Equipo 2 – ML + Gemini (Guía EXHAUSTIVA desde cero)

Checklist completo y recetas listas para copiar/pegar en Colab o Jupyter.

1) Datasets y columnas concretas

Dataset	Filas	Columnas (clave detectadas)
Kepler KOI	9564	koi_period, koi_duration, koi_depth, koi_kepmag, koi_steff, koi_disposition/koi_pdisp
K2	4004	pl_orbper, st_teff, sy_gaiamag/sy_vmag/sy_kmag, disposition
TESS	7703	pl_orbper, pl_trandep, st_teff, st_tmag, tfopwg_disp

Esquema unificado mínimo para entrenar:

- **period_days** (float) – Kepler: `koi_period`; K2/TESS: `pl_orbper`.
- **duration_hours** (float) – Kepler: `koi_duration`; K2/TESS: si no existe → NaN.
- **depth_ppm** (float) – Kepler: `koi_depth`; TESS: `pl_trandep`; K2: NaN si no existe.
- **teff_K** (float) – Kepler: `koi_steff`; K2/TESS: `st_teff`.
- **mag** (float) – Kepler: `koi_kepmag`; K2: `sy_gaiamag`→`sy_vmag`→`sy_kmag`; TESS: `st_tmag`.
- **label** (int) – 1 = planeta/candidato (CONFIRMED/CANDIDATE/PC/CP/KP/APC); 0 = falso positivo (FALSE POSITIVE/FP/FA/REFUTED).

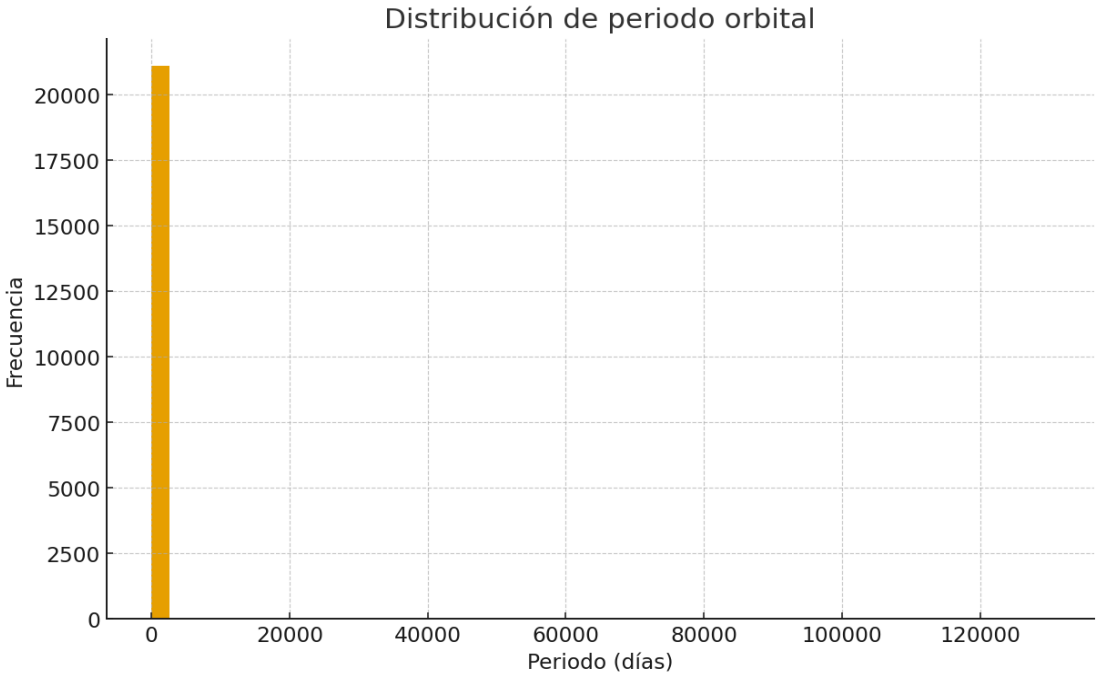
2) Limpieza de datos (exactamente qué hacer)

