat per a la seva execució en un contenidor Docker.

Les principals llibreries són: pandas, numpy, sklearn, tensorflow, keras, statsmodels, matplotlib, prophet

S'han implementat múltiples models de *forecasting* (ARIMA, Prophet i LSTM). L'avaluació comparativa mitjançant MAE i RMSE ha permès seleccionar els models més adients per al comportament de les sèries temporals reals.

Pandas facilita la preparació i manipulació eficient de les dades, mentre que Matplotlib permet la visualització comparativa entre prediccions i dades observades.

6.3.2 ARIMA

Un model ARIMA (Autoregressive Integrated Moving Average) permet analitzar i predir sèries temporals mitjançant la combinació de tres components: l'autoregressiu (AR), que utilitza valors passats de la sèrie; el d'integració (I), que aplica diferenciacions per eliminar tendències; i el de mitjana mòbil (MA), que incorpora la dependència dels errors passats. [21]

6.3.3 Prophet

Prophet és un procediment de predicció de sèries temporals basat en un model additiu on les tendències no lineals s'ajusten a l'estacionalitat anual, setmanal i diària, a més dels efectes de les vacances. Funciona especialment bé en sèries amb forts patrons estacionals i diverses temporades de dades històriques. Prophet és robust a les dades perdudes i als canvis en la tendència, i normalment gestiona bé els valors atípics. [22]

A part de la sèrie temporal, Prophet permet incorporar diverses variables externes com a regressors que complementen la sèrie, cadascuna amb el seu pes ajustable. Aquests permeten al model captar millor les variacions en la sèrie temporal que no es poden explicar per si mateixes.

6.3.4 LSTM

Long short-term memory (LSTM) és una tècnica per a millorar el desenvolupament de la intel·ligència artificial for-

mada per xarxes neuronals recurrents (RNN). La unitat bàsica LSTM està formada per una cèl·lula de memòria, una porta d'entrada, una porta de sortida i una porta d'oblit. S'apliquen en els camps de la classificació i predicció estadística sobre un conjunt de dades temporals. [23]

En aquest cas, LSTM no permet incorporar explícitament regressors ni variables amb festius, sinó que s'afegeixen de manera separa el que s'anomenen *features* o característiques.

6.3.5 Ampliació de la pàgina web

Abans de començar el desenvolupament, s'ha dissenyat el *layout* i els components principals del nou *dashboard* de prediccions amb Figma afegint la nova funcionalitat al diagrama existent per validar la distribució i coherència visual amb l'eina actual.

S'han afegit nous components al *front end* per incorporar el *dashboard* de prediccions: un calendari interactiu, gràfiques comparatives i filtres per tipus d'allotjament, horitzó temporal i data d'inici. Es mostra una captura de pantalla a la Figura 4 de l'Apèndix A.4.

L'accés a aquesta secció està limitat als usuaris amb el pla *Premium*, controlat mitjançant *claims* a Firebase. A més, s'ha ampliat el *dashboard* administratiu (Figura 5 de l'Apèndix) per permetre gestionar aquest pla de forma senzilla.

6.3.6 Adaptació del *back end*

S'ha dissenyat una arquitectura escalable per generar les prediccions setmanalment utilitzant AWS ECS i Lambda. Un cop els contenidors ECS dels *scrapers* finalitzen, una *rule* d'AWS activa una Lambda que crea un lot de treball per usuari i l'encua a SQS. Cada missatge desencadena un contenidor ECS que executa el model predictiu i desa els resultats a Amazon RDS. Aquesta arquitectura es pot veure a la Figura 6 de l'Apèndix A.5.

Finalment, s'han creat nous *endpoints* a l'API per donar suport al *dashboard* de prediccions, mantenint l'arquitectura modular basada en microserveis.

7 IMPLEMENTACIÓ

En aquesta secció es detallen els aspectes més rellevants de la implementació dels models predictius i la seva integració en el sistema complet. S'explica el procés seguit per desenvolupar, entrenar i validar els models, així com les decisions tècniques preses durant la construcció del *back end*, l'API i el *front end*.

7.1 Models predictius d'ocupabilitat

El primer de tot va ser estudiar les dades i entendre les tendències del càmping de referència. Cal remarcar que les dades només contenien dues temporades completes (2024 i 2025), fet que va dificultar l'entrenament dels models.

7.1.1 ARIMA

El primer de tot és comprovar l'estacionarietat de les dades mitjançant la prova d'*Augmented Dickey-Fuller*. Un cop comprovada l'estacionarietat, es van identificar els

paràmetres òptims (p , d , q) mitjançant l'anàlisi de les funcions d'autocorrelació (ACF) i autocorrelació parcial (PACF). Per determinar els valors òptims dels paràmetres p i q , s'han provat diverses combinacions i s'ha escollit el model amb millors valors dels criteris AIC (Akaike Information Criterion) i BIC (Bayesian Information Criterion). Aquests indicadors mesuren la qualitat del model equilibrant l'ajust i la simplicitat: com més baixos són, millor és el model. [24] Després es va entrenar el model ARIMA amb els paràmetres seleccionats i es van realitzar prediccions per a l'horitzó desitjat. Finalment, es va avaluar el rendiment del model fent predicció sobre el conjunt de dades de *train* i mesurant el RMSE, i fent una predicció de 365 dies.

7.1.2 Prophet

Es va preparar la sèrie temporal amb les columnes *ds* (data) i *y* (ocupació). A més, es van afegir diversos regressors externs que podrien influir en l'ocupació, com ara festius, temporada alta/baixa i dades sobre el clima (temperatura, precipitació).

Un cop preparades les dades, es va entrenar el model Prophet i es van realitzar prediccions per a l'horitzó desitjat. Finalment, es va avaluar el rendiment del model fent predicció sobre el conjunt de dades de *train* i mesurant el RMSE, i fent una predicció de 365 dies.

7.1.3 LSTM

S'han desenvolupat diversos models LSTM amb diferents arquitectures i paràmetres, cadascun amb els seus avantatges i inconvenients, ja que semblen més prometedors per aquest cas d'ús encara que siguin més complexos de desenvolupar i entrenar. Quasi tots comparteixen una estructura bàsica similar, dues branques separades que s'uneixen abans de la capa de sortida. Una branca s'encarrega de processar la sèrie temporal d'ocupació i l'altra les *features* addicionals. La branca d'ocupació té una finestra temporal de vint-i-un dies per capturar tendències de tres setmanes.

Primer es va desenvolupar un model que només utilitzava la sèrie temporal d'ocupació com a input i se li afegien 12 *features* d'estacionalitat (els quals seran els mateixos per a tots els models següents). A l'apèndix A.8 es poden veure totes les *features* que s'han contemplat per als diferents models, en concret les d'estacionalitat es poden veure a la Taula 3. També es pot veure l'arquitectura del model a la Figura 21 de l'Apèndix.

Cal remarcar que les *features* d'estacionalitat que són cícliques (com el dia de la setmana o mes de l'any) s'han codificat mitjançant funcions sinus i cosinus per tal que el model pugui captar millor la seva naturalesa cíclica. Per exemple, el dia de la setmana es codifica com dos valors: un amb la funció sinus i l'altre amb la funció cosinus del dia de la setmana normalitzat entre -1 i 1. Això es fa perquè el model entengui que el dia 7 (diumenge) està a prop del dia 1 (dilluns) i no llunyà com passaria si s'utilitzés una codificació ordinal simple.

Després es va decidir afegir el preu mitjà diari del càmping com a *feature* addicional, ja que és un factor que influeix en l'ocupació, a aquest model se l'anomena *Multi-variant*. Això fa que s'hagi de canviar lleugerament l'arquitectura i que s'hagin d'afegir 4 noves *features* relacionades

amb el preu que es poden veure a la Taula 4. L'arquitectura del model es pot veure a la Figura 22 de l'Apèndix. A partir del pressupost de l'any següent (2026) es van generar prediccions d'ocupació per a tota la temporada.

Seguidament, es va redefinir la manera com s'integra el preu, en comptes de com a una *feature* addicional, es va integrar dins de la sèrie temporal d'entrada amb l'objectiu de què el model pugui captar millor la relació entre preu i ocupació. En aquesta nova configuració, cada finestra temporal inclou simultàniament els valors històrics d'ocupació, el preu diari i la ràtio entre el preu actual i la mitjana mòbil dels darrers dies. A aquest model se l'anomena *Bivariant*. També s'ha fet que el model s'entreni més lentament i amb més paciència a l'hora de fer *early stopping*. L'arquitectura del model es pot veure a la Figura 23 de l'Apèndix. També es van generar prediccions d'ocupació per a tota la temporada 2026.

