VisaPredict AI UACJ · MIAAD

Ingeniería de características y limpieza de datos

Cómo el Visa Bulletin crudo se convierte en un objetivo entrenable, decisión por decisión y con ledger vivo

Reporte automático generado con el boletín de julio 2026 — se rehace con cada boletín nuevo.

8
decisiones de FE documentadas

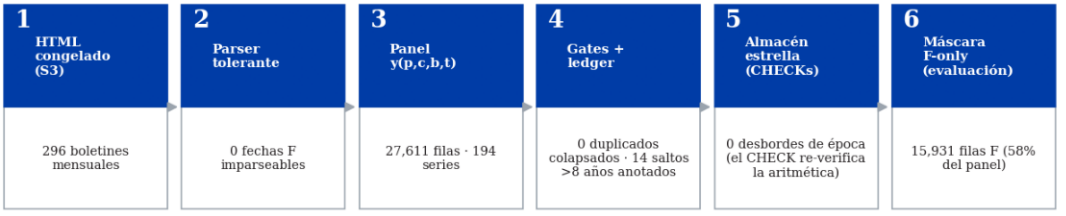
44→1
selección de características (FDR)

≤3 meses
tope de interpolación de huecos

58%
filas F entrenables

De HTML congelado a objetivo evaluable: 6 etapas, cero correcciones en silencio

Cada etapa publica su ledger: los ceros son inalcanzables por construcción (guardias que abortan el build) y lo anómalo se anota, jamás se recorta.



Fuente: U.S. Department of State, Visa Bulletin (dic 2001 – julio 2026).

Decisiones magistrales de limpieza (1 de 2)

Las 12 decisiones que blindan el dato crudo; cada una vive en código citado, no en prosa suelta.

1 Régimen C/F/U/UNK como anotación, F como único objetivo

`vp_data/visa_common.py:classify_status`

Aplanar C→fecha y U→NaN destruía el régimen administrativo. La columna status lo preserva; solo las celdas F (fecha específica) son objetivo predictivo (formulación v5.1) y la evaluación enmascara todo lo demás (B1).

2 Centinela UNK (nunca el string NA)

`vp_data/visa_common.py:classify_status`

El literal 'NA' colisiona con la coerción por defecto de pandas.read_csv (lo lee como NaN) y borraba la anotación. UNK distingue 'sin dato' de 'Unavailable' y sobrevive a cualquier consumidor downstream.

3 Pivote de siglo con guardia de época

`vp_data/visa_common.py:string_to_datetime`

Las celdas publican años de 2 dígitos ('01MAY16'); strptime pivota 69..99→19xx. Una fecha F anterior a t0=1975 haría days_since_base negativo: build_panel aborta (underflow) y el CHECK days_is_datediff del almacén re-verifica la aritmética completa.

4 Tolerancia a erratas y notas de la fuente

`vp_data/visa_common.py:string_to_datetime/classify_status`

20 años de boletines traen footnotes (C*/U*), espacios y erratas ('4rd'). El parser normaliza sin descartar el mes; lo imparseable queda UNK con su raw_value intacto (nada se corrige en silencio: la celda cruda se conserva).

5 Deduplicación por preferencia de régimen F>C>U>UNK

`pipeline/build_panel.py:main`

En transiciones de etiqueta (p.ej. EB-5 'Unreserved' 2022) una categoría canónica aparece dos veces el mismo mes. 'first' era una moneda al aire que podía tirar una observación F entrenable; se prefiere F y se ABORTA si dos F del mismo mes discrepan (conflicto de fuente que resuelve un humano).

6 Fechas imparseables abortan en la causa

`pipeline/build_panel.py:main`

Una fecha F coercionada a NaT violaría days_iff_F lejos de su causa (en el CHECK del almacén); un bulletin_date NaT viajaría hasta el merge de dim_date. Ambos abortan en build_panel con las filas culpables (AA3).

Decisiones magistrales de limpieza (2 de 2)

Las 12 decisiones que blindan el dato crudo; cada una vive en código citado, no en prosa suelta.

