
Instituto Tecnológico y de Estudios Superiores de Monterrey

Inteligencia Artificial Avanzada para la Ciencia de Datos

Preparación de datos

Los concentrados v2

Daniel Queijeiro Albo - A01710441

Diego Alfaro Pinto - A01709971

Diego Isaac Fuentes Juvera - A01705506

Jesus Ramirez Delgado - A01274723

Mauricio Anguiano Juarez - A01703337

Luis Adrián Uribe Cruz - A01783129

# Adaptaciones de CRISP-DM

CRISP-DM plantea una preparación compuesta por seleccionar, limpiar, construir, integrar y formatear datos, nuestro proyecto requirió ajustar esa secuencia para reflejar un proceso iterativo y la naturaleza dual del dataset (v1.0 por vaca y v2.0 por sesión).

1. Select data  Selección de datos.

La actividad original se mantuvo, pero:

- La decisión de inclusión o exclusión se registró explícitamente por archivo (datasets incluidos, excluidos y derivados).
- Se agregó un paso previo no formalizado en CRISP-DM: fusión de los 33 archivos individuales de vacas, generando un dataset que dio origen a ambas versiones (v1.0 y v2.0).

2. Clean data  Limpieza de datos.

Se preservó el propósito original, pero:

- La limpieza e imputación fue documentada como un proceso propio.
- La imputación se aplicó antes de ramificar los datasets, algo no contemplado explícitamente por CRISP-DM, pero necesario para asegurar coherencia entre versiones.

3. Construct data  Construccion de datos.

La construcción de atributos derivó en dos niveles, ajustando el enfoque original:

- **A nivel sesión** (producción total, flujos, conductividad, etiqueta de inquietud).
- **A nivel vaca** (resúmenes, LOWESS, tasa de decaimiento, agregaciones de comportamiento).

Esto obligó a que la actividad existieran dos veces primero en sesiones y luego al consolidar por vaca.

4. Integrate data  Integración de datos

Se mantuvo la intención original, pero:

- La integración clave se realizó **antes** de la fase formal (al fusionar CSV por vaca).
- Luego se documentó nuevamente al unir el resumen por vaca con el ranking experto (v1.0) y al proyectar la tabla única hacia datasets especializados (v2.0).

5. Format data  Formateo de datos

La actividad se conservó, pero:

- Se documentó  la generación explícita de datasets.

- El formateo se entendió como “dataset final para consumo del modelo”, no solo reestructuración tabular..

En esencia se mantuvieron las actividades para documentar el proceso de preparación de datos de cada dataset y se tradujo al español las actividades.

## 1.0 Descripción de los datos

De nuestra fuente de datos tenemos dos grupos, datos alfanuméricos ordenados proporcionados en formato “.csv” e imágenes proporcionadas por el sistema de DeLaval. El desglose de los datos son los siguientes:

**Nota:** Se tienen en total 33 registros de vacas.

| Datos alfanuméricos (.csv) | | | | | |
|---|---|---|---|---|---|
| Archivo | Descripción | Instancias (suma total) | Atributos (columnas) | Tamaño (suma total) | Almacenamiento de origen |
| idVaca.csv | Registro detallado de sesiones de ordeño de una vaca específica, con hora, duración, producción de leche, flujos por pezón, conductividad, presencia de sangre, patadas e indicadores de calidad del ordeño y destino de la leche. | 7,272 | 35 | 1.08 MB | Carpeta compartida de OneDrive propiedad del Instituto Tecnológico y de Estudios Superiores de Monterrey. |
| inventario_total_180725.csv | Registro del inventario ganadero con datos productivos, reproductivos y de alimentación de cada vaca del establo. | 33 | 41 | 0.01MB | |
| patadas_180725.csv | Registro por animal de sesiones de ordeño: fecha/hora y DEL, intervalos/éxito de ordeño, conteos por extremidad (DI/DD/TI/TD) asociados a “patadas”, e indicadores de calidad de leche (OCC/RCS). | 37 | 43 | 0.01 MB | |
| reporte_180725 | Registro por animal con métricas de actividad y reproducción: grupo, pasos por puerta y MDI, ventana de ordeño, estado reproductivo (preñada/abierta/fresca), días en ordeño y días desde eventos clave (celo, inseminación, parto, tratamientos). | 22 | 43 | 0.01 MB | |

| Imágenes | | | | |
|---|---|---|---|---|
| Numero de | Formato | Resolución | Tamaño total | Fuente de origen |

| | | | | |
|---|---|---|---|---|
| imagenes | | | | |
| 57870 | .jpg | 1920×1080 | 60.1 GB | Carpeta compartida de OneDrive propiedad del Instituto Tecnológico y de Estudios Superiores de Monterrey. |

**2.0 Selección de datos**

De todos los datos que recibimos sólo nos resultan útiles algunos, por lo que hacemos una selección de cuales datos incluir y excluir de nuestro dataset final. Adicionalmente, manipulamos el dataset para adecuarlo a nuestras necesidades, por ejemplo, combinar columnas de cada cuarto de la vaca para obtener el total.