Igual que amb el model desenvolupat amb Prophet, també s'ha creat una versió del model LSTM que incorpora variables meteorològiques. Aquestes variables són rellevants per a l'ocupació i sobretot per a les cancel·lacions. Les dades atmosfèriques s'han obtingut mitjançant l'API Open-Meteo Archive, que proporciona registres històrics de variables climàtiques per a qualsevol rang temporal [25]. Les *features* meteorològiques que s'han afegit es poden veure a la Taula 5 on se sumen a les de preu i estacionalitat. L'arquitectura està basada en el model *Bivariant*, però es va dissenyar tenint en compte que la inclusió de variables meteorològiques podia dificultar l'estabilitat de l'entrenament, motiu pel qual es va optar per una estructura lleugerament més senzilla. Aquesta arquitectura es pot veure a la Figura 24 de l'Apèndix. Amb aquest model, però només es poden fer prediccions a curt termini (fins a quinze dies), ja que les dades meteorològiques futures no es poden conèixer amb exactitud.

A l'hora d'optimitzar preus, cal que el model pugui estimar l'ocupació esperada per a cada preu ofert en una data concreta. Així, es genera una sèrie de preus, el model prediu l'ocupació per a cadascun, i s'escull el preu que maximitza els ingressos. Tanmateix, aplicar aquesta tècnica directament sobre models entrenats amb dades reals va provocar un problema: el dataset mostrava una correlació positiva entre preu i ocupació (més preu, més ocupació), ja que a l'estiu coincideixen demanda alta i preus elevats. Això va fer que el model aprengué una relació incorrecta i l'optimitzador recomanés sempre el preu més alt, impeding una optimització realista. Per solucionar aquest comportament i forçar el model a aprendre una elasticitat coherent (on un augment del preu redueix la demanda esperada) es va decidir generar dades sintètiques basades en les dades reals.

El model creat a partir de les dades sintètiques està basat en el model *Bivariant* on la seqüència d'entrada està formada per la sèrie temporal d'ocupació, preu i un indicador relatiu al preu respecte a la seva mitjana mòbil de 30 dies. També s'afegeixen les mateixes *features* d'estacionalitat i la tendència del preu. L'arquitectura del model es pot veure a la Figura 25 de l'Apèndix.

Per últim, per tal d'entendre millor el comportament dels diferents models, s'ha mesurat la importància de les diferents *features* utilitzades en els models LSTM mitjançant permutacions. Aquesta tècnica consisteix a mesurar la pèrdua de rendiment del model quan es barregen els valors

d'una *feature* específica, mantenint les altres constants. Si la pèrdua de rendiment és significativa, indica que aquesta *feature* és important per a les prediccions del model. Aquesta anàlisi ajuda a identificar quines variables tenen més impacte en l'ocupació i a comprendre millor el funcionament intern dels models [26].

7.2 Ampliació de la pàgina web

Es va dissenyar el *layout* i els components principals del nou *dashboard* de prediccions amb Figma. També es va fer una prova d'usabilitat amb usuaris reals per recollir feedback i ajustar detalls abans de la implementació.

S'han afegit nous components al *front end* per incorporar el *dashboard* de prediccions: un calendari interactiu, gràfiques comparatives i filtres per tipus d'allotjament, horitzó temporal i data d'inici. Tot plegat es mostra a la figura 4.

L'accés a aquesta secció està limitat als usuaris amb el pla *Premium*, controlat mitjançant *claims* a Firebase. A més, s'ha ampliat el *dashboard* administratiu (Figura 5) per permetre gestionar aquest pla de forma senzilla.

7.3 Adaptació del *back end*

Un cop validats els models, es va dissenyar una arquitectura escalable per generar les prediccions setmanalment, utilitzant AWS ECS com a motor d'execució i Lambda per orquestrar el procés.

Es va crear un contenidor Docker amb el *pipeline* complet de generació de prediccions, que inclou l'obtenció de dades de RDS, el preprocessament, l'entrenament amb dades històriques, la generació de prediccions per a l'horitzó desitjat i l'emmagatzematge dels resultats a RDS. El contenidor es va validar localment amb dades reals abans de desplegar-lo a AWS, es va pujar a ECR (Elastic Container Registry) i es va definir una tasca ECS basada en aquest contenidor.

L'orquestració es va implementar mitjançant una *rule* d'AWS EventBridge que s'activa en finalitzar els *scrapers*. Aquesta *rule* desencadena una funció Lambda que crea un lot de treball per a cada usuari amb pla Premium i l'encua a SQS, on cada missatge conté la informació necessària per executar una tasca ECS individual. Una *AWS Pipe* s'encarrega de llançar un contenidor ECS per cada missatge, garantint una execució independent i escalable per usuari. L'arquitectura completa es detalla a l'Apèndix A.5.

Finalment, es van crear nous *endpoints* a l'API per donar suport al *dashboard* de prediccions, permetent recuperar les prediccions emmagatzemades a RDS i proporcionar-les al *front end* per a la seva visualització. Els nous *endpoints* són:

- **GET /forecasting/filters:** Retorna els filtres disponibles per al *dashboard* de prediccions, com ara tipus d'allotjament i horitzó temporal.
- **GET /forecasting/calendar:** Retorna les dates amb prediccions disponibles per a un tipus d'allotjament i horitzó temporal específics per al càmping de referència.

- **GET /forecasting/graphic:** Retorna les dades de predicció i dades reals per a un rang de dates, tipus d'allotjament i horitzó temporal específics per al càmping de referència.
- **GET /forecasting/download:** Genera i retorna un informe en format PDF, Excel o CSV amb les dades de predicció i dades reals per a un rang de dates, tipus d'allotjament i horitzó temporal específics per al càmping de referència, depenent si estàs a la vista de calendari o gràfica.

Tambe s'han afegit nous *endpoints* per gestionar els plans d'usuari Premium des del *dashboard* administratiu i perquè l'usuari pugui gestionar la configuració relativa a les prediccions des del seu perfil d'usuari.

8 RESULTATS

En aquesta secció es presenten els resultats obtinguts durant el desenvolupament del projecte, incloent l'avaluació dels models predictius d'ocupabilitat i la integració de la nova funcionalitat a la pàgina web existent.

8.1 Models predictius d'ocupabilitat

El primer que s'observa en les dades és que l'ocupació està fortament marcada pels caps de setmana. Un altre aspecte amb rellevància és la tendència a l'alça de juny, on l'ocupació va escalant a poc a poc fins a arribar al juliol i agost on es donen els pics més alts i hi ha una certa estabilitat al voltant del 80%. Al setembre sobta com cau l'ocupació quasi al 20% i segueix fortament marcada pels caps de setmana. Veure gràfica de l'Apèndix A.5.1.

A l'Apèndix A.6 es poden veure els resultats complets de tots els models desenvolupats. A continuació es detallen els resultats més rellevants de cada model.

8.1.1 ARIMA

A la Figura 8 es pot veure la predicció realitzada pel model ARIMA juntament amb les dades reals del càmping de referència i la predicció feta sobre les dades de *train*. Es pot observar que el model segueix bastant bé la tendència de les dades de *train* que ja ha vist amb un RMSE d'11.41. Pel que fa a la predicció a llarg termini, el model sembla que no aconsegueix captar bé la tendència de les dades reals i la predicció és plana menys al principi on hi ha un riçat.

8.1.2 Prophet

A la Figura 10 es pot veure la predicció realitzada pel model Prophet juntament amb les dades reals del càmping de referència i la predicció feta sobre les dades de *train*. Es pot observar que el model segueix bastant bé la tendència de les dades de *train* que ja ha vist amb un RMSE de 9.32.

8.1.3 LSTM

El model LSTM que només utilitza la sèrie temporal d'ocupació com a input aconsegueix un RMSE d'11.72 sobre les dades de *train*, però la predicció a llarg termini no és gaire bona, ja que no aconsegueix captar bé la tendència de les

dades reals, en concret, fa prediccions baixes. Els resultats es poden veure a la Figura 11.

El següent model desenvolupat és el *Multivariant*, que obté un RMSE de 7.62 sobre les dades de *train*. Pel que fa a les dades de *test*, el RMSE és de 10.5 com es veu a la Figura 12.

El següent model desenvolupat és el *Bivariant*. Aquest model aconsegueix un RMSE de 6.18 sobre les dades de *train*. Pel que fa a les dades de *test*, el RMSE és de 10.63 com es veu a la Figura 14.

El penúltim model desenvolupat és el que incorpora variables meteorològiques. Aquest model aconsegueix un RMSE de 8.4 sobre les dades de *train*. Pel que fa a les dades de *test*, el RMSE és de 14.91 com es veu a la Figura 16.