7 Dominios de categoría validados al leer

`pipeline/build_panel.py:load_employment/load_family`

keep_default_na=False protege el centinela UNK pero desactiva la coerción NA de todo el frame; un literal extraño en F_level/EB_level pasaría como string. El dominio se valida explícitamente tras cada read_csv (AA4).

8 Huecos: interpolar ≤3 meses; largos NaN; relleno solo para entrenar

`vp_model/preprocess.py:to_regular_monthly · vp_model/models.py:to_timeseries`

Los huecos son meses C/U (MNAR: la ausencia es señal). Corridas ≤3 meses se interpolan linealmente; las largas quedan NaN (todo-o-nada por corrida, sin rampas parciales). to_timeseries rellena los NaN residuales SOLO para dar continuidad al entrenamiento — jamás son objetivo: la evaluación puntúa únicamente fechas F reales (máscara B1, fuente única metrics._aligned).

9 Caracterización EDA imputa con Kalman, nunca rampas sin tope

`vp_model/series_characterization.py:_clean · vp_model/missingness.py:kalman_impute`

STL/espectro/catch22 exigen series completas. Los huecos largos se imputan con suavizado de Kalman (espacio de estados, imputeTS::na_kalman), no con interpolación lineal multi-año ni extrapolación de bordes: una rampa inventada fabrica tendencia y contamina Hurst/changepoints/entropía (AB1).

10 Pruebas formales sobre las F crudas (con caveat de espaciado)

`experiments/build_eda_facts.py:_formal_tests`

ADF/KPSS/DF-GLS corren sobre las observaciones F sin imputar: imputar antes de una prueba de raíz unitaria sesga hacia 'integrada'. Costo aceptado y documentado: en series con huecos el índice queda comprimido y la estructura de rezagos asume espaciado regular (AB3).

11 Retrogresiones = señal; outliers se cuentan, jamás se recortan

`vp_model/series_characterization.py:count_outliers · pipeline/mega_audit.py:d9_jumps`

Las retrogresiones y saltos >8 años son eventos administrativos reales que el modelo debe tolerar (la tesis lo argumenta). Ningún paso winsoriza ni elimina valores extremos del dato; solo se CUENTAN con estadísticos robustos (z-STL, Hampel) y las figuras anotan lo que quede fuera de rango en vez de recortarlo en silencio (AC1/AC2).

12 El contrato se re-verifica declarativamente en el almacén

`schema.sql:days_iff_F/pdate_iff_F/days_is_datediff/rank_iff_F`

Las invariantes de limpieza no viven solo en Python: los CHECK/PK/FK del esquema estrella rechazan en la carga cualquier fila que las viole, con el nombre exacto de la invariante rota.

Decisiones magistrales de FE (1 de 2)

Las 8 transformaciones que convierten fechas administrativas en regresores sin fuga.

1 Objetivo = días desde t0 (1975-01-01), solo estado F

`pipeline/build_panel.py · schema.sql:days_is_datediff`

La fecha de prioridad se convierte a un entero de días desde una época fija anterior a la prioridad más antigua observada (1979-11, Filipinas F4). Un objetivo numérico continuo, con contrato aritmético re-verificado en el almacén, en lugar de fechas crudas imposibles de regresar.

2 Rejilla mensual regular con huecos acotados

`vp_model/preprocess.py:to_regular_monthly`

Los modelos exigen índice regular; los meses C/U no son objetivo. Corridas de hueco ≤3 meses se interpolan linealmente; las largas quedan NaN (todo-o-nada por corrida) y el relleno de continuidad posterior jamás se puntúa (máscara F-only B1).

3 Árboles predicen la primera diferencia, no el nivel

`vp_model/models.py:Differenced · vp_model/preprocess.py:difference/undifference`

Un árbol no extrapola fuera del rango visto: sobre el nivel (tendencia creciente de décadas) se satura al máximo histórico. Modelar Δy mensual (estacionario) y reintegrar de forma causal (cumsum anclado al último nivel observado) resuelve la extrapolación gratis.

4 Calendario fiscal codificado cíclicamente (seno/coseno)

`vp_model/preprocess.py:calendar_features`

El año fiscal de visas arranca en octubre (las cuotas se reinician ahí). Codificar mes y posición fiscal con seno/coseno evita imponer un orden falso entre diciembre y enero — un entero 1..12 haría que el modelo viera esos meses vecinos como los más lejanos.

