

# ¡Ya casi empezamos!



Centro de  
e-Learning



Clase 9

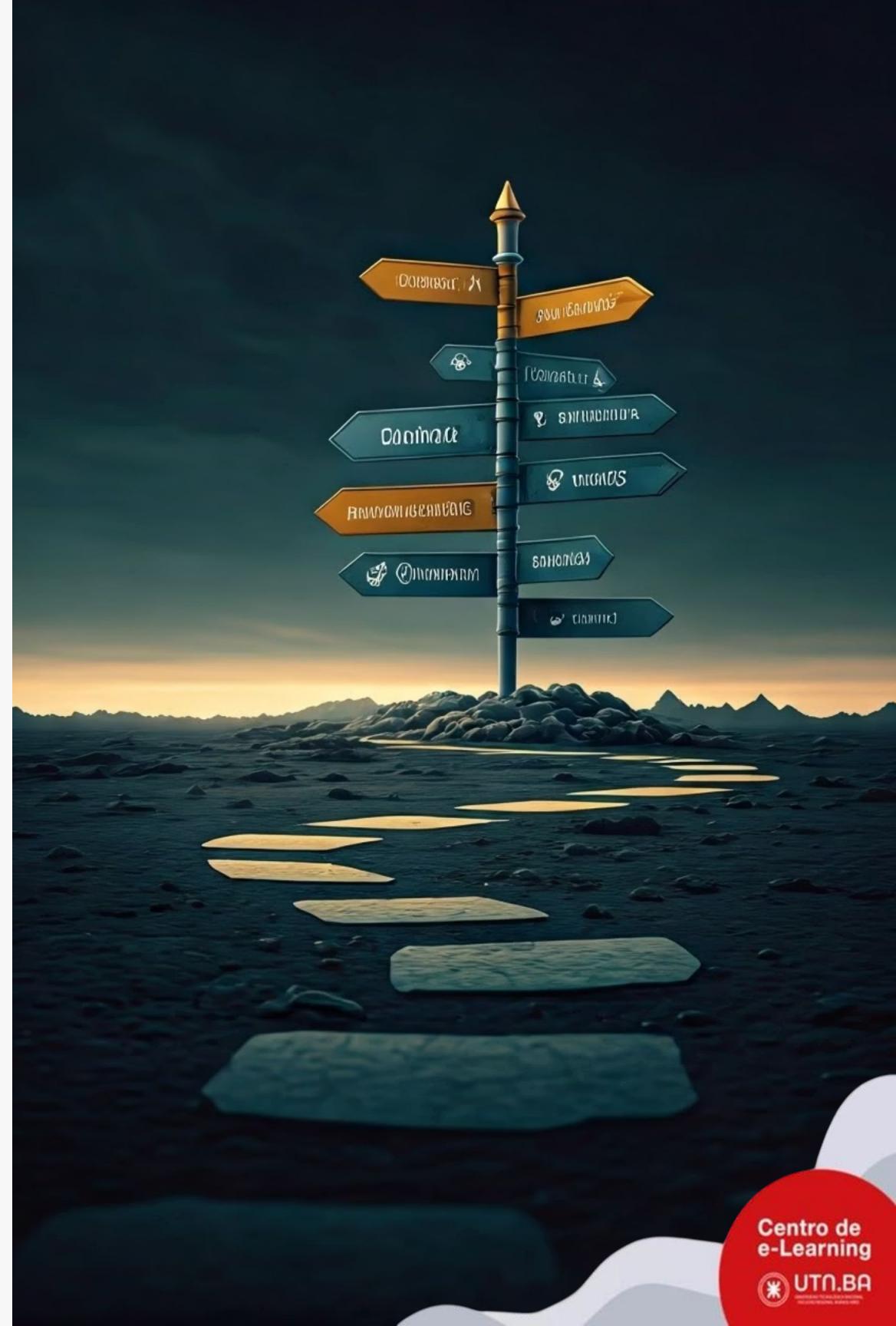
# Automatización de tareas II



# Hoja de Ruta



1. Introducción y objetivos
2. Diseño de flujos de disparadores, ¿qué son los triggers?
3. Validación inicial de datos - Semáforo
4. Demo y Ejercicio - Trigger + Validación de datos
5. Acciones con IA - La IA decidiendo y enriqueciendo
6. Monitoreo y gestión de errores
7. Estrategias de Reintentos y Notificaciones
8. Demo y Ejercicio - Monitoreo de errores y notificación
9. Cierre, preguntas y próxima clase



# 1. Objetivos de la Clase



**Entender cómo se disparan las automatizaciones**



**Entender las buenas prácticas implicadas en la gestión de errores**



**Revisar casos reales de triggers y gestión de errores**



**Reflexionar sobre cómo llevar esto a nuestros procesos**

## 2. Diseño de flujos de disparadores

### ¿Qué son los Triggers?

Los **triggers** son el punto de inicio de cualquier flujo de automatización. Actúan como sensores que detectan cuándo debe ejecutarse un proceso, transformando eventos del mundo real en acciones automatizadas.