L'últim model desenvolupat és el que utilitza dades sintètiques per entrenar-se i així poder fer optimització de preus. Aquest model aconsegueix un RMSE de 7.91 sobre les dades de *train*.

8.2 Ampliació de la pàgina web

El nou *dashboard* de prediccions s'ha integrat amb l'API existent mitjançant els nous *endpoints*, permetent obtenir i visualitzar les dades de predicció. El calendari interactiu permet seleccionar dates disponibles, mentre que la gràfica comparativa mostra les prediccions juntament amb dades reals passades per facilitar l'anàlisi visual, oferint una visió clara de l'ocupació prevista.

El sistema de filtres permet personalitzar la visualització segons el tipus d'allotjament, l'horitzó temporal i la data d'inici. A més, la descàrrega d'informes en formats PDF, Excel i CSV facilita la compartició i l'anàlisi *offline*.

La integració amb Firebase assegura que només els usuaris amb el pla Premium puguin accedir a aquesta funcionalitat, mantenint la seguretat i exclusivitat del servei ofert.

8.3 Adaptació del back end

La nova arquitectura per a la generació setmanal de prediccions ha demostrat ser escalable i eficient. L'ús d'AWS ECS permet escalar dinàmicament el nombre de contenidors segons la demanda.

La funció Lambda encarregada de l'orquestració simplifica la gestió dels lots de treball, mentre que SQS actua com a buffer de missatges, aportant resiliència mitjançant reintents automàtics en cas d'errors.

Els nous *endpoints* de l'API faciliten la integració amb el *front end*, garantint una experiència fluida i coherent. La modularitat basada en microserveis permet incorporar noves funcionalitats sense afectar les operacions existents i compleix l'objectiu de temps de resposta, amb latències inferiors als 300 ms al *dashboard* de prediccions.

9 DISCUSSIÓ DELS RESULTATS

En aquesta secció s'analitzen els resultats del projecte, amb especial èmfasi en l'avaluació dels models predictius d'ocupabilitat, ja que la integració de la nova funcionalitat a la pàgina web no presenta aspectes rellevants a discutir.

En el cas del model ARIMA, la raó per la qual la predicció a llarg termini és plana és que el model no és capaç de

captar bé l'estacionalitat de les dades amb només dues temporades completes i per aquest motiu no és el model més adequat per aquest cas d'ús.

Prophet funciona descomponent la sèrie temporal en diversos components: tendència, estacionalitat diària, setmanal i anual, i altres regressors. Aquesta descomposició radica de la teoria del senyal on es considera que una sèrie temporal és la suma de diversos components que poden ser modelats per separat com a suma de sinusoides. Aquesta descomposició es pot veure a la Figura 9 on es pot observar com el model ha captat bé l'estacionalitat setmanal i anual de les dades, però no ha captat bé la tendència que afegeixen els regressors externs. Pel que fa a la predicció a llarg termini, el model sembla que no aconsegueix captar bé la tendència de les dades reals i la predicció és artificialment creada a partir dels regressors afegits més l'estacionalitat setmanal que crea el ritme als caps de setmana. Per aquest motiu tampoc és el model més adequat per aquest cas d'ús.

Pel que fa als models LSTM, el model inicial va ser una primera aproximació per veure com funcionava LSTM amb aquestes dades i si el projecte era viable. Aquest model es va descartar ràpidament però va assentar la base dels següents.

En el cas del model *Multivariant*, que afegeix el preu mitjà diari del càmping com a *feature* addicional, afegir més variables implica que el model és més costós d'entrenar; tot i això, la millora d'un 35% respecte al model anterior sobre dades de *train* implica que les prediccions que fa aquest model ja són millors que les del model anterior sobre dades conegudes. Es pot veure una comparació del model *Multivariant* amb l'anterior a la Figura 13.

Respecte al model *Bivariant*, que integra el preu dins de la sèrie temporal d'entrada, tot i millorar un 19% respecte al model anterior en *train*, les prediccions sobre *test* són lleugerament pitjors. Tot i això, s'ha considerat que aquest model és millor, ja que el seu entrenament es fa més lentament (més *epochs* i amb més paciència a l'hora de fer *early stopping*) i s'ha comprovat que és més estable i robust a l'hora de fer prediccions. Mentre que l'anterior model donava prediccions molt variables depenent de l'entrenament (entre 9 i 7 d'RMSE sobre les dades de *train*), aquest sempre dona resultats similars (entre 6.5 i 6 d'RMSE sobre les dades de *train*). Es pot veure una comparació del model *Bivariant* amb els anteriors a la Figura 15.

En el model que incorpora variables meteorològiques, el fet que el RMSE de *test* sigui elevat implica que les prediccions que fa aquest model són pitjors que les dels models anteriors. S'ha considerat que aquest model no és adequat per aquest cas d'ús, ja que les dades meteorològiques no aporten gaire informació addicional.

Pel que fa al model amb dades sintètiques, encara que és pitjor en termes de RMSE, cal recordar que aquest model està entrenat per fer optimització de preus específicament amb el triple de dades, per tant, només és útil per a aquest propòsit. És un model també molt consistent en el seu re-entrenament, ja que sempre dona resultats similars (entre 7.5 i 8 d'RMSE sobre les dades de *train*). L'important sobre aquest model és que s'ha aconseguit que aprengui una elasticitat coherent (on un augment del preu redueix la demanda esperada) i, per tant, l'optimitzador de preus funciona correctament, permetent trobar el preu que maximitza els ingressos. Es pot veure a la Figura 17 com per a un dia

concret, a mesura que s'augmenta el preu ofert, l'ocupació esperada disminueix. Amb aquest mètode es poden optimitzar els preus de tota la temporada que després es mostraran al *dashboard* de prediccions, es pot veure la optimització de preus de tota la temporada 2026 a la Figura 18.

Per últim, es pot veure una comparació completa de tots els models a la Figura 19. Observant la comparació completa de tots els models, es veu com el model *Bivariant* és el que millor s'ajusta a les dades reals i, per tant, el model escollit per a fer les prediccions d'ocupació del càmping de referència.

Pel que fa a la importància de les diferents *features* utilitzades en els models LSTM, els resultats es poden observar a la Figura 20, les *features* més importants generalment són les relacionades amb l'estacionalitat, dia de la setmana i si es cap de setmana sobretot, ja que són les que marquen els pics dels caps de setmana, per exemple. Tot i així hi ha algunes *features* que no aporten gaire informació al model i es podrien eliminar per simplificar-lo, com per exemple el mes de l'any.

En el model *Multivariant*, la tercera *feature* més important és el preu mitjà diari del càmping, el que indica que aquesta variable té un impacte significatiu en l'ocupació i és rellevant per a les prediccions del model.

En el model *Bivariant*, la quarta *feature* més important és la ràtio entre el preu actual i la mitjana mòbil dels darrers dies, el que indica que aquesta variable també té un impacte significatiu en l'ocupació i és rellevant per a les prediccions del model, en canvi, el preu en si mateix no és tan rellevant.

El model amb dades meteorològiques mostra que les variables relacionades amb la precipitació i la temperatura tenen una importància molt baixa, el que suggereix que aquestes variables no aporten gaire informació addicional al model per a les prediccions d'ocupació, cosa que sobta perquè es pensava que aquestes variables serien més rellevants. Segurament serà perquè s'utilitzen dades de reserves ja fetes i no es tenen en compte les cancel·lacions que poden dependre més del clima.

Per al model entrenat amb dades sintètiques, la segona *feature* més important és la ràtio entre el preu actual i la mitjana mòbil dels darrers dies. En aquest model que és molt semblant al *Bivariant*, aquesta variable té un impacte significativament major en l'ocupació com també el preu en si, cosa que té sentit, ja que aquest model està dissenyat específicament per a l'optimització de preus.

10 CONCLUSIÓ I TREBALL FUTUR

Aquest treball ha assolit l'objectiu principal d'evolucionar un *rate shopper* bàsic cap a una plataforma completa de *Revenue Intelligence* especialitzada en el sector del càmping. A través de la integració de models predictius d'intel·ligència artificial, s'ha aconseguit transformar una eina de monitoratge de preus en un sistema capaç d'anticipar la demanda i optimitzar la presa de decisions estratègiques.

10.1 Assoliments principals

Els models predictius desenvolupats han permès generar pronòstics per a la temporada 2026 que coincideixen amb les expectatives del càmping de referència. Tot i això, cal

remarcar que la disponibilitat de només dues temporades completes d'històric constitueix el principal *handicap* del projecte. Aquesta limitació restringeix la capacitat dels models per captar amb total precisió les tendències i estacionalitats complexes inherents al sector. Malgrat aquesta restricció, els models LSTM han demostrat una capacitat notable per captar els patrons d'ocupació, especialment el model *Bi-variant*, que ha aconseguit un equilibri entre precisió (RMSE de 6.18 sobre dades d'entrenament) i estabilitat en les prediccions.

