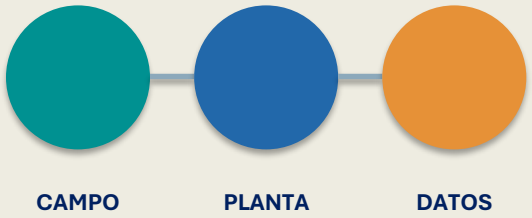
ARQUITECTURAS, AGENTES Y MODELOS

Ingeniería de Datos y Machine Learning en la Agroindustria





Pipelines escalables con PySpark, modelos predictivos y despliegue en entornos reales

Caso práctico: predicción de mermas por turno

Arturo Mark Chura Noriega
II Congreso Académico IA-LATAM 2026



SOBRE MÍ

Arturo Mark Chura Noriega

Docente · Universidad Privada del Norte· Perú

Especialista senior en Business Intelligence, Ingeniería de Datos y Analítica Avanzada, con amplia trayectoria liderando arquitecturas de datos, proyectos de Inteligencia Artificial y MLOps en los sectores de agroindustria, minería, logística y retail en Latinoamérica. Máster por la Universidad Politécnica de Valéncia (España) y especialista en pipelines distribuidos con PySpark, modelado predictivo y soluciones en la nube

arturo.chura@upn.edu.pe · https://www.linkedin.com/in/arturo-chura-tech/

La agroindustria ya genera datos; el reto es convertirlos en acción

Múltiples sistemas describen una misma operación, pero no siempre conversan entre sí.

CAMPO

Clima, lote, variedad, proveedor y cosecha

RECEPCIÓN

Peso, calidad, Brix, temperatura y humedad

PROCESO

Línea, turno, paradas, velocidad y rendimiento

NEGOCIO

Costos, cumplimiento, exportación y margen

El desafío no es acumular más información: es disponer de señales confiables a tiempo para decidir.

El problema: conocemos la merma cuando ya ocurrió

La supervisión recibe el consolidado al cierre del turno; la oportunidad de corregir se reduce.

HOY

La merma se consolida al cierre.
La investigación es reactiva.
Las causas se analizan tarde.

RIESGO

Materia prima variable.
Sensores con interrupciones.
Paradas y condiciones cambiantes.

NECESIDAD

Estimar el riesgo del siguiente turno.
Priorizar inspecciones.
Actuar antes de procesar.

Pregunta guía: ¿podemos anticipar una merma elevada con la información disponible antes del siguiente turno?

Una desviación que merece atención

Impacto operativo: +7.0 pts de merma equivalen a 840 kg perdidos en un solo turno (~$2,100 USD/turno).

Indicador	Escenario
Materia prima recibida	12,000 kg
Producto aprovechable	10,200 kg
Merma observada	15.0%
Referencia histórica	8.0%
Brecha	+7.0 pts (840 kg perdidos / turno)

DECISIÓN ACTUAL

¿Qué ocurrió?
¿En qué etapa?
¿Se pudo anticipar?

Objetivo del proyecto: estimar la merma del siguiente turno y entregar señales explicables al jefe de planta.

Una cadena de valor: del dato a la intervención

Cada componente resuelve una parte del problema operativo.

01

Capturar

Sensores · ERP · calidad

02

Confiar

Validación · limpieza · trazabilidad

03

Predecir

Variables · modelo · validación

04

Actuar

Alerta · explicación · seguimiento

El modelo no reemplaza al jefe de planta: mejora su anticipación y priorización.

Arquitectura Medallion: separar recepción, confianza y consumo

Diseñada para mantener trazabilidad y preparar datos reutilizables.

BRONZE

Datos originales con trazabilidad

CSV · sensores · ERP

SILVER

Datos depurados y homologados

Tipos · nulos · duplicados

GOLD

Datos listos para analítica y ML

Agregados por línea/turno

Resultado: una tabla Gold por línea y turno, con variables disponibles antes de la predicción.

Los datos reales llegan incompletos, tarde y con variaciones

El pipeline debe manejar imperfecciones sin inventar información.

Hora	Línea	Temperatura °C	Brix
08:00	L01	24.5	12.4
09:00	L01	—	12.6
10:00	L01	—	12.5
11:00	L01	27.1	12.8

CONTROLES

- Duplicados y claves
- Tipos y rangos válidos
- Esquema versionado
- Interpolación acotada
- Registro de calidad

Regla de gobierno: si la brecha supera el límite definido, marcar el dato como no confiable; no rellenar a ciegas.

PySpark distribuye el trabajo, pero el diseño sigue importando

Particionado, ventanas y control del sesgo de distribución evitan cuellos de botella.

WINDOW FUNCTIONS

Orden temporal por línea.
LAG / LEAD y agregaciones.
Deduplicación reproducible.

PARTICIONADO

Organizar por fecha.
Evitar archivos diminutos.
Leer solo períodos necesarios.

DATA SKEW

Detectar claves dominantes.
Revisar distribución.
Aplicar salting o estrategias adecuadas.

La meta no es usar un clúster más grande; es procesar lo necesario con una arquitectura eficiente.

Convertimos señales operativas en variables útiles

El modelo necesita contexto temporal, no solo una fotografía del turno.

Variable	Lectura de negocio
Brix actual	Calidad de materia prima disponible
Merma lag 1	Resultado del turno anterior
Media móvil previa	Comportamiento reciente
Paradas acumuladas	Exposición a interrupciones
Temperatura media	Condición del proceso

CONTROL CLAVE

Cada variable debe existir al momento de predecir.

No usar datos del turno futuro: evitar data leakage.

La estacionalidad y el calendario deben considerarse en el diseño de las variables.

Validar en el tiempo, como ocurrirá en producción

El pasado entrena; el futuro evalúa. Nunca mezclar aleatoriamente turnos.



MAE

Error absoluto promedio

RMSE

Penaliza errores grandes

R^2

Variabilidad explicada

Validación rigurosa: LightGBM se evalúa contra el Baseline de Persistencia reduciendo el MAE en >35%.

Una predicción útil debe explicar dónde mirar y por qué

Tablero operativo de alertas para el cierre del turno actual (Inferencia batch en PySpark).

MERMA ESTIMADA

14.8%

Riesgo elevado

Referencia: 8.0%

SEÑALES

Temperatura
Paradas acumuladas
Calidad de materia prima

ACCIÓN SUGERIDA

Revisar parámetros y calidad.
Validar sensores.
Priorizar inspección antes del turno.

Explicabilidad accionable (SHAP): desglosa cuántos puntos de merma aporta cada variable en el turno.

Del experimento a un proceso repetible y gobernado

Un modelo productivo necesita versiones, responsables, métricas y trazabilidad.

ENTRENAR

Código + datos

REGISTRAR

MLflow tracking

VERSIONAR

Model Registry

INFERIR

Batch por turno

MONITOREAR

Calidad + error

GOBERNANZA

Propietario · versión · aprobación · trazabilidad

COSTO CLOUD

Escalar por demanda · programar jobs · apagar recursos