Comprender los diferentes tipos de triggers y cuándo usarlos es fundamental para diseñar flujos eficientes y confiables. Cada tipo tiene sus casos de uso específicos y características técnicas que debes conocer.



# Tipos Principales



## Webhook

Recibe datos en tiempo real desde aplicaciones externas. Ideal para integraciones entre sistemas que necesitan comunicación instantánea.



## Cron (Programado/Scheduled)

Ejecuta tareas a intervalos específicos (cada hora, diario, semanal). Perfecto para procesos batch y reportes periódicos.



## Nuevo registro en plataformas

Se activa cuando se agrega una nueva fila en una hoja de cálculo por ejemplo. Útil para procesar formularios y datos tabulares.



## Formulario Web

Dispara el flujo cuando un usuario envía un formulario. Común en captura de leads y solicitudes de servicio.

# 3. El Semáforo de Entrada: Validación Inicial

Antes de procesar cualquier dato, implementa un **semáforo de entrada** que valide campos mínimos requeridos. Este paso crítico previene errores costosos downstream.

## Define campos obligatorios

Establece qué información es imprescindible para continuar: email, nombre, ID de cliente, etc.

## Valida formato y tipo

Verifica que los datos tengan el formato correcto antes de procesarlos (email válido, número de teléfono, fecha).

## Detén el flujo si falta algo

Si la validación falla, el flujo no debe continuar. Envía el caso a una cola de revisión manual.



# 4. Demo

## Sumar un trigger + validación de datos

Caso de ejemplo:

- Necesito que el proceso corra cada 5 minutos (Trigger tipo Cron)
- Que procese solo los archivos tipo pdf (filtrado por tipo de archivo encontrado)



# 4. Ejercicio

## Sumar un trigger + validación de datos en tu escenario

Pasos:

1. En un escenario armado, configurar un trigger tipo Cron (correr cada 5 minutos).
2. Sumar un filtro relevante para validar que lleguen los datos necesarios para el proceso
3. Activar el escenario
4. Aguardar a que ejecute