Un dels resultats més rellevants ha estat el desenvolupament del model d'optimització de preus entrenat amb dades sintètiques, que ha permès aprendre una elasticitat preu-demanda coherent. Aquesta funcionalitat ha habilitat la generació automàtica de recomanacions de preus per a tota una temporada, transformant un procés que abans requeria setmanes d'anàlisi manual en un càlcul instantani accessible amb un sol clic.

La plataforma desenvolupada integra de manera eficient la infraestructura de núvol (AWS ECS, Lambda, RDS), l'API RESTful i el *dashboard* web amb Vue.js, oferint una experiència d'usuari fluida i escalable. El sistema de generació setmanal de prediccions mitjançant contenidors ECS garanteix que cada usuari Premium disposi de pronòstics actualitzats sense comprometre el rendiment de la plataforma.

10.2 Treball futur i millores proposades

Malgrat els resultats positius obtinguts, s'han identificat diverses línies de millora orientades a incrementar la precisió, la robustesa i la utilitat operativa de la plataforma.

10.2.1 Millora dels models predictius

La principal limitació actual és la disponibilitat de només dues temporades completes d'històric, fet que dificulta que els models captin adequadament la variabilitat interanual i les tendències a llarg termini. L'ampliació d'aquest històric és clau per millorar la robustesa i la fiabilitat dels pronòstics.

A més, la incorporació de noves *features* externes podria millorar significativament el rendiment dels models LSTM, com ara els preus i les puntuacions de les ressenyes dels càmpings competidors (ja recopilades pels *scrapers*, però amb històric insuficient), la presència d'esdeveniments especials a la zona, les prediccions meteorològiques històriques i altres variables climàtiques addicionals com la velocitat del vent.

L'anàlisi d'importància de *features* ha posat de manifest que alguns regressors aporten poca informació, de manera que la seva eliminació podria reduir l'*overfitting* i millorar la capacitat de generalització, especialment en el model meteorològic. Paral·lelament, es proposa explorar arquitectures alternatives als models actuals, com l'ús de diferents funcions d'activació, esquemes de normalització o arquitectures híbrides que combinin LSTM amb capes convolucionals per captar millor els patrons temporals.

10.2.2 Validació i escalabilitat del sistema

És necessari validar el comportament del sistema davant un increment significatiu del nombre d'usuaris Premium

mitjançant proves de càrrega sobre la infraestructura AWS, especialment els contenidors ECS, per garantir que el rendiment es manté estable amb l'escalabilitat. També es recomana realitzar sessions d'usabilitat amb gestors de càmpings per verificar que el *dashboard* de prediccions és intuïtiu i que la informació es presenta de manera clara i útil per a la presa de decisions.

Finalment, cal implementar un sistema de monitoratge en producció que permeti comparar les prediccions amb les ocupacions reals, detectar possibles degradacions en la qualitat dels models i activar processos de reentrenament quan sigui necessari.

10.2.3 Funcionalitats addicionals

Com a línia d'evolució funcional, es proposa la integració de la plataforma amb els *Property Management Systems* dels càmpings per automatitzar l'aplicació de les recomanacions de preus i reduir encara més la càrrega de treball manual.

10.3 Reflexió final

Aquest projecte ha demostrat la viabilitat i el valor d'aplicar tècniques avançades d'intel·ligència artificial al sector del càmping, un àmbit tradicionalment menys digitalitzat que l'hoteler. La plataforma desenvolupada no només resol una necessitat real del mercat, sinó que estableix les bases tecnològiques i metodològiques per continuar evolucionant cap a una eina cada vegada més sofisticada i útil.

Els resultats obtinguts confirmen que, tot i les limitacions de dades, és possible crear models predictius fiables que aportin valor immediat als usuaris. A mesura que s'acumulin més dades històriques i s'implementin les millores proposades, s'espera que la precisió dels models i la utilitat de la plataforma continuïn creixent, consolidant *Rateglamp* com una eina de referència per al *revenue management* en el sector del càmping.

AGRAÏMENTS

Agraeixo aquest treball a la persona que va idear *Rateglamp* i que l'ha fet possible, en Daniel Badenas, qui m'ha donat l'oportunitat de crear aquesta eina des de zero, fent que en el procés aprengué moltíssim sobre desenvolupament de *software*, intel·ligència artificial i *revenue management* en el sector del càmping. També vull agrair al meu tutor acadèmic, en Juan Antonio Martinez Carrascal per la seva guia i suport durant tot el procés de realització d'aquest treball.

US DE LA IA

S'ha emprat la IA generativa en aquest treball per millorar la redacció, formulació i estil de certes seccions per tal de millorar la lectura utilitzant el prompt: 'Millora aquest paràgraf donant tips per redactar millor en to formal però sense canviar molt les frases.'

També s'ha utilitzat la IA per a la correcció d'errors i *bugs* en el desenvolupament del codi.

REFERÈNCIES

- [1] Rate shopper dataria - dataria revenue management software. URL <https://www.dataria.com/es/soluciones/rate-shopper/>.
- [2] What is revenue intelligence? — salesforce. URL <https://www.salesforce.com/eu/sales/revenue-intelligence/what-is-revenue-intelligence/>.
- [3] Wikipedia contributors. Mean absolute error — Wikipedia, the free encyclopedia, 2025. URL https://en.wikipedia.org/w/index.php?title=Mean_absolute_error&oldid=1276071917. [Online; accessed 24-January-2026].
- [4] Wikipedia contributors. Root mean square deviation — Wikipedia, the free encyclopedia, 2025. URL https://en.wikipedia.org/w/index.php?title=Root_mean_square_deviation&oldid=1326682842. [Online; accessed 27-January-2026].
- [5] Wikipedia contributors. Linear regression — Wikipedia, the free encyclopedia, 2025. URL https://en.wikipedia.org/w/index.php?title=Linear_regression&oldid=1327852137. [Online; accessed 24-January-2026].
- [6] Hotel forecasting: Improve demand predictions with ai - beonx. URL <https://beonx.com/forecasting/>.
- [7] Revenue forecasting for hotels, casinos or resorts. URL <https://www.duettocloud.com/es/solutions/forecasting/>.
- [8] Hotel budgeting and forecasting software — ideas revplan. URL <https://ideas.com/hotel-budgeting-and-forecasting/>.
- [9] How predictive market intelligence improves hotel demand forecasting. URL <https://www.mylighthouse.com/resources/blog/hotel-demand-forecasting>.
- [10] Hotel budgeting and forecasting software by actabl. URL <https://actabl.com/business-intelligence-software/forecasting-budgeting/>.
- [11] Demand360® hotel business intelligence · amadeus hospitality. URL <https://www.amadeus-hospitality.com/solutions/business-intelligence/demand360/>.
- [12] Wikipedia. Desarrollo en cascada — wikipedia, la enciclopedia libre, 2024. URL https://es.wikipedia.org/w/index.php?title=Desarrollo_en_cascada&oldid=164306255. [Internet; descargado 25-diciembre-2024].
- [13] ¿qué es amazon vpc? - amazon virtual private cloud, . URL <https://docs.aws.amazon.com/es/es/vpc/latest/userguide/what-is-amazon-vpc.html>.
- [14] Aws — servicio de bases de datos relacionales (rds), . URL <https://aws.amazon.com/es/rds/>.
- [15] Wikipedia. Docker (software) — wikipedia, la enciclopedia libre, 2026. URL [https://es.wikipedia.org/w/index.php?title=Docker_\(software\)&oldid=171499433](https://es.wikipedia.org/w/index.php?title=Docker_(software)&oldid=171499433). [Internet; descargado 14-enero-2026].
- [16] Enrutador de eventos sin servidor - amazon eventbridge - aws, . URL <https://aws.amazon.com/es/eventbridge/>.
- [17] Amazon Web Services. Aws lambda - serverless computing service, 2024. URL <https://aws.amazon.com/lambda/>.
- [18] Colas de mensajes completamente administradas - amazon simple queue service – amazon web services, . URL <https://aws.amazon.com/es/sqs/>.
- [19] Amazon eventbridge pipes - amazon eventbridge, . URL <https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-pipes.html>.
- [20] Firebase — google’s mobile and web app development platform. URL <https://firebase.google.com/>.
- [21] Chapter 8 arima models — forecasting: Principles and practice (2nd ed). URL <https://otexts.com/fpp2/arima.html>.
- [22] Prophet — forecasting at scale. URL <https://facebook.github.io/prophet/>.
- [23] Long short-term memory - viquipèdia, l’enciclopèdia lliure, . URL https://ca.wikipedia.org/wiki/Long-short-term_memory.
- [24] Autoregressive integrated moving average - wikipedia, . URL https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average#cite_note-3-10.
- [25] open-meteo/open-meteo: Free weather forecast api for non-commercial use. URL <https://github.com/open-meteo/open-meteo?tab=readme-ov-file>.
- [26] Ava Mohammadi, Gonzalo Nápoles, and Yamisleydi Salgueiro. Deep learning for time-series forecasting with exogenous variables in energy consumption: A performance and interpretability analysis. *IE-EE Access*, 13:86746–86767, 2025. ISSN 21693536. doi: 10.1109/ACCESS.2025.3570618. URL <https://ieeexplore.ieee.org/document/11005580>.

