

# Desafío Salud – Coach de Bienestar Preventivo con IA Híbrida (NHANES)

1° Hackathon de Inteligencia Artificial Aplicada Duoc UC 2025

Duoc UC - Pontificia Universidad Católica de Chile

6-7 de noviembre de 2025

## Desafío Específico: Salud

### Coach de Bienestar Preventivo con IA Híbrida (NHANES)

---

#### Información del Evento

Detalle	Información
Institución	Duoc UC - Pontificia Universidad Católica de Chile
Evento	1° Hackathon de Inteligencia Artificial Aplicada Duoc UC 2025
Modalidad	Presencial
Duración	27 horas
Fechas	6-7 de noviembre de 2025
Versión del documento	2025-10-14

**Nota importante:** Este documento es un anexo específico del desafío **Salud** y se alinea completamente con las *Bases Oficiales* del Hackathon de IA de Duoc UC 2025.

---

#### Resumen Ejecutivo

**Objetivo:** Desarrollar un **Coach de Bienestar Preventivo con IA Híbrida** que estime riesgo cardiométrabólico usando datos NHANES y genere recomendaciones personalizadas con guardrails éticos.

**Duración:** 27 horas (6-7 noviembre 2025)

**Modalidad:** Presencial en Duoc UC

**Entregables clave:** - API FastAPI funcional (`/predict`, `/coach`) - App demo deployada en Hugging Face Spaces - Modelo ML con validación temporal y métricas de fairness - Sistema RAG con base de conocimiento local - Reporte técnico y plan PDF descargable

**Puntuación total:** 100 puntos distribuidos en 5 categorías principales

---

## 1) Propósito e Impacto

### Objetivo Principal

Construir un **asistente de bienestar preventivo** que estime **riesgo cardiometaobólico** (alto riesgo de diabetes o hipertensión) usando datos tipo **NHANES** y que genere **recomendaciones personalizadas** en lenguaje natural, con **guardrails** para un uso **ético y responsable**.

### Impacto Esperado

- **Innovación en salud preventiva:** Desarrollo de soluciones IA híbridas para promoción de salud
- **Educación y conciencia:** Fomento de hábitos saludables a través de tecnología
- **Escalabilidad:** Prototipo **productizable** (API + App) que pueda implementarse en contextos reales
- **Formación profesional:** Experiencia práctica en IA aplicada a salud para estudiantes de Duoc UC

## 2) Problema a Resolver

### Descripción del Reto

Dado un perfil individual (edad, sexo, antropometría, hábitos de sueño, actividad física, tabaquismo, dieta), el sistema debe:

#### Funcionalidades Requeridas:

1. **Estimación de riesgo**
  - Calcular un puntaje de riesgo (0–1) de evento cardiometaobólico
  - Enfoque en **alto riesgo de diabetes o hipertensión arterial (HTA)**
2. **Explicabilidad**
  - Identificar y explicar las principales variables que impulsan el riesgo para cada persona
  - Proporcionar insights claros y comprensibles
3. **Plan de acción personalizado**
  - Sugerir un plan de 2 semanas con acciones **SMART** (específicas, medibles, alcanzables, relevantes, temporales)
  - Usar **LLM** reforzado con **mini-base de conocimiento local (RAG)**
  - Incluir **disclaimer** explícito: *no es diagnóstico médico*

#### Restricción Crítica (Anti-fuga de datos):

Si la etiqueta (*label*) usa analitos de laboratorio (ej: **A1c** o **glucosa**), **NO** pueden usarse esos mismos campos como *features*. Esta regla **anti-fuga** es **obligatoria** y será estrictamente evaluada.

## 3) Datos (NHANES) y alcance

### Descripción de los datos:

- **Fuente:** NHANES (National Health and Nutrition Examination Survey) - EE. UU., NCHS/CDC
- **Data pack:** El Data Team proveerá CSV con **diccionarios de variables** y **codebooks** de referencia

### Componentes de datos disponibles:

- **Demographics:** Edad, sexo, grupo étnico

- **Examination:** Presión arterial (PA), antropometría (peso, altura, cintura)
- **Dietary:** Recordatorio alimentario 24 horas
- **Questionnaire:** Sueño, actividad física, tabaquismo
- **Wearables** (opcional): Acelerometría en ciclos específicos

#### Variables objetivo (Labels):

**Etiqueta principal (Label A): Alto riesgo de diabetes** - Definida con **A1c** (variable típica LBXGH) y/o **glucosa** - Según criterio entregado en el starter