# 4. Ejercicio (final)

**IMPORTANTE:**

**Desactivar el escenario**



# 5. Acciones con IA:

## La IA como Enriquecedor y Decisor

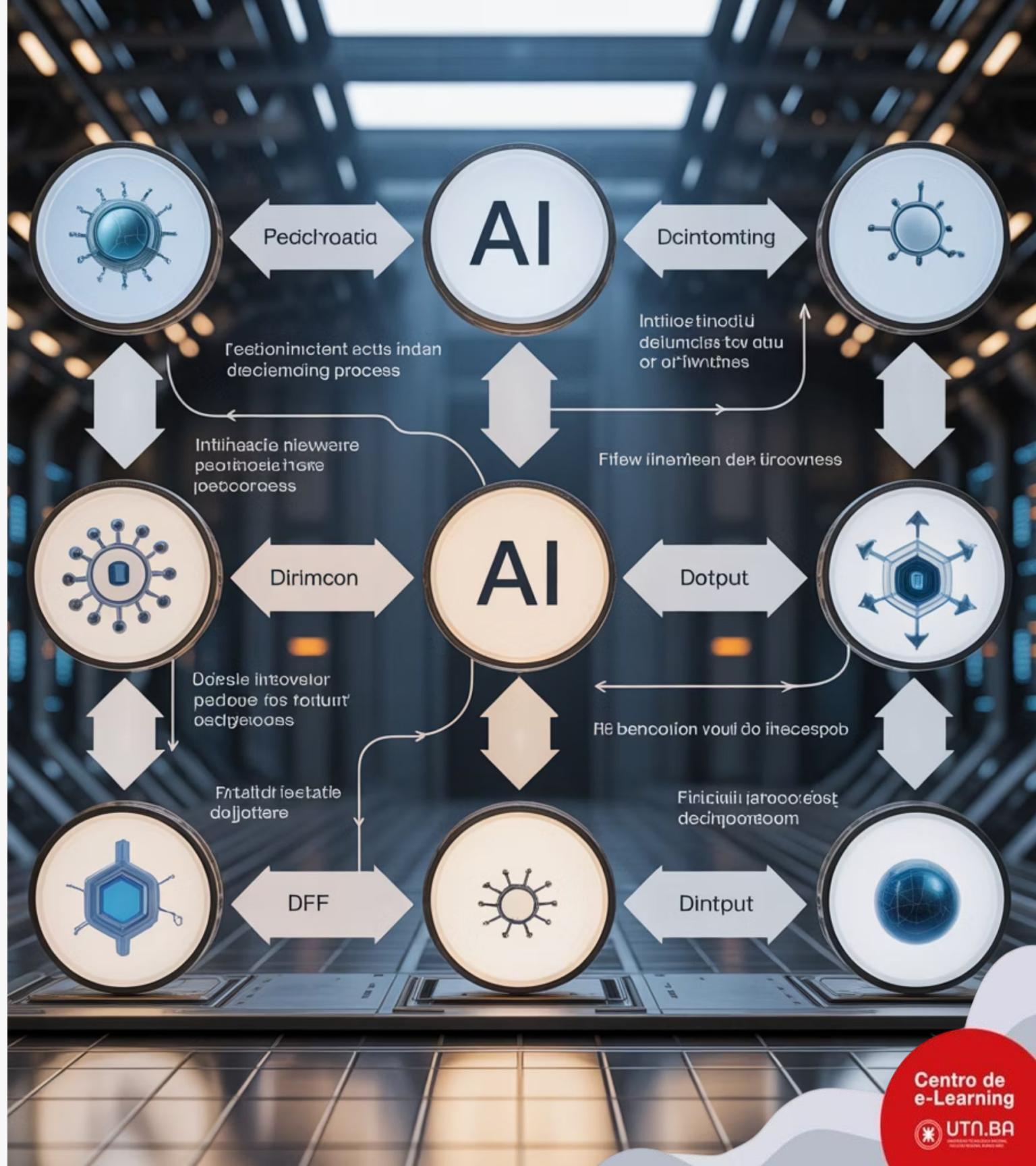
### Roles de la IA en el Flujo

Los pasos de IA no solo procesan datos, sino que toman decisiones inteligentes que determinan el camino del flujo. Son el cerebro de tu automatización.

**Clasificar:** Categoriza tickets, emails o documentos

**Extraer:** Obtiene información estructurada de texto libre

**Rutear:** Decide el siguiente paso según el contenido



# Ejemplo: Flujo de Clasificación Automática



## Email entrante

Nuevo mensaje llega al sistema

## Validación

Semáforo verifica campos mínimos



## IA clasifica

Determina categoría: ventas, soporte, facturación

## Ruteo

Envía al equipo correcto automáticamente

# 6. Monitoreo y Gestión de Errores

Todo flujo de producción enfrentará problemas. La diferencia entre un sistema robusto y uno frágil está en cómo detectas, manejás y resuelves estos errores. No es cuestión de *si* ocurrirán problemas, sino *cuándo*.

Una estrategia sólida de gestión de errores te permite mantener la operación funcionando mientras identificas y corriges issues sistemáticamente.



# Dos Categorías de Problemas

## 1. Errores Técnicos

Problemas de infraestructura y conectividad que escapan al control del negocio:

- Cortes de servicio y timeouts
- Códigos HTTP 4xx/5xx
- Límites de cuota API excedidos
- Datos con formato técnicamente inválido

## 2. Errores de Negocio

Problemas relacionados con la lógica y datos del negocio:

- Campos requeridos faltantes
- Clasificación que retorna vacía
- Email sin formato válido
- Datos fuera de rango esperado

# 7. Estrategia de Reintentos



El reinicio simple resuelve la mayoría de errores transitorios sin intervención manual, mejorando la resiliencia del sistema.

# Trazabilidad: El Mapa de Cada Ejecución para entender errores

Mantén un registro detallado de cada paso para debugging y auditoría. La trazabilidad completa es esencial en producción.