Decisiones magistrales de FE (2 de 2)

Las 8 transformaciones que convierten fechas administrativas en regresores sin fuga.

5 24 rezagos mensuales como memoria de los regresores

`vp_model/config.py:HYPERPARAMS['trees']/'rlinear'`

Dos años de historia por origen: cubre un ciclo fiscal completo con margen y deja grados de libertad suficientes (series evaluables ≥84 F). Constante externalizada en config, no enterrada por modelo.

6 Escalado ajustado SOLO en la ventana inicial

`vp_model/feature_builder.py:fit_scaler`

Las redes torch operan mal sobre magnitudes de ~18,000 días. El Scaler se ajusta únicamente con la ventana de entrenamiento explícita y se invierte tras predecir: ajustarlo sobre la serie completa filtraría el futuro al pasado (leakage).

7 Política de covariables explícita por familia de modelo

`vp_model/config.py:COVARIATES`

Solo los árboles diferenciados reciben calendario (la campaña canónica se derivó así); rlinear y las NN van conscientemente sin covariables. 'year' (monótona, no acotada sobre target diferenciado) fue RETIRADA en la re-campaña AQ del 4-jul-2026 (PENDIENTES #12).

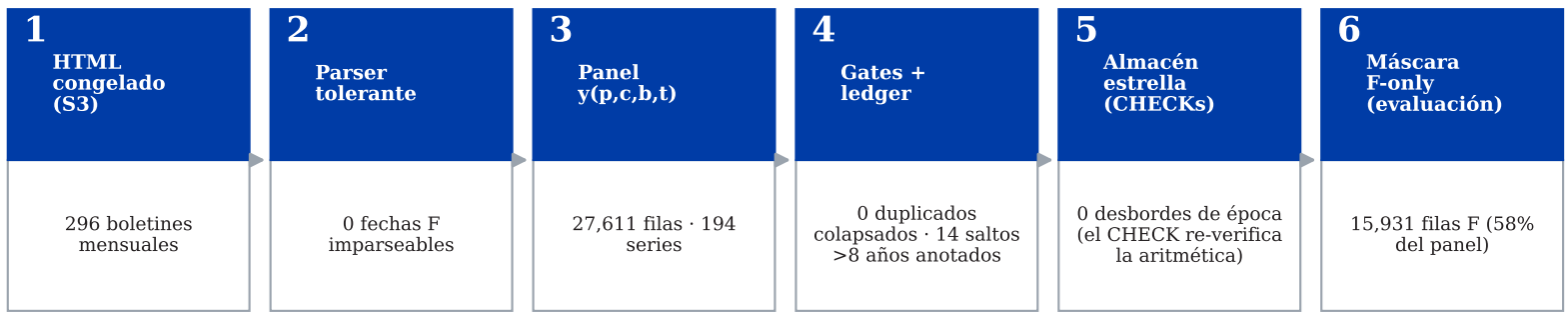
8 Selección FRESH (FDR) + des-redundancia mRMR del catálogo

`vp_model/feature_select.py:select · experiments/build_fe_facts.py`

Con series de longitud corta (decenas a cientos de observaciones) cada grado de libertad cuenta. El conjunto unido de características de caracterización (catch22 + descriptores) se filtra por relevancia con corrección Benjamini-Yekutieli y se colapsa la colinealidad conservando una representante por grupo (|Spearman|>0.9), contra la dificultad real de pronóstico de cada serie (MASE del campeón).

De HTML congelado a objetivo evaluable: 6 etapas, cero correcciones en silencio

Cada etapa publica su ledger: los ceros son inalcanzables por construcción (guardias que abortan el build) y lo anómalo se anota, jamás se recorta.



Fuente: U.S. Department of State, Visa Bulletin (dic 2001 – julio 2026).