APÈNDIX

A.1 Requisits Funcionals i No Funcionals

TAULA 1: REQUISITS FUNCIONALS DEL PROJECTE

ID	Descripció	Prioritat
MODELS PREDICTIUS		
RF-01	Desenvolupar un prototip de model predictiu d'ocupabilitat a partir de les dades del càmping de referència	O
RF-02	Avaluar diferents models predictius: ARIMA, Prophet i LSTM	D
RF-03	Mesurar el rendiment mitjançant MAE, RMSE i regressió lineal amb dades de <i>train</i> i <i>test</i>	O
RF-04	Seleccionar el model que millor s'ajusti a les dades segons les mètriques obtingudes	O
RF-05	Incorporar variables externes (temps atmosfèric, preu mitjà de l'allotjament, festius)	D
DASHBOARD DE PREDICCIONS		
RF-06	Implementar filtres al <i>dashboard</i> : tipus d'allotjament, horitzó temporal i data d'inici	O
RF-07	Desenvolupar un calendari interactiu amb ocupació real, ocupació predita, preu actual, preu suggerit i preu mitjà dels competidors	O
RF-08	Implementar una gràfica comparativa de prediccions amb dades històriques	O
RF-09	Permetre la descàrrega d'informes en format PDF, Excel i CSV	O
RF-10	Mostrar visualitzacions clares i interpretables de les prediccions	O
GESTIÓ DEL PLA PREMIUM		
RF-11	Afegir al <i>dashboard</i> d'administració una opció per assignar/revocar el pla <i>Premium</i> als clients	O
RF-12	Restringir l'accés al <i>dashboard</i> de prediccions exclusivament als usuaris <i>Premium</i>	O
RF-13	Permetre als usuaris gestionar la configuració de prediccions des del seu perfil	O
API I BACK END		
RF-14	Crear l'endpoint GET /forecasting/filters per retornar els filtres disponibles	O
RF-15	Crear l'endpoint GET /forecasting/calendar per retornar dates amb prediccions disponibles	O
RF-16	Crear l'endpoint GET /forecasting/graphic per retornar dades de predicció i dades reals	O
RF-17	Crear l'endpoint GET /forecasting/download per generar informes en diferents formats	O
RF-18	Implementar endpoints per gestionar els plans d'usuari <i>Premium</i> des del <i>dashboard</i> administratiu	O
GENERACIÓ DE PREDICCIONS		
RF-19	Implementar una Lambda Function per crear <i>batches</i> de treball setmanals	O
RF-20	Utilitzar contenidors ECS (un per usuari <i>Premium</i>) per generar prediccions	O
RF-21	Encuar tasques a SQS per gestionar l'execució dels contenidors ECS	O
RF-22	Emmagatzemar les prediccions generades a la base de dades Amazon RDS	O
RF-23	Activar automàticament la generació de prediccions amb EventBridge després dels <i>scrapers</i>	O
RF-24	Reentrenar els models amb noves dades històriques periòdicament	D

TAULA 2: REQUISITS NO FUNCIONALS DEL PROJECTE

ID	Descripció	Prioritat
RENDIMENT		
RNF-01	L'API ha de respondre a les peticions del <i>dashboard</i> de prediccions en menys de 300ms	O
ESCALABILITAT		
RNF-02	L'arquitectura ha de permetre afegir nous càmpings sense canvis estructurals	O
RNF-03	La base de dades Amazon RDS ha de ser escalable per suportar creixement de dades històriques	O
RNF-04	El sistema ha de poder escalar horitzontalment mitjançant l'addició de contenidors ECS	O
DISPONIBILITAT		
RNF-05	L'aplicació web ha d'estar disponible el 99% del temps (uptime)	O
RNF-06	El sistema ha de recuperar-se automàticament d'errors transitoris	D
SEGURETAT		
RNF-07	Autenticació d'usuaris mitjançant Firebase amb tokens JWT	O
RNF-08	Les dades sensibles (contrasenya, informació personal) han de ser xifrades amb SSL/TLS	O
RNF-09	Implementar control d'accés basat en rols (<i>claims</i>) per al pla <i>Premium</i>	O
RNF-10	Les peticions a l'API han de validar tokens d'autenticació vàlids	O
RNF-11	Registre d'accions administratives (assignació de plans <i>Premium</i>)	D
USABILITAT		
RNF-12	La interfície del <i>dashboard</i> ha de ser intuïtiva i fàcil d'interpretar	O
RNF-13	Les gràfiques i calendaris han de ser interactius i permetre zoom	O
RNF-14	Els colors i estils han de mantenir coherència amb l'eina existent (<i>Rateglamp</i>)	O
RNF-15	Proporcionar missatges d'error clars i accions suggerides	D
MANTENIBILITAT		
RNF-16	El codi ha de seguir estàndards de qualitat	O
RNF-17	L'arquitectura de microserveis ha de facilitar la modularitat i independència dels components	O
RNF-18	Documentar els endpoints de l'API amb exemples d'ús (Postman)	D
RNF-19	El codi dels models predictius ha d'estar ben comentat i estructurat	O
RNF-20	Utilitzar control de versions (Git) amb commits descriptius	O
COMPATIBILITAT		
RNF-21	L'aplicació web ha de ser compatible amb navegadors moderns (Chrome, Firefox, Safari, Edge)	O
RNF-22	Els contenidors Docker han de ser compatibles amb AWS ECS Fargate	O
RNF-23	Les llibreries utilitzades han de tenir versions estables i suport actiu	O
COMPLIMENT I REGULACIÓ		
RNF-24	Complir amb la normativa de protecció de dades (GDPR) per a dades d'usuaris europeus	O
RNF-25	Permetre als usuaris sol·licitar l'eliminació de les seves dades	O

A.2 Gantt de planificació

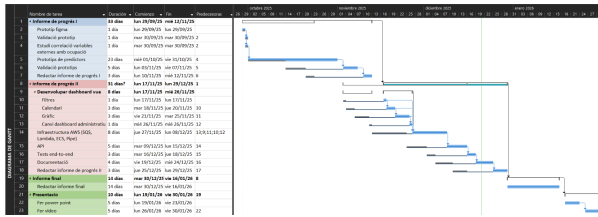


Fig. 2: Diagrama de Gantt de planificació, MS Project.

A.3 Gantt de seguiment

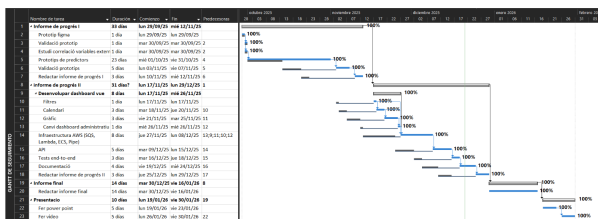


Fig. 3: Diagrama de Gantt de seguiment, MS Project

A.4 Captures de pantalla del dashboard de prediccions

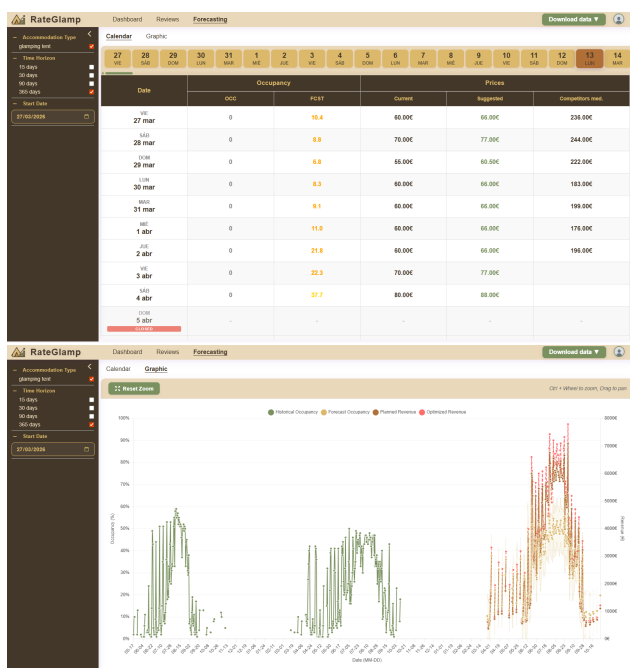


Fig. 4: Captures de pantalla del nou dashboard de prediccions: calendari i gràfica comparativa.

