**Hugo Córdoba Leal**

**Sprint 4. Diseño y Estrategias en Producción para Soluciones de IA**

### **Diseña estrategias de puesta en producción**

**• Cuestión 1.** Selecciona la estrategia de puesta en producción, que consideres más adecuada para la solución, y diseña el diagrama de flujo que represente cómo se llevaría a cabo el despliegue.

**• Cuestión 2.**  Diseña un posible ‘feedback loop’ que permita recolectar la información del usuario y llevar a cabo una evaluación para mejorar la solución actual. Recuerda incluir la información que se recolecta, la frecuencia de recolección y cómo se usaría esta información para mejorar la solución.

**→ Propuesta de solución a las cuestiones.**

Para la puesta en producción y el mantenimiento continuo de una solución basada en modelos de machine learning como la que estamos discutiendo, una buena estrategia es el despliegue continuo con un robusto ciclo de feedback del usuario. Vamos a detallar ambas partes:

1. **Estrategia de puesta en producción: Despliegue Continuo (CI/CD)**

Diagrama de flujo del proceso de despliegue

- **Desarrollo**: Los desarrolladores escriben código en su entorno local y lo suben al repositorio central tras pruebas básicas.

- **Integración Continua (CI)**:

- Pruebas Automatizadas: Se ejecutan pruebas de unidad, integración y aceptación para verificar que los cambios no rompan funcionalidades existentes.

- Construcción Automática: Se crea el paquete o contenedor de la aplicación.

- **Entrega Continua (CD)**:

- Pruebas de pre-producción: El paquete se despliega en un entorno de staging donde se simulan operaciones reales.

- Revisión Manual/Aprobación: Una vez que se verifica que todo funciona bien en staging, se solicita una aprobación manual para pasar a producción.

- **Despliegue en Producción**:

- Despliegue Automatizado: Utilizando herramientas como Jenkins, GitLab CI/CD, o similares, el software se despliega automáticamente en el entorno de producción.

- Monitorización: El sistema es monitoreado constantemente para detectar problemas de rendimiento o errores en tiempo real.

- **Retroalimentación y Mantenimiento**:

- Se recogen datos de logs y feedback del usuario para iterar y mejorar la aplicación.

**2. ‘Feedback Loop’ para la Mejora Continua**

Diagrama de flujo del ciclo de feedback

- **Recolección de Datos**:

- Feedback del Usuario: Encuestas y formularios de feedback integrados en la aplicación para recoger la satisfacción del usuario y comentarios específicos.

- Datos de Uso: Análisis de cómo los usuarios interactúan con la aplicación, qué características usan más, y dónde encuentran dificultades.

- Errores y Fallos: Logs de errores que los usuarios encuentran y reportan automáticamente.

- **Análisis de Datos**:

- Análisis periódico de los datos recogidos para identificar áreas de mejora, problemas comunes, y la efectividad de las características recientemente implementadas.

- **Planificación de Mejoras**:

- Basado en el análisis, se planifican mejoras para futuras iteraciones del producto, que pueden incluir correcciones de errores, mejoras de usabilidad o nuevas características.

- **Implementación de Mejoras**:

- Las mejoras identificadas se priorizan y se implementan en el ciclo de desarrollo siguiente.

- **Retroalimentación del Ciclo**:

- Una vez implementadas las mejoras, se monitorizan los efectos de los cambios para ver si han resuelto los problemas sin introducir nuevos.