VisaPredict AI

25 años de deriva de formato, normalizados a 4 regímenes

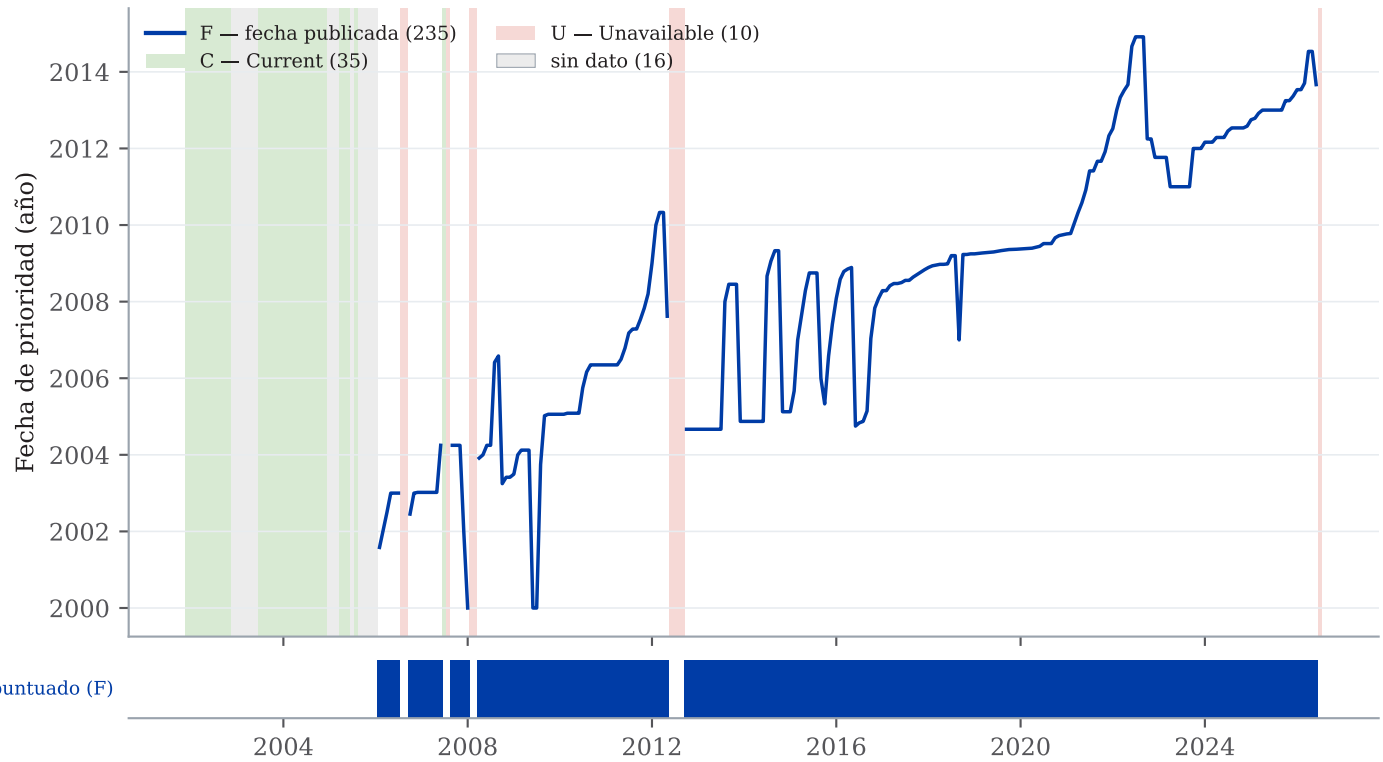
Cada fila llama al parser real (vp_data.visa_common) con la celda tal cual se publicó: nada se corrige en silencio y la celda cruda se conserva siempre en raw_value.

celda cruda	valor parseado		regla aplicada
01MAY16	F	2016-05-01	fecha exacta: strptime %d%b%y
08OCT79	F	1979-10-08	pivote de siglo: %y 69–99 → 19xx (guardia anti-futuro contra el boletín)
15JUL05*	F	2005-07-15	footnote tolerado: se extrae el token de fecha y se valida con strptime
C	C	Current	Current — sin atraso ese mes; anotación descriptiva, no objetivo
U*	U	Unavailable	footnote tolerado también en el estado (U* → U)
(vacía)	UNK	sin dato	celda vacía → centinela UNK (nunca el string NA); raw_value se preserva

Fuente: U.S. Department of State, Visa Bulletin (dic 2001 – julio 2026).