**Etiqueta alternativa (Label B): Hipertensión** - Estadio 1/2 usando mediciones clínicas de PA (ej: BPXSY\*, BPXDI\*) - Features se limitan a estilo de vida y demografía

#### Split temporal (obligatorio):

- **Entrenamiento/validación:** Ciclos **2007–2016**
- **Test ciego:** Ciclos **2017–Mar 2020**
- Objetivo: Robustez temporal y prevención de *leakage* por ciclo

#### Pesos muestrales y diseño complejo:

- Para estadísticas poblacionales: indicar **peso** (ej: WTMEC2YR)
- Considerar diseño complejo (estrato SDMVSTRA, PSU SDMVPSU)
- **No obligatorio** para el modelo ML, pero se **premia** su correcto uso en descriptivos

## 4) Arquitectura de referencia (IA híbrida)

#### Componentes principales:

1. **Motor de riesgo (ML tabular)**
  - Baseline: **Logistic Regression / XGBoost** con validación temporal
  - Métricas a reportar:
    - **AUROC** (métrica principal)
    - **AUPRC** (precisión-recall)
    - **Brier Score** (calibración)
    - **Fairness** (diferencia absoluta de desempeño entre subgrupos por sexo/edad/grupo étnico NHANES)
2. **Extractor NL→JSON (LLM)**
  - Convierte texto libre del formulario en un **JSON validado** (rangos/unidades)
  - Utiliza *function/tool calling* y validadores
3. **Coach (LLM + RAG)**
  - Genera plan de hábitos personalizado con citas **solo** desde la **mini-KB** local (/kb) proporcionada (guías validadas)
  - **Prohibido alucinar fuentes**
4. **Guardrails & Safety**
  - Umbral de **derivación** cuando el riesgo supere un valor crítico
  - Lenguaje claro, inclusivo y no-diagnóstico
  - Disclaimers visibles
5. **App & API**
  - **FastAPI** para endpoints `/predict` y `/coach`
  - **Streamlit/Gradio** para la demo interactiva
  - Opción de **exportar PDF** del plan personalizado

- **Link compatible** para resultados

## 5) Reglas técnicas obligatorias (anti-fuga y calidad)

Reglas críticas:

- **Anti-fuga de datos:**
  - Si el *label* usa A1c/glucosa/PA clínica, esos campos **NO** pueden ser *features* directas ni derivadas obvias
- **Validación temporal:**
  - **Obligatoria** (por ciclo NHANES)
  - Queda **prohibido** usar *k-fold* aleatorio como única validación
- **Calibración:**
  - Entregar curva de confiabilidad o **Brier Score**
- **Equidad (Fairness):**
  - Reportar métricas por subgrupos (sexo, grupo etario, raza/etnia NHANES)
  - Calcular **gap absoluto** y justificar mitigaciones (si aplica)
- **Reproducibilidad:**
  - Incluir `requirements.txt` o `environment.yml`
  - Seeds fijados en todo el código
  - Scripts documentados para entrenar/evaluar
- **Ética:**
  - *Disclaimer* visible en toda comunicación
  - Reglas de **derivación** a profesional de salud para riesgo alto
- **Privacidad:**
  - **No subir** datos sensibles de personas reales al repositorio

## 6) Entregables (obligatorios)

Lista de entregables requeridos:

1. **Repositorio GitHub**
  - Código organizado en carpetas: `/src`, `/api`, `/app`
  - **README.md** completo con instrucciones de instalación y uso
2. **API mínima (FastAPI)**
  - `POST /predict` → devuelve `{"score": float, "drivers": [top_features]}`
  - `POST /coach` → devuelve plan textual + **citas** a la base de conocimiento `/kb`
3. **App demo interactiva**
  - Implementada en **Streamlit** o **Gradio**
  - Deploy funcional en **Hugging Face Spaces**
4. **Reporte técnico** (2–3 páginas)
  - Debe incluir: datos, ingeniería de features, validación, calibración, fairness, guardrails y limitaciones
5. **Plan PDF descargable** (1–2 páginas)
  - Recomendaciones personalizadas para el usuario
  - *Disclaimer* médico visible
6. **Bitácora de prompts**
  - Prompts clave utilizados
  - Políticas de guardrails implementadas
7. **Presentación final**
  - **Duración:** 10 minutos de presentación + 5 minutos de Q&A
  - **Formato:** Pitch con demo en vivo de la aplicación