Dado nuestro objetivo el equipo decidió descartar el uso de las imágenes proporcionadas y enfocarnos en el uso de los datos alfanuméricos.

Es importante mencionar que se hizo un trabajo previo tanto para la fase previa de exploracion de datos como para esta misma, se integraron los 33 registros individuales de cada vaca en uno solo, el archivo que alberga estos datos y fue la base para todas las preparaciones de datos es ***registros_sesiones_merged.csv***

## 2.1 Datos incluidos

- Dataset 3 – Patadas (`patadas_180725.csv`)
    - 37 observaciones, 43 atributos.
    - Aporta información fina de comportamiento dentro del robot (número de patadas por cuarto, duración de extracción, fallos por pezón, puntaje AMD, intervalos entre ordeños, éxito del ordeño).

- Dataset 2 – Reporte (`reporte_180725.csv`)
    - 34 observaciones, 22 atributos.
    - Resume el estado productivo y sanitario de las vacas (grupo VMS, estado reproductivo, días en ordeño / preñez, pasos por puerta, MDI, producción acumulada, días desde eventos reproductivos y tratamientos).

- Registros de sesiones por vaca
    - 33 archivos CSV individuales (uno por vaca) con las sesiones de ordeño.
    - A partir de ellos se construye el dataset maestro `registros_sesiones_merged.csv` (7,239 sesiones, 34 atributos; marzo–julio 2025).

## 2.2 Datos excluidos

- Inventario total: descartado porque su estructura y fechas no permiten alinearlo de manera consistente con las sesiones de ordeño.
- Columna “Misc | Razón de la desviación” en los registros de sesiones: >99 % de valores faltantes, sin utilidad práctica para el modelado.
- Imágenes de vacas: requieren un pipeline de visión por computadora distinto al alcance actual; además presentan problemas de calidad (desalineación de cámaras, iluminación variable, obstrucciones).

# 3.0 Preparación de datos

Se requirieron dos fechas en total para el proyecto, que se describirán a continuación su preparación.

- **Versión 1.0:** Dataset original usado en los primeros modelos de random forest e isolation forest, estos fueron descartados en su totalidad por el enfoque erróneo y sus características limitadas, con muy pocas instancias y mayor cantidad de categorías.

- **Versión 2.0:** Este dataset aprovecha en su totalidad todas las instancias y disminuye los atributos por dataset para cada modelo en particular, de comportamiento y sanidad, aprovechando solo esas variables que significan un valor útil para el modelo. Este se implementó en todos los modelos a excepción de los previamente mencionados.

## 3.1 Limpieza e imputación a nivel sesión (común a v1.0 y v2.0)

A partir de los 33 CSV por vaca se construyó *registros_sesiones_merged.csv.* Antes de pasar a cada versión de dataset, se aplicó una estrategia de limpieza e imputación sobre las **sesiones de ordeño**:

1. **Separación por tipo de registro**
    - Se separaron filas con Acción = "Ordeño" (a imputar) de otras acciones como “Rechazada”, que se conservan sin cambios.
2. **Ordenamiento temporal por vaca**
    - Para las filas de “Ordeño” se ordenó por ID Vaca y fecha/hora de inicio, permitiendo usar ventanas temporales coherentes por animal.
3. **Columnas a imputar**

- Flujos medios y máximos por cuarto, producciones por cuarto, conductividad por cuarto y sangre por cuarto.

4. **Tratamiento de ceros**

    - En flujos, producciones y conductividad, los ceros en registros de “Ordeño” se interpretaron como **huecos de medición** y se reemplazaron por NaN para ser candidatos a imputación.

5. **Imputación con ventana ±3 días por vaca**

    - Para cada NaN se tomó una ventana de ±3 días para la misma vaca y se rellenó con el promedio local cuando había datos suficientes.