C y U son anotación, no objetivo: solo se puntúan las 235 fechas publicadas

India · EB-2 · FAD: la línea existe solo en los meses F; los fondos marcan el régimen del boletín (35 meses Current, 10 Unavailable, 16 sin dato). La franja inferior muestra los meses que la evaluación puntúa (máscara F-only).

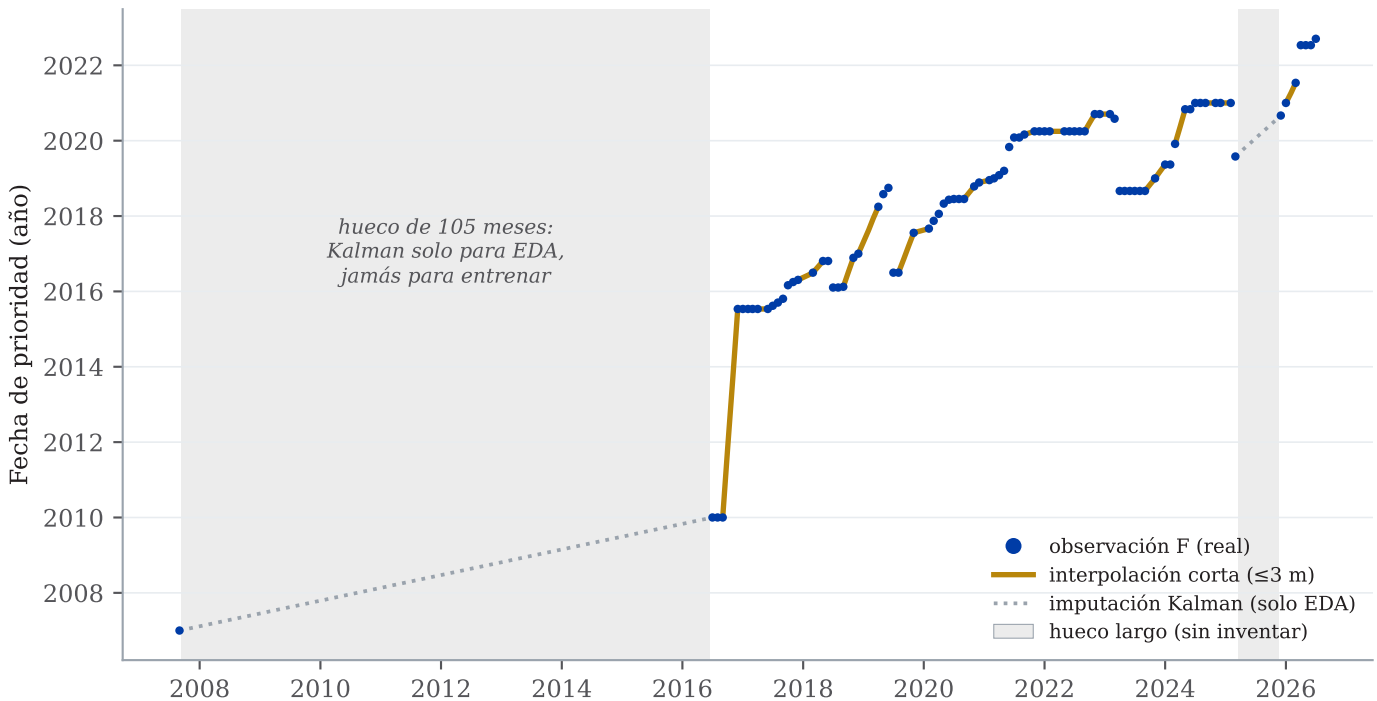


Fuente: U.S. Department of State, Visa Bulletin (dic 2001 – julio 2026).

VisaPredict AI

Huecos de hasta 3 meses se interpolan; los 2 tramos largos jamás se inventan

México · EB4_RW (trabajadores religiosos) · FAD: 140 meses sin fecha en 22 corridas (la más larga, 105 meses). Las corridas cortas (≤3) se interpolan linealmente para la rejilla de entrenamiento; las largas quedan vacías y solo el EDA las caracteriza con suavizado de Kalman.