- **Contenido obligatorio:**

- Problema y motivación (2 min)
- Solución técnica y arquitectura (3 min)
- Demo de la aplicación (3 min)
- Resultados y métricas (1 min)
- Impacto y próximos pasos (1 min)

## 7) Rúbrica de evaluación (100 puntos)

### A. Rigor técnico ML (30 pts)

Criterio	Puntos	Descripción de niveles de desempeño
<b>A1. Métrica principal (AUROC en test)</b>	<b>12 pts</b>	<b>12 pts:</b> AUROC 0.80 <b>10 pts:</b> AUROC 0.75–0.79 <b>7 pts:</b> AUROC 0.70–0.74 <b>4 pts:</b> AUROC < 0.70
<b>A2. Calibración (Brier Score en test)</b>	<b>6 pts</b>	<b>6 pts:</b> Brier 0.125 <b>5 pts:</b> Brier 0.13–0.15 <b>3 pts:</b> Brier 0.16–0.18 <b>1 pt:</b> Brier > 0.18
<b>A3. Validación temporal &amp; anti-fuga</b>	<b>6 pts</b>	<b>6 pts:</b> Cumple split por ciclo + sin fuga de datos <b>4 pts:</b> Dudas menores en implementación <b>2 pts:</b> Errores de fuga o validación
<b>A4. Explicabilidad (drivers locales)</b>	<b>6 pts</b>	<b>6 pts:</b> Drivers claros, consistentes con modelo y caso <b>4 pts:</b> Explicaciones parciales <b>2 pts:</b> Explicaciones confusas o incorrectas

### B. LLMs, RAG y guardrails (25 pts)

Criterio	Puntos	Descripción de niveles de desempeño
<b>B1. Extractor NL→JSON con validación</b>	<b>8 pts</b>	<b>8 pts:</b> 100% JSON válido + rangos/unidades correctos <b>6 pts:</b> Leves correcciones necesarias <b>3 pts:</b> Errores frecuentes de validación
<b>B2. Coach con RAG y citas válidas</b>	<b>9 pts</b>	<b>9 pts:</b> Todas las recomendaciones con fuentes de /kb <b>7 pts:</b> Alguna omisión menor de citas <b>4 pts:</b> Alucinaciones o citas inválidas
<b>B3. Safety &amp; Derivación</b>	<b>8 pts</b>	<b>8 pts:</b> Umbrales correctos + lenguaje no-diagnóstico + derivación implementada <b>5 pts:</b> Implementación parcial <b>2 pts:</b> Ausente o inadecuado

### C. Producto y UX (25 pts)

Criterio	Puntos	Descripción de niveles de desempeño
<b>C1. App funcional y fluida</b>	<b>10 pts</b>	<b>10 pts:</b> Formulario claro + feedback inmediato + manejo de errores + deploy en Spaces <b>7 pts:</b> Funcional con problemas menores <b>4 pts:</b> Funcionalidad básica limitada
<b>C2. Export &amp; sharing</b>	<b>5 pts</b>	<b>5 pts:</b> PDF descargable + enlace compatible funcional <b>3 pts:</b> Solo una funcionalidad implementada <b>1 pt:</b> Implementación deficiente

Criterio	Puntos	Descripción de niveles de desempeño
<b>C3. Claridad para el usuario</b>	<b>10 pts</b>	<b>10 pts:</b> Mensajes simples + inclusivos + explicación clara del score + próximos pasos <b>7 pts:</b> Claridad adecuada con mejoras menores <b>4 pts:</b> Comunicación confusa o incompleta

#### D. Reproducibilidad y buenas prácticas (15 pts)

Criterio	Puntos	Descripción de niveles de desempeño
<b>D1. Repo &amp; scripts</b>	<b>6 pts</b>	<b>6 pts:</b> requirements.txt/env + Makefile o scripts + semillas fijadas + rutas limpias <b>4 pts:</b> Reproducible con ajustes menores <b>2 pts:</b> Dificultades significativas para reproducir
<b>D2. Documentación</b>	<b>5 pts</b>	<b>5 pts:</b> README claro con pasos + supuestos + estructura de datos <b>3 pts:</b> Documentación básica funcional <b>1 pt:</b> Documentación insuficiente
<b>D3. Métricas por subgrupos (fairness)</b>	<b>4 pts</b>	<b>4 pts:</b> Reporte completo por sexo/edad/grupo étnico + análisis de gap + mitigaciones <b>2 pts:</b> Análisis parcial de equidad <b>1 pt:</b> Análisis superficial o ausente