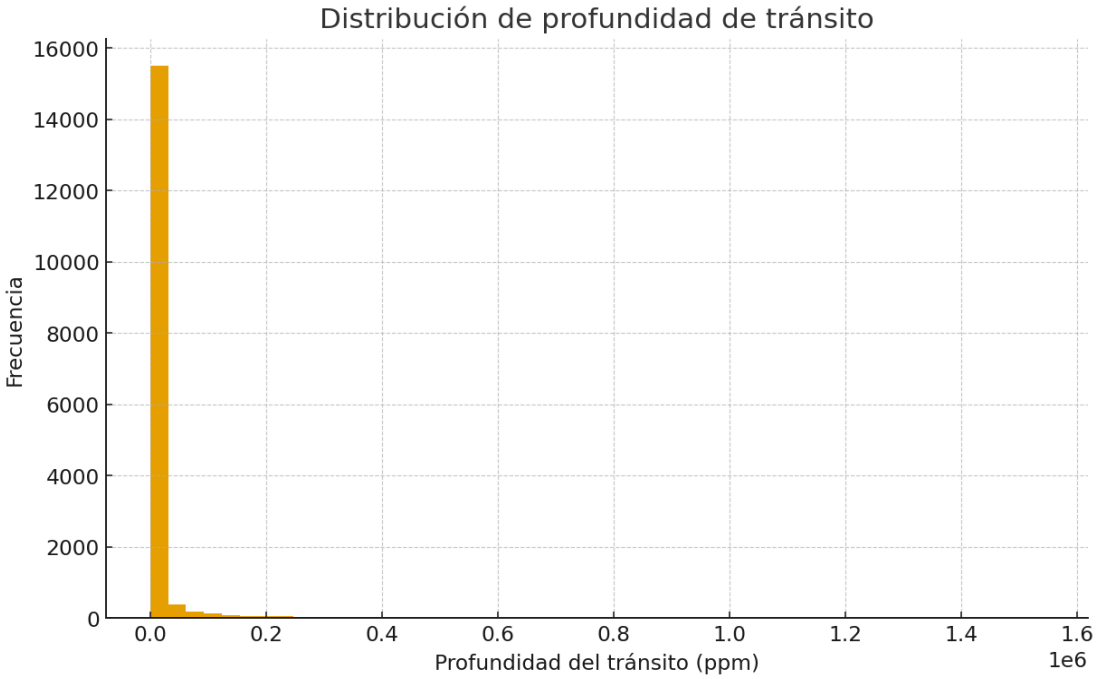
- Eliminar filas sin `period_days` o sin `label`.
- Convertir todas las columnas a numéricas con coerción (valores inválidos → NaN).
- Imputación rápida para entrenar: mediana por columna numérica (o eliminar filas si el % de NaN es bajo).
- Estandarizar (opcional) con `StandardScaler` para modelos lineales.
- Features derivadas sugeridas: `log_period = log1p(period_days)`, `log_depth = log1p(depth_ppm)` y `z_teff` (estandarizado).

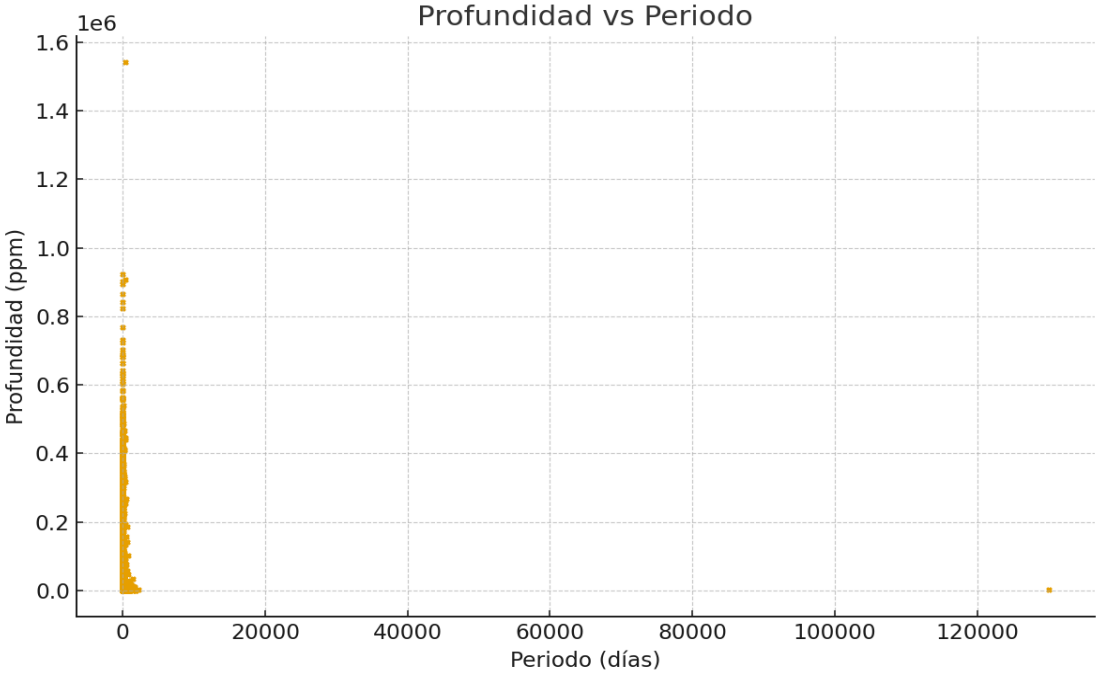
3) Fusión de los 3 catálogos (ya aplicada en esta guía)