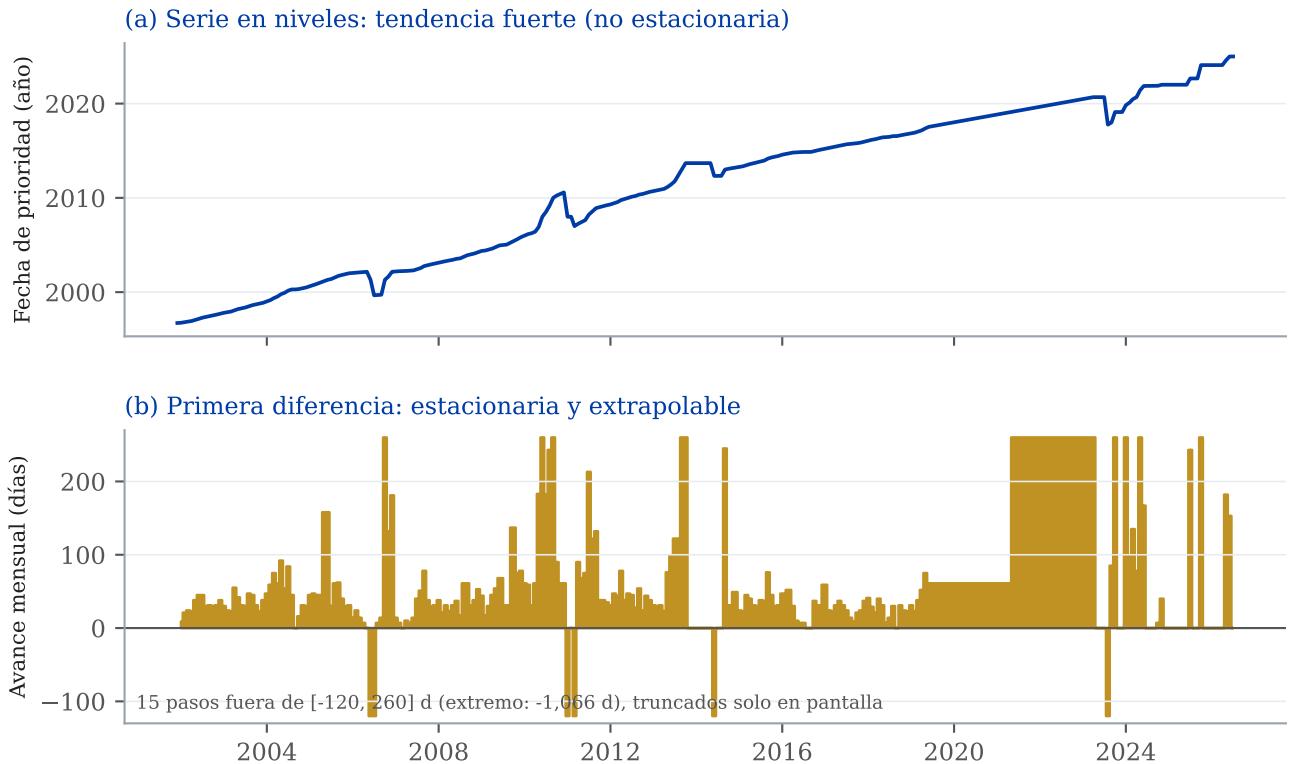


Fuente: U.S. Department of State, Visa Bulletin (dic 2001 – julio 2026).

VisaPredict AI

Los árboles no extrapolan: se les entrena sobre el avance mensual, no el nivel

Serie Resto del mundo · F2A · FAD: el nivel arrastra décadas de tendencia (un árbol se satura en el máximo visto); la primera diferencia es estacionaria y el pronóstico se reintegra de forma causal.