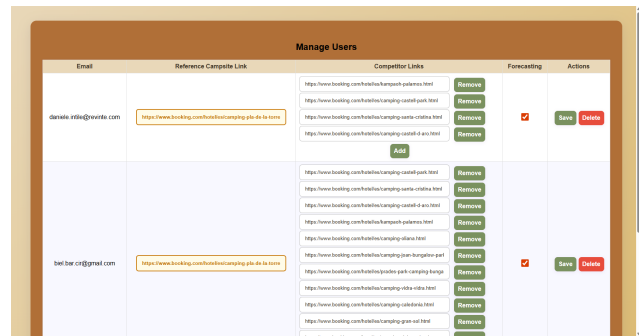


Fig. 5: Dashboard administratiu amb l'opció d'assignar el pla Premium.

A.5 Arquitectura de la solució a AWS

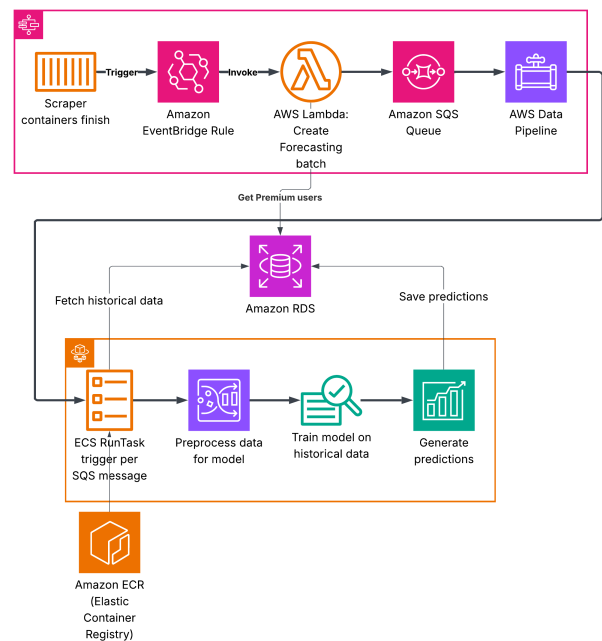


Fig. 6: Arquitectura de la solució implementada a AWS.

A.5.1 Ocupació Càmping de referencia

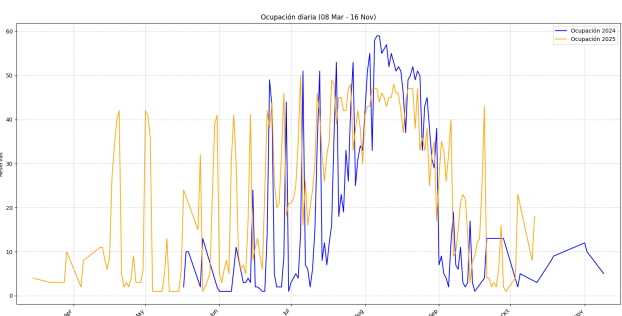


Fig. 7: Ocupació del càmping de referència durant 2024 i 2025.

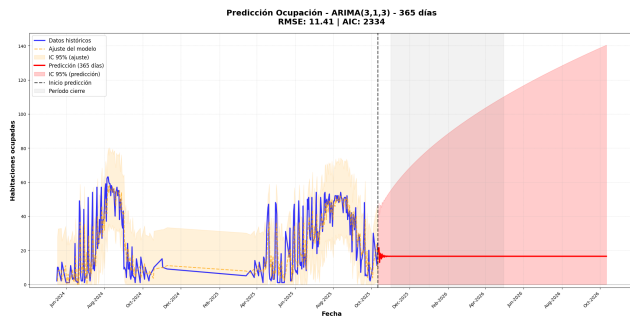


Fig. 8: Resultats de predicció del model ARIMA.

A.6 Resultats models predictius

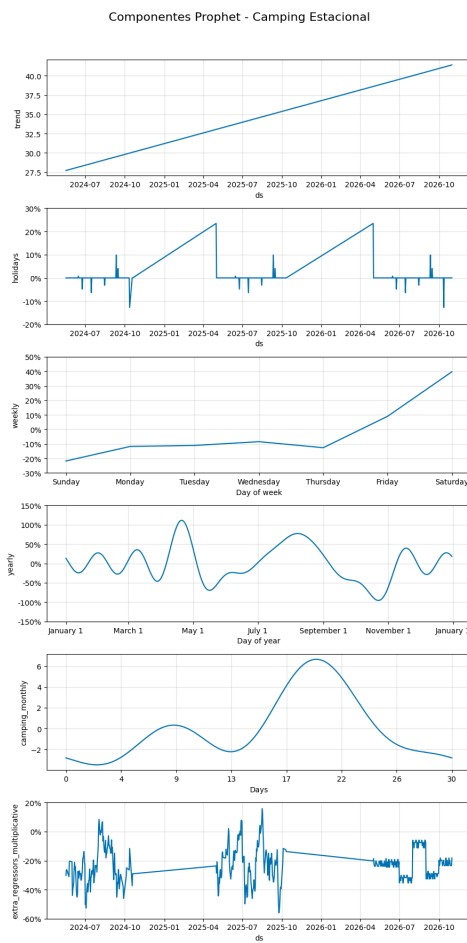


Fig. 9: Components del model Prophet.

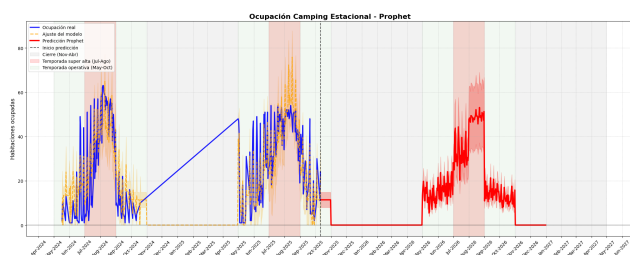
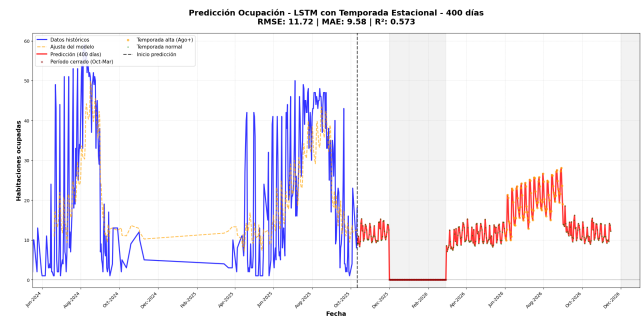
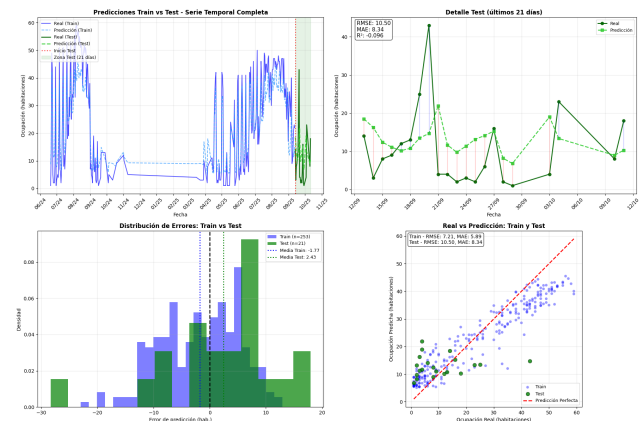
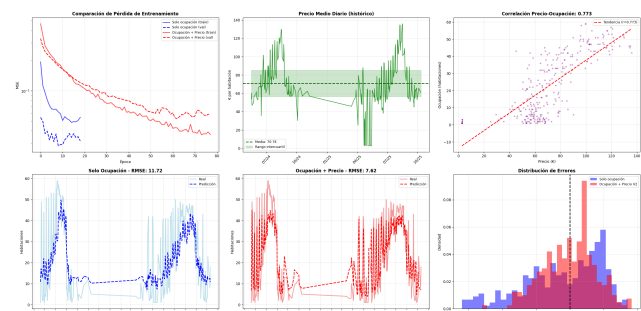


Fig. 10: Resultats de predicció del model Prophet.