1

**ID de Ejecución**

2

**ID de Escenario**

3

**Entrada Original (Input)**

4

**Tipo de Error**

5

**Mensaje del Error**

6

**Detalle del Error (si se existe)**

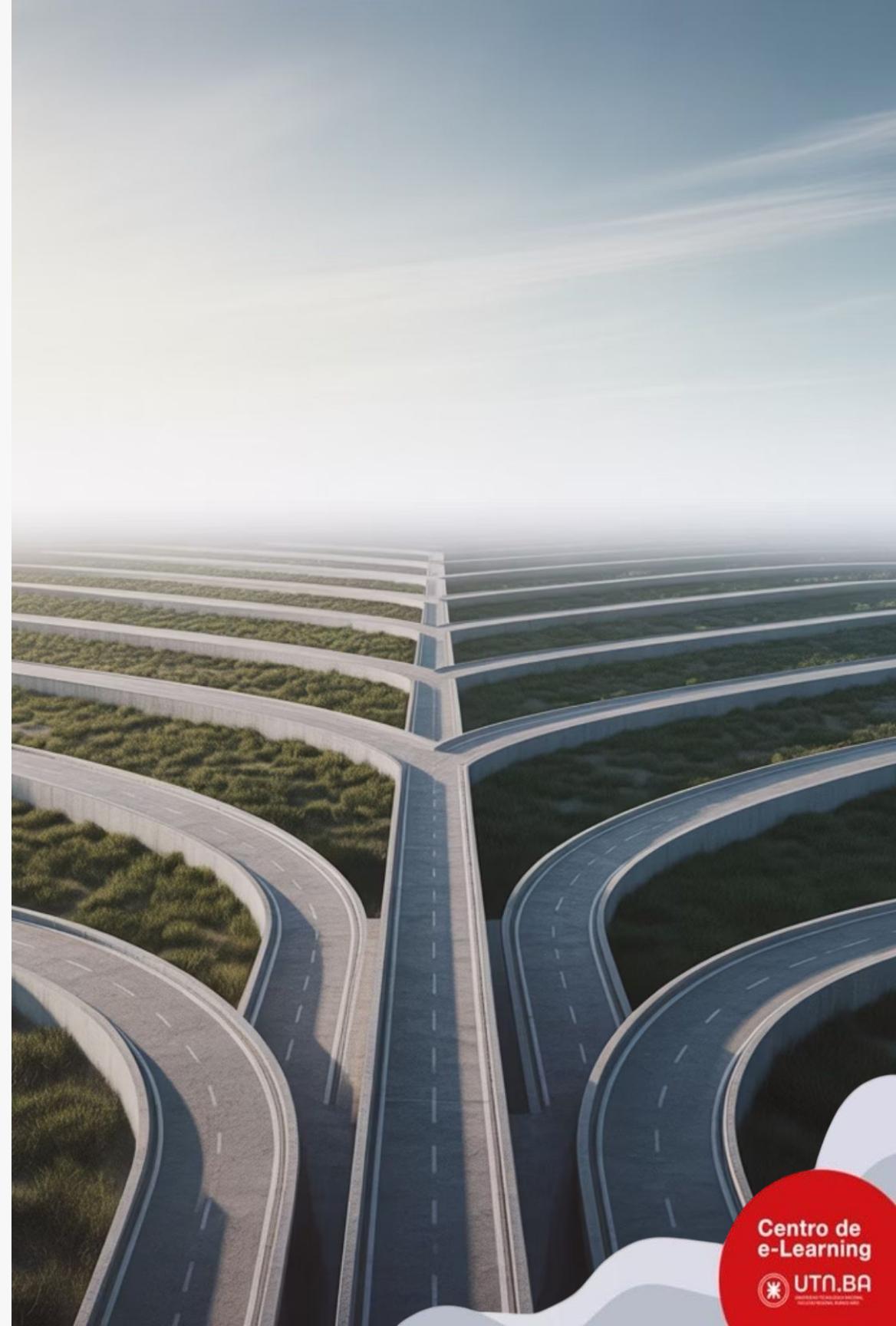
Estos campos permiten reconstruir exactamente qué pasó en cualquier ejecución, facilitando la resolución de problemas y mejora continua.

# Separación con Router: OK vs Error

Implementa un **router** que separe explícitamente los casos exitosos de los errores de negocio. Esto permite manejarlos de forma diferenciada.

Los errores técnicos deben ir directo a reintentos, mientras que los errores de negocio (como entrada incompleta) necesitan intervención humana o lógica diferente.

Esta separación clara simplifica el debugging y mejora la visibilidad operacional del sistema.



# 8. Demo

## Sumar gestión de errores en el escenario

Caso de ejemplo:

- Necesito que el NO “rompa” (finalice con error) cuando no se pueda enviar un correo
- Que me llegue un correo a mi casilla, avisando del error y entregando detalles del mismo.



# 8. Ejercicio

## Sumar un Manejo de Error en un nodo del escenario

Pasos:

1. En un escenario armado, configurar un “Error handler” a uno de los nodos.
2. Que el tipo, mensaje y detalle del error se envíe por correo.



# ¿Preguntas?

## Debatamos un poco los temas de hoy.



**¡Nos vemos en la próxima clase!**

Seguiremos con...

**Creación de herramientas internas y paneles I**