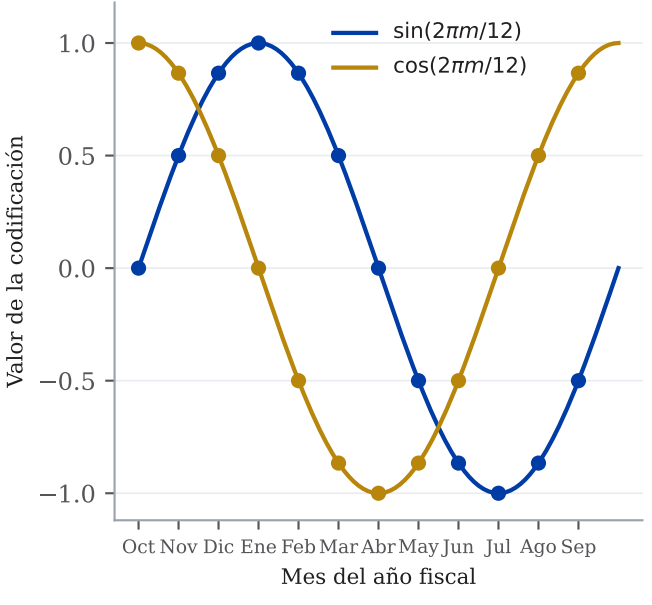
Fuente: U.S. Department of State, Visa Bulletin (dic 2001 – julio 2026).

VisaPredict AI

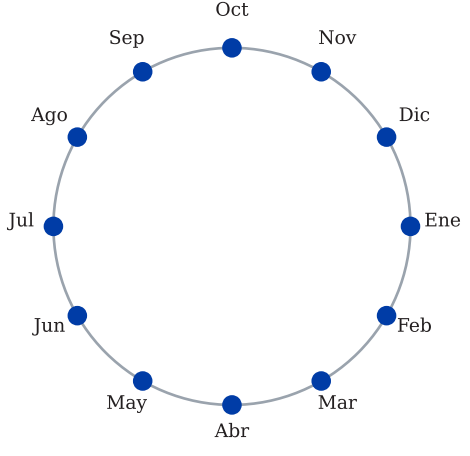
Diciembre y enero son vecinos: el calendario fiscal se codifica en círculo

Seno/coseno de la posición del mes en el año fiscal (arranca en octubre, cuando se reinician las cuotas), generados por el encoder canónico que consumen los modelos.