Fig. 11: Resultats de predicció del model LSTM utilitzant només ocupació com a serie d'entrada i 12 *features* d'estacionalitat.Fig. 12: Resultats de predicció del model LSTM *Multivariant* utilitzant ocupació i preu mitjà diari com a sèries d'entrada i 12 *features* d'estacionalitat i 4 de preu, predicció de 21 dies i comparació amb les dades de *test*.Fig. 13: Comparativa entre els models LSTM d'ocupació i *Multivariant* sobre les dades de *train*.

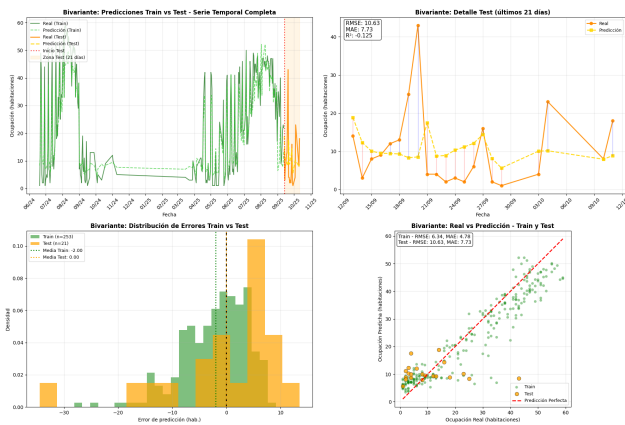


Fig. 14: Resultats de predicció del model LSTM *Bivariant* integrant ocupació i preu mitjà diari dins de la sèrie temporal d'entrada i 12 *features* d'estacionalitat i 4 de preu, predicció de 21 dies i comparació amb les dades de *test*.

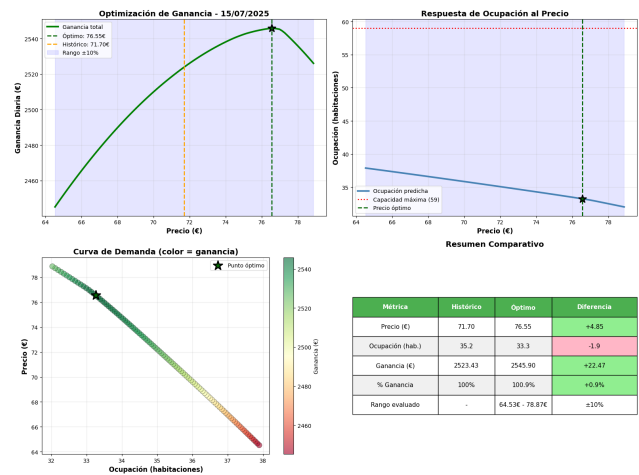


Fig. 17: Optimització de preu per a un dia concret utilitzant el model LSTM entrenat amb dades sintètiques.

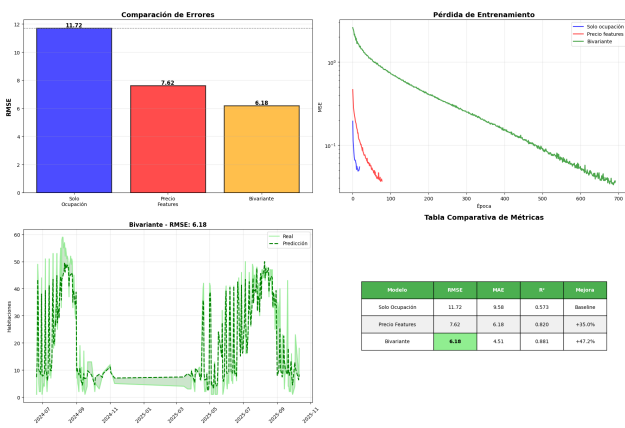


Fig. 15: Comparativa entre els models LSTM d'ocupació, *Multivariant* i *Bivariant* sobre les dades de *train*.

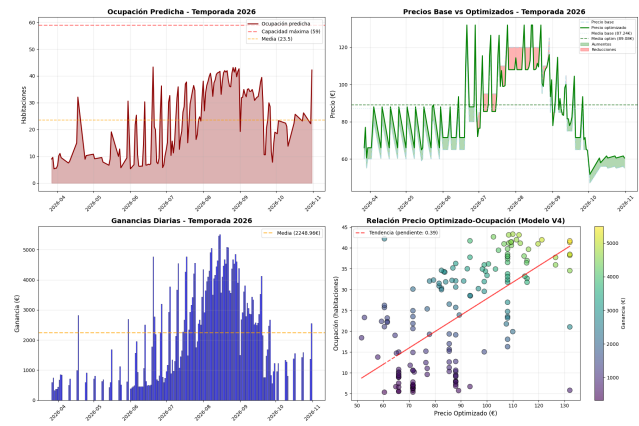


Fig. 18: Optimització de preus per a tota la temporada 2026 utilitzant el model LSTM entrenat amb dades sintètiques.

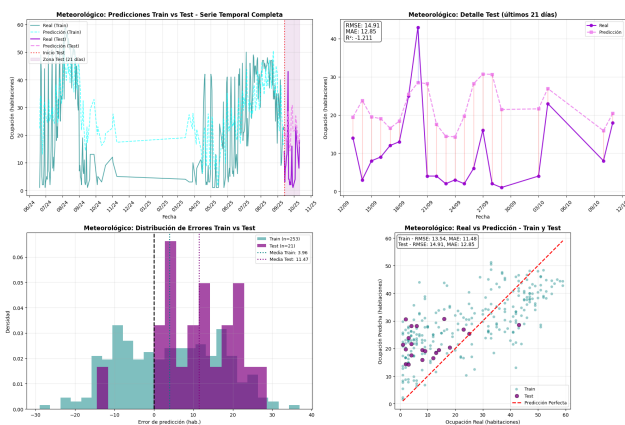


Fig. 16: Resultats de predicció del model LSTM amb dades meteorològiques afegides com a *features* addicionals, predicció de 21 dies i comparació amb les dades de *test*.

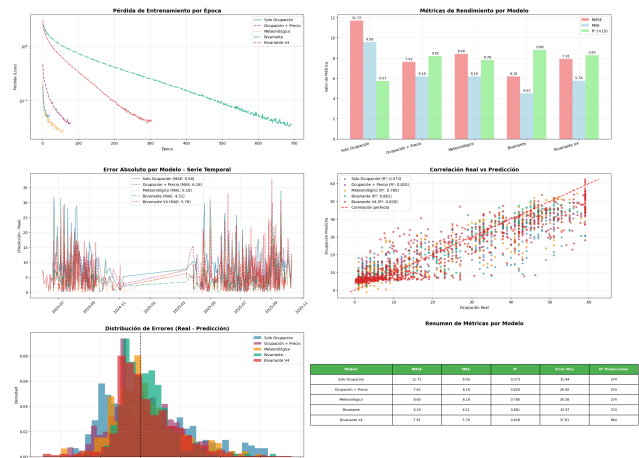


Fig. 19: Comparativa entre tots els models LSTM desenvolupats sobre les dades de *train*.

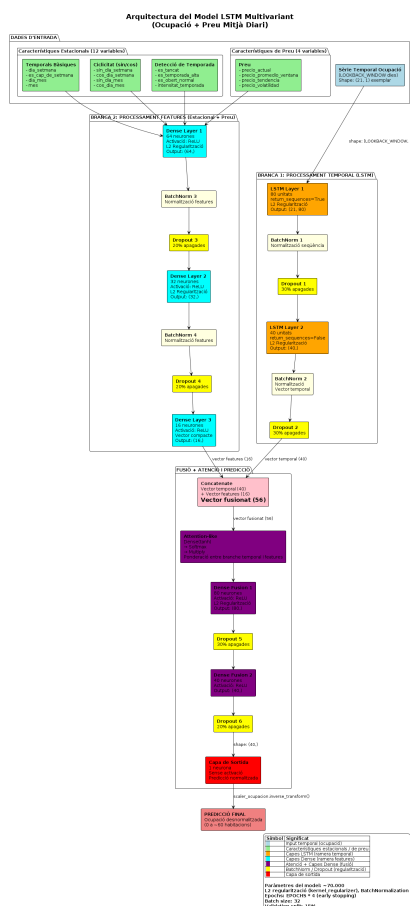


Fig. 22: Arquitectura del model LSTM (ocupació + preu).