#### E. Presentación y Pitch Final (15 pts)

Criterio	Puntos	Descripción de niveles de desempeño
<b>E1. Storytelling e impacto</b>	<b>6 pts</b>	<b>6 pts:</b> Narrativa clara + problema bien definido + impacto cuantificado + propuesta de valor convincente <b>4 pts:</b> Storytelling adecuado con elementos menores faltantes <b>2 pts:</b> Presentación confusa o impacto poco claro
<b>E2. Comunicación técnica</b>	<b>5 pts</b>	<b>5 pts:</b> Explica conceptos complejos de forma accesible + demuestra comprensión profunda + responde preguntas técnicas con precisión <b>3 pts:</b> Comunicación técnica adecuada con algunas dificultades <b>1 pt:</b> Dificultades significativas para explicar aspectos técnicos
<b>E3. Formato y tiempo</b>	<b>4 pts</b>	<b>4 pts:</b> Respeta tiempo límite + estructura clara + uso efectivo de herramientas visuales + demo fluida <b>2 pts:</b> Cumple requisitos básicos con problemas menores <b>1 pt:</b> Incumplimiento significativo de formato o tiempo

**Puntuación total:** 100 puntos

**Criterio de desempate:** Se privilegia **mejor calibración** (menor Brier Score) y **menor gap de equidad** entre subgrupos.

#### 8) Cronograma sugerido (27 horas)

**Timeline del Hackathon:**

Horas	Actividades principales
H0–H2	Kickoff + setup del entorno + verificación anti-fuga + baseline inicial
H2–H6	ETL + ingeniería de features + primer modelo ML
H6–H10	Extractor NL→JSON + implementación de validadores
H10–H16	Coach con RAG + implementación de guardrails y reglas de derivación
H16–H22	Desarrollo de App + API + exportación PDF + deploy en Spaces
H22–H24	Calibración del modelo + análisis de fairness + pulido de UX
H24–H26	Preparación de presentación + slides + ensayo de pitch
H26–H27	Presentación final + evaluación

**Tip:** Asignen roles en el equipo desde H0 para trabajar en paralelo (ML, LLM, Frontend, Doc, Presentación).

## 9) Starter kit (estructura de referencia)

Estructura del proyecto sugerida:

```
salud-hackathon-nhanes/
  data/                                # Datos CSV train/test + diccionarios
  kb/                                    # Base de conocimiento (fichas .md para RAG)
  src/
    load.py                             # Código fuente principal
    features.py                         # Carga y merge de datos + tipos
    targets.py                          # Ingeniería de features (IMC, cintura, sueño/actividad, HEI-lite)
    model.py                            # Reglas de etiqueta (LAB solo para label)
    eval.py                             # Entrenamiento + persistencia (joblib)
    rag.py                              # Métricas (AUROC, AUPRC, Brier, fairness)
    prompts.py                         # Entrenamiento + persistencia (joblib)
    prompts.py                         # Buscador simple sobre /kb (BM25)
    prompts.py                         # Templates del Extractor y Coach
  api/
    main.py                            # FastAPI endpoints: /predict y /coach
  app/
    app.py                             # Streamlit/Gradio (formulario + resultados + PDF)
  requirements.txt                     # Dependencias de Python
  README.md                           # Documentación del proyecto
```

Descripción de componentes:

- **/data:** Contiene los archivos CSV con datos NHANES y diccionarios de variables
- **/kb:** Base de conocimiento local con fichas markdown sobre hábitos saludables
- **/src:** Módulos principales del sistema (ML, RAG, validación)
- **/api:** API REST con FastAPI para integración
- **/app:** Interfaz de usuario con Streamlit o Gradio