(a) Componentes seno/coseno



(b) Meses sobre el círculo unitario

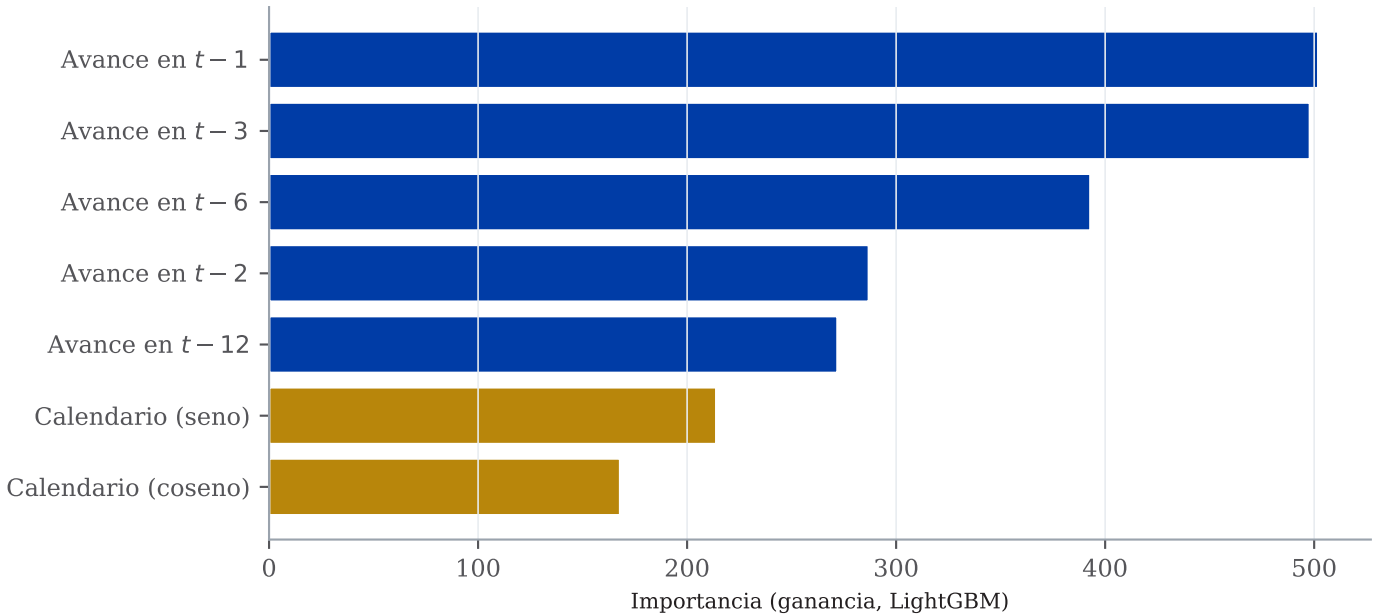


Fuente: U.S. Department of State, Visa Bulletin (dic 2001 – julio 2026).

VisaPredict AI

La característica dominante: Avance en $t-1$

Importancia por ganancia de un LightGBM entrenado sobre el avance mensual de las series familiares FAD (rezagos del avance + calendario cíclico).



Fuente: U.S. Department of State, Visa Bulletin (dic 2001 – julio 2026).

VisaPredict AI

Ledger de limpieza del corte vigente

Publicado por pipeline/build_panel.py en cada build; el reporte lo re-lee, no lo re-calcula.

corte del panel	**2026-07**
filas del panel	**27,611**
series estructurales	**194**
filas F (fecha publicada)	**15,931**
filas C (Current)	**11,058**
filas U (Unavailable)	**621**
filas UNK (sin dato)	**1**
duplicados colapsados (preferencia F>C>U>UNK)	**0**
fechas de boletín imparseables	**0**
fechas F imparseables	**0**
desbordes de época (fecha F < t0)	**0**
saltos >8 años (anotados, no recortados)	**14**

Por qué los ceros son ceros

Los ceros del ledger no son suerte: son inalcanzables por construcción. Una fecha F imparseable o un desborde de época ABORTAN el build en la causa (build_panel), y el almacén estrella re-verifica el contrato con CHECKs declarativos (days_iff_F, pdate_iff_F, days_is_datediff). Los 14 saltos >8 años sí existen: son eventos administrativos reales que el modelo debe tolerar — se anotan en la auditoría, jamás se recortan del dato.

Selección de características: 44 → 1 → 1

FRESH (relevancia con FDR Benjamini-Yekutieli, α=0.05) + des-redundancia mRMR (|Spearman|>0.9) sobre 50 series con campaña canónica.

La superviviente

```
DN_OutlierInclude_p_001_mdrmd
```

Variable objetivo de la selección

hold-out MASE del campeón por serie (elegido con sel_mase; campaign pool canónico)

Lectura honesta

Con 50 series efectivas y 44 candidatas, la corrección por FDR deja UNA sola característica de caracterización asociada de forma robusta a la dificultad de pronóstico. El catálogo completo se conserva como herramienta descriptiva del EDA; NO entra como covariable a los modelos.

Notas metodológicas y linaje

Fuente única

Todas las cifras de este reporte provienen de reports/fe/fe_facts.json (generado por experiments/build_fe_facts.py, versión de builder v1.1.0) y del panel data/processed/visa_panel_long.csv. Ninguna cifra visible está escrita a mano.

Código citado

Cada decisión referencia su módulo vivo: vp_model/feature_builder.py (FE_DECISIONS, FE_VERSION), vp_data/cleaning.py (CLEANING_DECISIONS + ledger), vp_model/preprocess.py (huecos, diferenciación, calendario) y vp_data/visa_common.py (parser). El detalle narrativo vive en docs/CLEANING.md.

Figuras

Las figuras f01–f07 se insertan como la MISMA figura viva de experiments/make_fe_figures.py en vector (cero re-rasterización); la galería PNG bilingüe (claro/oscuro) que consume el web sale del mismo script.

Reproducibilidad

make fe-all (fe-facts + fe-figures + fe-report) en el pipeline público github.com/UACJ-MIAAD/VisaPredictAI. Se regenera automáticamente con cada boletín nuevo; el gate de vintage aborta si el catálogo va detrás del panel.

Aviso

Documento académico y demostrativo (UACJ · MIAAD). No constituye asesoría migratoria ni predicción oficial.