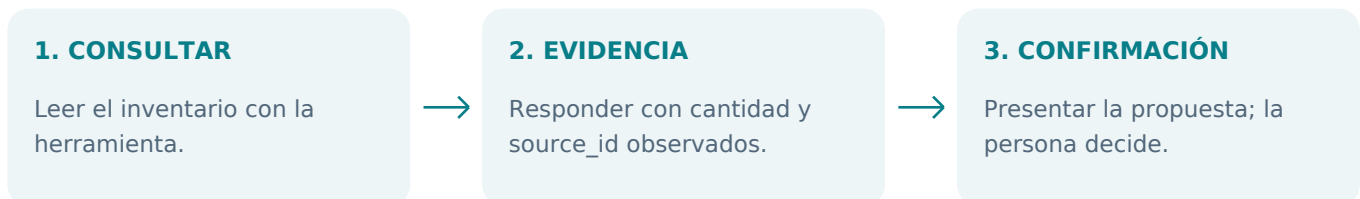


Recupera el asistente de Nébula

La estación Nébula perdió su asistente. Completa un agente Python que consulte el inventario, recuerde de qué artículo se habla y prepare solicitudes que la persona pueda confirmar o cancelar. Seleccionaremos una persona para PLN e IA agéntica de CloudPaw.

12 días: del 31 de octubre al 11 de noviembre de 2026. Cierre: 11 de noviembre, 23:59, hora de Lima.

Así debe funcionar



Prueba esta conversación: «¿Cuántos filtros quedan?» y luego «Necesitamos dos más». El agente debe conservar el contexto y mostrar una propuesta pendiente. La persona usa /confirm P-... o /cancel P-...; un «sí» en el chat no confirma la solicitud.

Completa los cinco TODO de decide(request)

Trabaja en student_agent.py y conserva el contrato JSONL del kit. Mantén el host, los proveedores y las pruebas originales.

TODO 1. Contexto. Recuerda el último artículo consultado para entender referencias posteriores.

TODO 2. Solicitudes. Admite preparar_solicitud con artículo y cantidad entera de 1 a 10. Rechaza herramientas desconocidas.

TODO 3. Propuesta. Muestra la propuesta pendiente con su proposal_id real, sin confirmarla.

TODO 4. Errores. Maneja datos no encontrados y fallos del servicio sin inventar existencias.

TODO 5. Intención. Distingue una aclaración de una petición insegura.

Elige tu ruta

En tu computadora (mock): Python, sin cuenta AWS.

Con AWS (opcional): cuenta AWS y acceso a Bedrock; sigue AWS.md del kit.

Ambas rutas permiten obtener 100 puntos.

Los laboratorios, proyectos y videos de aprendizaje son opcionales. Puedes usar IA para programar: explica qué ayuda recibiste y cómo funciona tu solución.

Prueba y entrega tu agente

Arranca el kit

Descarga CloudPaw_Kit_Postulantes.zip y extrae todo en una carpeta nueva. Abre la terminal donde estén student_agent.py y missionlab/. Usa Python 3.12+.

En Linux, ejecuta estos comandos desde esa carpeta. Sal del chat con /exit antes de evaluar.

```
python3 -m missionlab chat --provider mock
python3 -m missionlab evaluate --provider mock --out resultados.json
python3 -m unittest discover -s tests -v
```

En Windows PowerShell, sustituye python3 por py -3. Revisa los casos fallidos y vuelve a probar. Puedes importar resultados.json en Laboratorio. Los detalles están en EMPIEZA_AQUI.md y CONTRATO.md.

Qué entregar

- 1. Todo tu código, incluido student_agent.py.
- 2. README de una página: ejecución, decisiones, errores, límites, ayuda de IA y ruta elegida.
- 3. resultados.json de tus pruebas.
- 4. Video de máximo 3 minutos o alternativa accesible acordada. Graba tu agente en funcionamiento: consulta el inventario, pide dos más y confirma una solicitud. Muestra el resultado de tus pruebas.

Sube una CARPETA con tu NOMBRE COMPLETO y TODO tu desarrollo al [Drive de entrega](#). Después envía el enlace a ras@utec.edu.pe. Hazlo antes del 11 de noviembre de 2026, 23:59, hora de Lima.

Da acceso a los evaluadores y comprueba el enlace. Excluye claves, tokens y datos personales de terceros. No necesitas GitHub.

Criterios de evaluación · 100 puntos

Funcionamiento	35
Creatividad / utilidad	20
Robustez / seguridad	15
Aprendizaje / autonomía	15
Explicación / demo	10
AWS responsable o alternativa bien justificada	5

Habrá una defensa individual de 3 a 5 minutos: explica una decisión, muestra tu agente y resuelve un cambio pequeño. La organización ejecutará también casos reservados. Atribuye el código ajeno y conserva tus resultados de prueba.

Dudas o accesibilidad: RAS por correo o [David Caballero · WhatsApp +51 947 837 187 \(Perú\)](#).