Se unificaron Kepler, K2 y TESS con el esquema anterior y se guardó una muestra en:
/mnt/data/demo_candidates_unified.csv

4) Visualizaciones de control de calidad


Distribución de periodo orbital
Frecuencia
20000
17500
15000
12500
10000
7500
5000
2500
0
0
20000
40000
60000
80000
100000
120000
Periodo (días)


Distribución de profundidad de tránsito
Frecuencia
16000
14000
12000
10000
8000
6000
4000
2000
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1e6
Profundidad del tránsito (ppm)


Profundidad vs Periodo
1e6
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Profundidad (ppm)
0
20000
40000
60000
80000
100000
120000
Periodo (días)

5) Notebook listo para Colab (entrenamiento + exportación)

Te dejo un notebook `.ipynb` con todas las celdas: carga, limpieza, unión, entrenamiento, métricas, exportación `model.pkl`, y funciones para la App.

6) Código clave (Colab)

El notebook incluye: limpieza, unión, entrenamiento con RandomForest, exportación de `model.pkl` + `columns.json` y módulo `predict.py` listo para la App.

7) Orange – pasos exactos

- Prepara `orange_dataset.csv` con columnas: period_days, duration_hours, depth_ppm, teff_K, mag, label (0/1). Puedes usar `demo_candidates_unified.csv`.
- Abre Orange → **File** (cargar CSV) → **Select Columns** (Target: `label`; Features: numéricas; Meta: `source`/`id`).
- **Impute** (mediana) → **Normalize** (z-score) → **Test & Score** (5-fold, estratificado).
- AÑADE clasificadores: **Random Forest** (n_estimators=400, min_samples_leaf=2), **Gradient Boosting** (learning_rate=0.05, n_estimators=400).
- Conecta a **Confusion Matrix** y **ROC Analysis** para visualizar. Guarda capturas para el pitch.
- Si deseas exportar: **Save Model** → genera `.pickle`. Nota: para la App, replica hiperparámetros en scikit-learn (el pipeline de Colab ya lo deja listo).

8) Qué revisar antes de integrar con la App

- Que `ml/model.pkl` exista y cargue sin error.
- Que `ml/predict.py` responda a `predict_row` y `top_features` con un diccionario de ejemplo (prueba local).
- Que `columns.json` esté en `ml/` y tenga el orden de columnas (para evitar mismatch).
- Proveer a Equipo 1 un CSV de muestra (`demo_candidates_unified.csv`) y 3 casos ilustrativos (claro, dudoso, FP).

9) Parámetros recomendados (arranque rápido)

- RandomForestClassifier: n_estimators=400, max_depth=None, min_samples_leaf=2, class_weight=None.
- GradientBoostingClassifier: n_estimators=400, learning_rate=0.05, max_depth=3, subsample=0.9.
- Métrica foco: **recall** (no perder candidatos) y F1. Reporta matriz de confusión.

10) Plan de contingencia (si algo falla)

- Si el entrenamiento tarda: reduce a 200 árboles o usa solo Kepler+TESS.
- Si faltan columnas en K2: entrena con period/teff/mag y deja `duration/depth` como NaN (imputación).
- Si la App no puede cargar el modelo: usa el mock de `predict.py` (devuelve probabilidades razonables).

Notebook de Colab listo:
/mnt/data/ML_Exoplanetas_24h_Colab.ipynb