## 10) JSON Schema para el Extractor NL→JSON

Esquema de validación (ejemplo de referencia):

```
{  
    "$schema": "http://json-schema.org/draft-07/schema#",  
    "title": "Perfil de Usuario para Evaluación de Riesgo Cardiometabólico",  
    "description": "Esquema de validación para datos de entrada del sistema",  
    "type": "object",  
    "properties": {  
        "age": {  
            "type": "integer",  
            "minimum": 18,  
            "maximum": 85,  
            "description": "Edad del usuario en años completos"  
        },  
        "sex": {  
            "type": "string",  
            "enum": ["F", "M"],  
            "description": "Sexo biológico (F=Femenino, M=Masculino)"  
        },  
        "height_cm": {  
            "type": "number",  
            "minimum": 120,  
            "maximum": 220,  
            "description": "Altura en centímetros"  
        },  
        "weight_kg": {  
            "type": "number",  
            "minimum": 30,  
            "maximum": 220,  
            "description": "Peso corporal en kilogramos"  
        },  
        "waist_cm": {  
            "type": "number",  
            "minimum": 40,  
            "maximum": 170,  
            "description": "Circunferencia de cintura en centímetros"  
        },  
        "sleep_hours": {  
            "type": "number",  
            "minimum": 3,  
            "maximum": 14,  
            "description": "Horas de sueño promedio por día"  
        },  
        "smokes_cig_day": {  
            "type": "integer",  
            "minimum": 0,  
            "maximum": 60,  
            "description": "Número de cigarrillos fumados por día"  
        }  
    }  
}
```

```

"days_mvpa_week": {
    "type": "integer",
    "minimum": 0,
    "maximum": 7,
    "description": "Días por semana con actividad física moderada-vigorosa"
},
"fruit_veg_portions_day": {
    "type": "number",
    "minimum": 0,
    "maximum": 12,
    "description": "Porciones de frutas y verduras por día"
},
},
"required": ["age", "sex", "height_cm", "weight_kg", "waist_cm"],
"additionalProperties": false
}

```

**Nota:** Este schema debe usarse para validar la salida del LLM y garantizar tipos, rangos y unidades correctas.

## 11) Política de uso responsable

Principios éticos fundamentales:

### 1. No diagnóstico médico

- El sistema **NO** realiza diagnósticos médicos
- Siempre informar límites del sistema
- Recomendar consulta profesional cuando corresponda
- Disclaimer visible en todas las comunicaciones

### 2. RAG obligatorio y veracidad

- El Coach **DEBE** usar solo la mini-base de conocimiento local (/kb)
- **Prohibido** citar fuentes inexistentes o alucinar información
- Todas las recomendaciones deben estar respaldadas por fuentes verificables

### 3. Transparencia y explicabilidad

- Explicar en lenguaje claro y simple el score de riesgo
- Mostrar qué variables contribuyen al riesgo
- Indicar cómo puede el usuario mejorar su perfil de salud
- No usar terminología técnica sin explicación

### 4. Accesibilidad e inclusión

- Texto comprensible para público general (sin jerga médica)
- Sensible a la diversidad cultural y de género
- Lenguaje inclusivo y respetuoso
- Interfaz clara y fácil de usar

## Anexos

### A. Fórmulas y Métricas

Definiciones técnicas de las métricas de evaluación: - **AUROC** (Area Under ROC Curve) - **AUPRC** (Area Under Precision-Recall Curve) - **Brier Score** (calibración de probabilidades) - **Gap absoluto** (fairness entre subgrupos)

### B. Umbrales de Derivación

Criterios sugeridos para derivación a profesional de salud (se entregan en el starter kit).

### C. Base de Conocimiento (mini-KB)

Lista de fichas temáticas (10–15 tópicos): - Hábitos saludables generales - Sueño y descanso - Actividad física y ejercicio - Nutrición y alimentación - Manejo del estrés - Prevención cardiovascular

### D. Guía para la Presentación Final

Consejos para maximizar la puntuación en la categoría E (Presentación y Pitch):

#### Estructura recomendada:

1. **Hook inicial** (30 seg): Estadística impactante o pregunta provocativa
2. **Problema** (1.5 min): ¿Por qué es importante este desafío en salud?
3. **Solución** (2.5 min): Arquitectura híbrida y enfoque técnico
4. **Demo** (3 min): Funcionalidad clave con caso de uso real
5. **Resultados** (1 min): Métricas principales y validación
6. **Impacto** (1 min): Escalabilidad y próximos pasos

#### Tips de presentación:

- **Preparen backup:** Screenshots por si falla la demo
- **Practiquen el timing:** 10 minutos exactos
- **Preparen respuestas:** Para preguntas técnicas comunes
- **Visuales claros:** Slides simples, pocos textos
- **Storytelling:** Conecten con el impacto en salud real

---

## ¡Éxito en el Hackathon!

*Desarrollando el futuro de la salud preventiva con Inteligencia Artificial*

Duoc UC - Formando profesionales para el mañana