6. **Promedio por vaca y promedio global**

    - Si aún quedaban NaN, primero se usó el promedio de la vaca; si seguían faltantes, el promedio global de la columna.

7. **Imputación de sangre (ppm)**

    - Para sangre en leche, los valores faltantes se rellenaron con 0, interpretando la ausencia de registro como ausencia de detección.

8. **Reconstrucción del dataset final**

    - Se volvieron a unir las filas imputadas de “Ordeño” con las filas de otras acciones, restaurando el orden original de las observaciones.

Con esto se obtuvo un *df_final* con el mismo número de registros que el original, pero con muchos menos valores faltantes en variables clave de producción, flujos, conductividad y sangre, que luego alimenta tanto v1.0 como v2.0.

### 3.2 Construcción de atributos derivados a nivel sesión

Sobre *registros_sesiones_merged.csv* (ya imputado) se generaron múltiples **atributos derivados** que luego se usan en los distintos datasets:

- **Totales por cuarto** (flujos medios, flujos máximos, producción, conductividad, sangre) calculados como suma DI + DD + TI + TD.

- **Fecha y hora separadas** (+ versión en segundos) para analizar patrones diarios y duraciones.

- **Producción suavizada (LOWESS)** y **tasa de decaimiento** de la curva de lactancia, para capturar tendencias y estabilidad productiva en el tiempo.

- **Identificador de vaca** extraído por regex del nombre de archivo, clave para unir patadas, reporte y sesiones.

- **Conteo de patadas** y métricas derivadas (patadas por día, por ordeño y por hora en máquina).

- **Conteo de pezones no encontrados** por cuarto y total.

- **Etiqueta de inquietud (label_inquieta)**, construida combinando flags de patada, ordeño incompleto, pezones no encontrados y ordeños con duración anómalamente larga (percentil 95).

Estos atributos son la base de los **indicadores por vaca** (v1.0) y de los **datasets de sesiones para comportamiento y sanidad** (v2.0).

## 3.3 Data Preparation v1.0 – Dataset de resumen y ranking por vaca

Script principal:
load_dataframe_vacas, fill_missing_data, generate_summary, save_dataframe, join con ranking_vacas_df_final.

### 3.3.1 Select Data (Selección de datos)

- **Fuente base**: *registros_sesiones_merged.csv* (7,239 sesiones imputadas y enriquecidas con atributos derivados).

- **Fuentes complementarias**:

    - Dataset de **Patadas** y métricas de comportamiento agregadas.
    - Dataset de **Reporte** con información productiva y reproductiva.
    - ranking_vacas_df_final.csv con el Puntaje_final asignado por expertos.

- Se decide trabajar a **nivel vaca**, construyendo un dataset compacto de ~30–40 vacas con:

    - indicadores agregados de producción, salud y comportamiento,
    - más el ranking experto para análisis exploratorio y modelos v1.0.

### 3.3.2 Clean Data (Limpieza de datos)

Se carga el dataset maestro:

```
df = load_dataframe_vacas('datos/registros_sesiones_merged.csv')
```

- 

Se aplica:

```
df_filled = fill_missing_data(df)
```

- Esta función encapsula la estrategia de imputación descrita en la sección 2.1, garantizando que no haya huecos críticos para el cálculo de promedios/totales por vaca.

### 3.3.3 Construct Data (Construcción de datos)

A partir de df_filled se genera un resumen por vaca:

```
df_summary = generate_summary(df_filled)
```

- En este paso se consolidan, por vaca:
    - Producción total y promedio,
    - Totales y promedios de flujos, conductividad y sangre,
    - Indicadores derivados de la curva de producción (LOWESS, tasa de decaimiento),
    - Métricas de comportamiento (patadas por día/ordeño/hora, pezones no encontrados),
    - Otros agregados relevantes.
- Resultado: df_summary, un dataset con **pocas instancias y muchas variables agregadas**, base de los primeros modelos v1.0.

### 3.3.4 Integrate Data (Integración de datos)

Se carga el ranking de expertos:

```
df_ranking = load_dataframe_ranking(
   'datos/ranking_vacas_df_final.csv'
)[["ID Vaca","Puntaje_final"]]
```

Se alinean índices y se realiza el join:

```
df_ranking_indexed = df_ranking.set_index(('Ranking','ID Vaca'))
df_final = df_summary.set_index(('ID','ID Vaca')).join(
   df_ranking_indexed,
   how='left'
).reset_index()
```

- df_final contiene ahora, por vaca:
    - sus indicadores agregados de producción, comportamiento y sanidad,
    - y el PuntajeFinal proveniente de la evaluación experta.