A.7 Arquitectura models LSTM

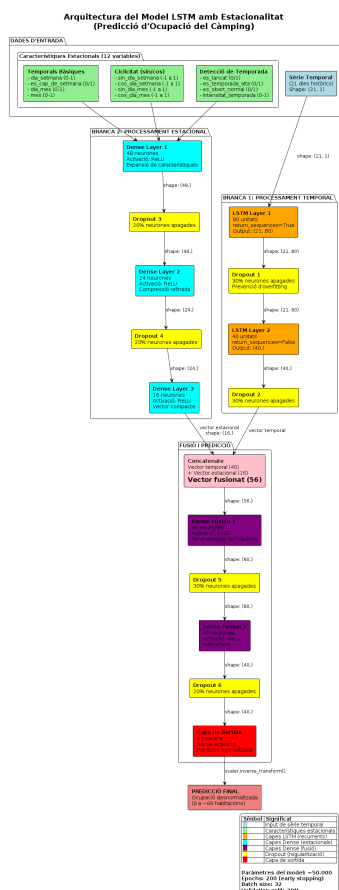
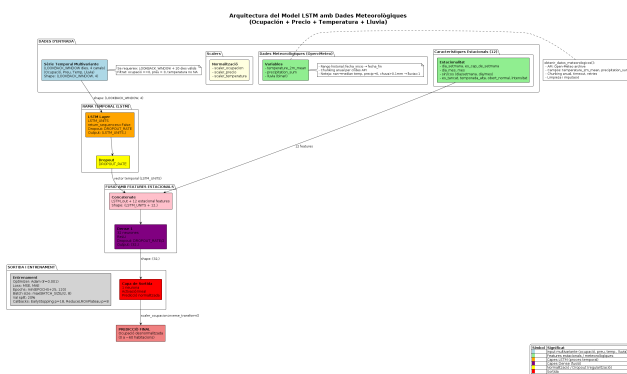
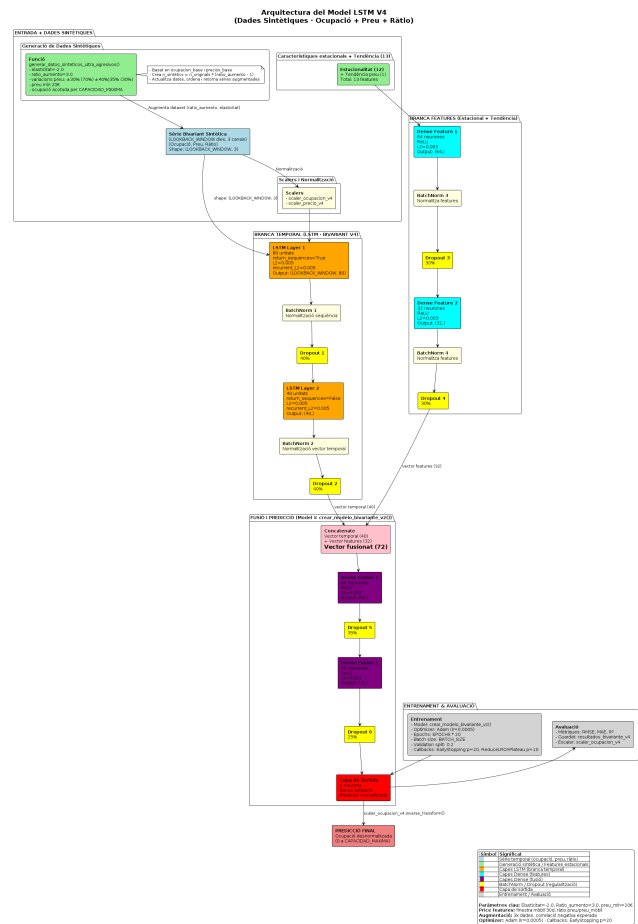
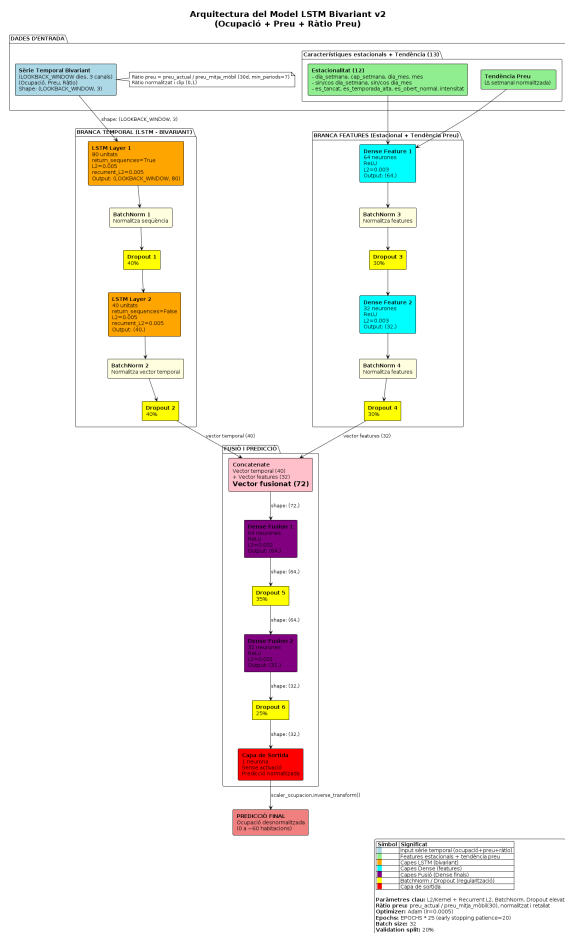


Fig. 21: Arquitectura del model LSTM (només ocupació).



A.8 Features models LSTM

TAULA 3: *Features* TEMPORALS DEL MODEL LSTM (NOMÉS OCUPACIÓ)

Variable	Tipus	Rang	Finalitat
dia_semana	Continu	0.0–1.0	Captura patró setmanal (dilluns = 0, diumenge = 1). Normalitzat perquè la xarxa aprengui millor.
es_fin_semana	Binari	0, 1	Identifica divendres–diumenge (=1). Crític: el cap de setmana té 1.2–1.5x més ocupació.
dia_mes	Continu	0.0–1.0	Posició del dia dins del mes (dia 1 = 0, dia 31 $\approx$ 1).
mes	Continu	0.0–1.0	Mes de l'any normalitzat (gener = 0, desembre = 1). Captura estacionalitat anual bàsica.
sin_dia_semana	Continu	-1.0–1.0	Sin del dia de la setmana; permet capturar la ciclicitat setmanal de forma contínua.
cos_dia_semana	Continu	-1.0–1.0	Cos del dia de la setmana; juntament amb sin representa el cicle setmanal en coordenades polars.
sin_dia_mes	Continu	-1.0–1.0	Sin del dia del mes; captura patrons mensuals cíclics.
cos_dia_mes	Continu	-1.0–1.0	Cos del dia del mes; completa la representació cíclica mensual.
es_cerrado	Binari	0, 1	Indica si el càmping està tancat aquell dia (novembre–abril). Força predicció = 0.
es_temporada_alta	Binari	0, 1	Indica temporada alta (agost principalment). Espera ocupació $\approx$ 2x respecte temporada normal.
es_abierto_normal	Binari	0, 1	Indica temporada operativa normal (maig–juny, setembre–octubre). Ocupació moderada.
intensidad_temporada	Continu	0.0–1.0	Ràtio d'ocupació del més actual respecte a la màxima històrica.

TAULA 4: *Features* RELACIONATS AMB EL PREU DEL MODEL LSTM

Variable	Tipus	Rang	Finalitat
precio_actual	Continu	0.0–1.0	Preu normalitzat del dia actual. Captura el nivell de preu en el moment de la predicció; clau per aprendre la relació preu-demanda.
precio_promedio_ventana	Continu	0.0–1.0	Mitjana de preus a la finestra (últims 21 dies). Proporciona context del nivell de preus recent i suavitza fluctuacions puntuals.
precio_tendencia	Continu	-1.0–1.0	Diferència entre preu actual i preu de fa 7 dies (preu[i] - preu[i-7]). Captura si els preus estan pujant (+) o baixant (-); indica momentum.
precio_volatilidad	Continu	0.0–1.0	Desviació estàndard de preus a la finestra (últims 21 dies). Mesura estabilitat/variabilitat de preus; alta volatilitat (>0.15) suggereix preus menys predecibles.

TAULA 5: *Features* METEOROLÒGIQUES DEL MODEL LSTM

Variable	Tipus	Rang	Finalitat
temperatura	Continu	0.0–1.0	Temperatura mitjana diària normalitzada. Captura l'efecte del clima en la demanda de càmping. Temperatures altes (+25°C) impliquen major ocupació; variable clau per a estacionalitat climàtica.
lluvia	Binari	0, 1	Indicador de precipitació diària (1 si >0.1 mm, 0 si no plou). Captura l'impacte negatiu de la pluja en la demanda a l'aire lliure; dies plujosos impliquen reducció d'ocupació.