### 3.3.5 Format Data (Formateo de datos)

- Se asegura una estructura de columnas clara (MultiIndex para secciones como ID, Ranking, etc.) y tipos consistentes.

El dataset se persiste para análisis y modelado:

```
save_dataframe(df_summary, 'datos/resumen_vacas.csv')
```

- Uso: este dataset se utilizó para los **modelos v1.0** (Random Forest e Isolation Forest a nivel vaca) y para análisis descriptivo inicial; su principal limitante fue el **bajo número de instancias**.

## 3.4 Data Preparation v2.0 – Datasets de sesiones para comportamiento y sanidad

Script principal: data/etl.py
Genera sessions_behavior.csv y sessions_health.csv a partir de registros_sesiones_merged.csv).

### 3.4.1 Select Data (Selección de datos)

Fuente única de entrada:

```
CSV_INPUT = "data/registros_sesiones_merged.csv"
df = load_csv(CSV_INPUT, ...)
```

- Este CSV es el mismo dataset imputado y enriquecido descrito antes, que concentra las sesiones de las 33 vacas.
- Decisión clave de v2.0: mantener todas las sesiones disponibles (miles de filas) para explotar mejor la variabilidad temporal y construir modelos más robustos para comportamiento y sanidad.

### 3.4.2 Clean Data (Limpieza de datos)

Se normalizan los nombres de columnas:

```
df = df.rename(columns={c: normalize_column(c) for c in df.columns})
```

- Eliminación de BOM y acentos,
- Conversión a minúsculas,
- Reemplazo de espacios y símbolos por guiones bajos.

Se crea la duración en segundos:

```
df["dur_seconds"] = df["main_duracion_mm_ss"].apply(parse_duration_mm_ss)
```

- La mayor parte de los valores faltantes ya fue tratada en la fase común de imputación; aquí la limpieza se centra en asegurar nombres y tipos consistentes para el modelado.

### 3.4.3 Construct Data (Construcción de datos)

Etiqueta de comportamiento (label_inquieta):

```
patada_flag     = df["estado_patada"].notna().astype(int)
incompleto_flag = df["estado_incompleto"].notna().astype(int)
pezones_flag    = df["estado_pezones_no_encontrados"].notna().astype(int)

dur_threshold   = df["dur_seconds"].quantile(0.95)
dur_larga_flag  = (df["dur_seconds"] > dur_threshold).astype(int)

df["label_inquieta"] = (
   patada_flag
   + incompleto_flag
   + pezones_flag
   + dur_larga_flag
) > 0
df["label_inquieta"] = df["label_inquieta"].astype(int)
```

- Esta variable resume en una sola etiqueta binaria todos los eventos de mala interacción con el robot.

- **Selección de atributos de comportamiento** (behavior_feature_cols):
  duración, producción total, número de ordeño, flujos medios y máximos por cuarto, producciones por cuarto.

- **Selección de atributos de sanidad** (health_feature_cols):
  sangre ppm, conductividad, flujos y producciones por cuarto.

### 3.4.4 Integrate Data (Integración de datos)

- La integración principal ya ocurrió al crear *registros_sesiones_merged.csv* a partir de los 33 CSV por vaca; en v2.0 esta tabla integrada se "proyecta" en dos enfoques especializados:
    - una para **comportamiento** (features + label_inquieta),
    - otra para **sanidad** (solo features de salud).

- Ambas vistas comparten la misma estructura de sesiones y se apoyan en los atributos derivados descritos en la sección 3.2.

### 3.4.5 Format Data (Formateo de datos)

Dataset de comportamiento:

```
df_behavior = df[behavior_feature_cols + ["label_inquieta"]].copy()
save_csv(df_behavior, CSV_BEHAVIOR, ...)
```

- Output: data/sessions_behavior.csv, listo para modelos supervisados (Random Forest v2.0/v2.1, MLP).

Dataset de sanidad:

```
df_health = df[health_feature_cols].copy()
save_csv(df_health, CSV_HEALTH, ...)
```

- Output: data/sessions_health.csv, listo para modelos no supervisados (DBSCAN v1.0, Isolation Forest v2.0/v2.1).

- Ambos datasets:
    - Tienen columnas normalizadas y tipos numéricos consistentes.
    - Contienen solo las variables relevantes para cada modelo.
    - Constituyen la base de los modelos finales v2.1 y del IMR.